How Width Scaling Affects Neural Networks: Generalization, Optimal Hyperparameters, Feature Learning and Beyond

Dissertation
der Mathematisch-Naturwissenschaftlichen Fakultät
der Eberhard Karls Universität Tübingen
zur Erlangung des Grades eines
Doktors der Naturwissenschaften
(Dr. rer. nat.)

vorgelegt von
M.Sc. Moritz Haas
aus Usingen

Tübingen
2025

Gedruckt mit Genehmigung der Mathematisch-Naturwissenschaftlichen Fakultät der Eberhard Karls Universität Tübingen.

Tag der mündlichen Qualifikation	28. Juli 2025
Dekan:	Prof. Dr. Thilo Stehle
1. Berichterstatterin:	Prof. Dr. Ulrike von Luxburg
2. Berichterstatter:	Prof. Dr. Matthias Hein

Abstract

In recent years, deep artificial neural networks have shown increasingly impressive learning capabilities. With growing computation resources, the size of these networks keeps growing and their generalization performance continues to improve. In this thesis, we aim to bridge the gap between theory and practice of deep learning by developing theoretical understanding of large-scale artificial neural network training that is empirically predictive at moderate scale, proves fundamental limitations or results in practical benefits. In particular, we contribute to the understanding of width-scaling properties of neural networks through a combination of infinite-width theory and empirical evaluations of theoretical predictions.

First, we study when and how overfitting of wide neural networks in the *neural tangent parameterization* (NTP) can generalize well. For this purpose, we significantly generalize previous inconsistency results for kernel regression in fixed input dimension to overfitting with many common neural estimators beyond the minimum norm interpolant. But we also show that suitable spiky-smooth estimators can overfit benignly with minimax-optimal convergence rates in arbitrary covariate dimension. For wide neural networks in NTP, such a spiky-smooth inductive bias can be induced by adding a single shifted high-frequency low-amplitude sin-curve to the activation function. Thereby we demonstrate that overfitting is neither intrinsically beneficial nor harmful for generalization with the right choice of estimator, irrespective of the input dimension.

Second, we study width-dependent parameterizations for Sharpness Aware Minimization (SAM). While for stochastic gradient descent and Adam the *Maximal Update Parameterization* (μP) has been shown to induce hyperparameter transfer and improved generalization at large width, we prove that training with SAM in μP with a global perturbation radius only effectively perturbs the last layer. This observation motivates characterizing all possible width-dependent choices of layerwise initialization variances, learning rates and perturbation radii into unstable, vanishing, non-trivial and effective perturbation parameterizations. We find that there exists a unique stable parameterization, we call *Maximal Update and Perturbation Parameterization* (μP^2), that achieves width-independent feature learning and width-independent perturbations of the trainable weights in all layers. In experiments training multilayer perceptrons and ResNets on CIFAR-10 as well as Vision Transformers on ImageNet-1K with SAM, we observe that μP^2 improves generalization and jointly transfers both the optimal learning rate and perturbation radius from small to large width, as opposed μP with global perturbation scaling. This confirms that width-independent training dynamics induce empirically favorable

and predictable scaling properties, but also shows that non-standard optimization algorithms can require scaling considerations that go beyond μP.

Third, we study the dominant width scaling practice (*standard parameterization*, SP): Networks are initialized with He initialization and trained using a single learning rate for all trainable parameters, tuned at each model scale. Existing infinite-width theory would predict instability under large learning rates and vanishing feature learning under stable learning rates. However, empirically optimal learning rates consistently decay much slower than theoretically predicted. We identify the cross-entropy loss as the key component that enables stable training under large learning rates allowing stable hidden-layer feature learning despite logit divergence, even in the infinite-width limit. We empirically validate that infinite-width predictions hold at moderate width, and therefore provide the first infinite-width proxy for SP that remains predictive of practical neural networks in this controlled divergence regime.

Overall, our results suggest that the width dependence in practical neural network training is surprisingly predictable with Tensor Program-based analyses, even over the course of long training. This enables understanding and correcting real-world neural network scaling. Our findings reinforce that corrected width scaling has significant downstream impact on hyperparameter transfer, feature learning and consequently generalization and predictability at large model scale. However, all relevant architecture and training components such as the activation function, the loss function and intermediate perturbation steps need to be taken into account to arrive at the correct qualitative conclusions and to find the correct width-scaling rules that achieve width-independent training dynamics.

Zusammenfassung

In den letzten Jahren haben tiefe künstliche neuronale Netze zunehmend beeindruckende Lernfähigkeiten gezeigt. Mit zunehmenden Rechenressourcen steigt die Größe der verwendeten Netze immer weiter und ihre Generalisierungsleistung wird immer besser. In dieser Arbeit verfolgen wir das Ziel, die Lücke zwischen Theorie und Praxis des Deep Learning zu schließen, indem wir ein theoretisches Verständnis für das Training großer künstlicher neuronaler Netze entwickeln, das bei moderater Größe empirisch prädiktiv ist, grundlegende Beschränkungen aufzeigt oder zu praktischen Verbesserungen führt. Insbesondere tragen wir zum Verständnis der Breitenskalierungseigenschaften neuronaler Netze bei, durch eine Kombination von Theorie im Limes unendlicher Breite und empirischer Evaluation der theoretischen Vorhersagen.

Zunächst untersuchen wir, wann und wie überangepasste, breite neuronale Netzen in der Neural Tangent Parametrisierung (NTP) gute Vorhersagen auf neuen Testdaten liefern können. Zu diesem Zweck erweitern wir frühere Inkonsistenzresultate für Kernregression bei fester Datendimension auf viele gängige, überangepasste neuronale Schätzer jenseits des Minimum-Norm-Interpolierers. Wir zeigen aber auch, dass geeignete stachelig-glatte Schätzer mit minimax-optimalen Konvergenzraten in beliebiger Kovariaten-Dimension gutartig überanpassen können. Für breite neuronale Netze in NTP kann ein solcher stachelig-glatter induktiver Bias durch Hinzufügen einer einzelnen verschobenen hochfrequenten Sinuskurve mit niedriger Amplitude zur Aktivierungsfunktion induziert werden. Auf diese Weise zeigen wir, dass eine Überanpassung bei der richtigen Wahl des Schätzers, unabhängig von der Eingangsdimension, weder vorteilhaft noch schädlich für die Generalisierung auf unabhängige Testdaten ist.

Zweitens untersuchen wir breitenabhängige Parametrisierungen für Sharpness Aware Minimization (SAM). Während für stochastischen Gradientenabstieg und Adam gezeigt wurde, dass die *Maximal-Update-Parametrisierung* (μP) Hyperparameter-Transfer und eine verbesserte Generalisierung bei großer Breite bewirkt, beweisen wir, dass das Training mit SAM in μP mit einem globalen Perturbationsradius nur das letzte Netzwerklayer effektiv perturbiert. Diese Beobachtung motiviert die Charakterisierung aller möglichen breitenabhängigen layerweisen Initialisierungsvarianzen, Lernraten und Perturbationsradien in instabile, verschwindende, nicht-triviale und effektive Perturbationsparametrisierungen. Wir stellen fest, dass es eine einzige stabile Parametrisierung gibt, die wir als *Maximale Update- und Perturbationsparametrisierung* (μP^2) bezeichnen, die ein breitenunabhängiges Lernen und breitenunabhängige Perturbationen der trainierbaren Gewichte in allen Layern erreicht. In Experimenten, in denen wir mehrschichtige Perzeptren und ResNets auf CIFAR-10 sowie Vision

Transformer auf ImageNet-1K mit SAM trainieren, stellen wir fest, dass μP^2 die Generalisierung verbessert und sowohl die optimale Lernrate als auch den optimalen Perturbationsradius von kleiner zu großer Breite überträgt, im Gegensatz zu μP mit globaler Perturbationsskalierung. Dies bestätigt, dass breitenunabhängige Trainingsdynamik empirisch günstige und vorhersehbare Skalierungseigenschaften hervorruft, zeigt aber auch, dass neue Optimierungsalgorithmen Skalierungsüberlegungen erfordern können, die über μP hinausgehen.

Drittens untersuchen wir die vorherrschende Breitenskalierungspraxis (*Standardparametrisierung*, SP): Netze werden mit He-Initialisierung initialisiert und mit einer einzigen Lernrate für alle trainierbaren Parameter trainiert, die bei jeder Modellgröße angepasst wird. Die existierende Theorie im Limes unendlicher Breite würde Instabilität bei großen Lernraten und verschwindendes Muster-Lernen bei stabilen Lernraten vorhersagen. Empirisch optimale Lernraten schrumpfen jedoch durchweg viel langsamer als theoretisch vorhergesagt. Wir identifizieren die Cross-Entropie-Verlustfunktion als die Schlüsselkomponente, die ein stabiles Training bei großen Lernraten ermöglicht und trotz Logit-Divergenz ein stabiles Muster-Lernen erlaubt, sogar im Limes unendlicher Breite. Wir validieren empirisch, dass Vorhersagen bei unendlicher Breite auch bei moderater Breite zutreffen, und liefern somit den ersten Proxy für SP mit unendlicher Breite, der für praktische neuronale Netzwerke in diesem kontrollierten Divergenzregime prädiktiv bleibt.

Insgesamt deuten unsere Ergebnisse darauf hin, dass die Breitenabhängigkeit beim praktischen Training neuronaler Netze mit Tensor-Programm-basierten Analysen überraschend gut vorhersagbar ist, sogar im Verlauf langen Trainings. Dies ermöglicht das Verständnis und die Korrektur der Skalierung praktisch-verwendeter neuronaler Netze. Unsere Ergebnisse bestätigen, dass die korrigierte Breitenskalierung einen signifikanten Einfluss auf den Transfer von Hyperparametern, das Lernen von Mustern und folglich auf die Generalisierung und Vorhersagbarkeit bei großen Modellgrößen hat. Allerdings müssen alle relevanten Architektur- und Trainingskomponenten wie die Aktivierungsfunktion, die Verlustfunktion und eventuelle Perturbationsschritte berücksichtigt werden, um zu den richtigen qualitativen Schlussfolgerungen zu gelangen und die richtigen Breitenskalierungsregeln zu finden, die eine breitenunabhängige Trainingsdynamik erreichen.

Acknowledgements

I am deeply grateful for the mentors, collaborators, friends and family in my life without whom this thesis would not exist.

I feel very fortunate to have had Ulrike von Luxburg as a supervisor. I am grateful for her unceasing support over the past 4 years, for providing the freedom to explore and foster my own interests, and for serving as a prime role model that it is possible to stick to your own beliefs and ideals against partially misaligned incentive structures. Thank you for creating a friendly and open research environment and for your invaluable advice, both professionally and personally. I would also like to thank Bedartha Goswami, who embodies and encouraged the same core values of curiosity driven research, open-mindedness and independent thinking.

I would also like to thank my collaborators and friends who have taught me a lot, made my scientific endeavors much more enjoyable and who have played a key role in shaping my research. I thank Stefan Richter for his support during my master thesis and his unceasing display of devotion and genuine interest. Without you I would have probably never started this journey. I would like to thank David Holzmüller for taking on our close collaboration early in my PhD and guiding me during my entrance to kernel theory, fast email replies and enjoyable times at jazz concerts or on the road. I am deeply grateful for the mentorship by Leena Chennuru Vankadara. Her perspectives on research and life have been truly enriching, and her respect, patience and support have enabled me immensely. Also thank you and Faiz for your incredible hospitality in London. I am very grateful for our friendship and collaboration. I am also grateful for valuable discussions with and advice from Ingo Steinwart, Volkan Cevher and Jin Xu.

I had a great time with members of the tml-group and the mlcs-group. Whether we went hiking or to a bar, played spike ball or board games, I could have not wished for a friendlier atmosphere: Solveig Peter, Sebastian Bordt, Gunnar König, Karolin Frohnapfel, Robi Bhattacharjee, Michael Lohaus, Luca Rendsburg, Eric Günther, David Künstle, Damien Garreau, Jakob Schlör, Jannik Thümmel, and Felix Strnad. I would also like to thank all devoted organizers that make the Tübingen research community such a friendly and lively environment.

I am deeply grateful for the unconditional support by my family and friends, especially my parents, Arlette, Alex, Oscar, Silvia, Matias and Phillip who have formed my life's unshakable foundation. Thank you Caro for the invaluable experiences we share. To my mother, no amount of gratitude can pay back the selfless, unconditional love and countless lessons you have provided me to grow and flourish in my own ways.

Contents

Abstract ii

Zusammenfassung (German Abstract) iv

Acknowledgements vii

I Introduction 1

1 Recent Advances in Machine Learning Are Driven By Scale 3
- 1.1 A Short History of Deep Learning . . . 4
- 1.2 Progress through Principled Understanding . . . 4

2 Statistical Understanding of How and What Large Neural Networks Learn 7
- 2.1 Setting: Learning with Neural Networks . . . 7
- 2.2 Kernel Theory as a Tool for Understanding Representations and Learning Dynamics . . . 8
- 2.3 Infinite-width Limits of Neural Networks . . . 11
  - 2.3.1 Neural networks at initialization . . . 11
  - 2.3.2 Training Wide Neural Networks in Popular Parameterizations . . . 12
- 2.4 Tensor Program Framework . . . 17
- 2.5 Practical Considerations for Neural Network Scaling . . . 19

3 Thesis Contributions 23
- 3.1 Benign Overfitting of Kernels and Extensively Wide Neural Networks . . . 24
  - 3.1.1 Prior State of the Literature . . . 24
  - 3.1.2 Summary and Contributions . . . 28
- 3.2 Width-independent Training Dynamics for Sharpness Aware Minimization . . . 29
  - 3.2.1 Prior State of the Literature . . . 29
  - 3.2.2 Summary and Contributions . . . 31
- 3.3 Understanding the Effectiveness of Standard Width Scaling . . . 32
  - 3.3.1 Prior State of the Literature . . . 32
  - 3.3.2 Summary and Contributions . . . 35

II Publications 37

4 Mind the Spikes: Benign Overfitting of Kernels and Neural Networks in Fixed Dimension 39

5 μP^2: Effective Sharpness Aware Minimization Requires Layerwise Perturbation Scaling 105

6 On the Surprising Effectiveness of Large Learning Rates under Standard Width Scaling 179

III Discussion 251

7 Discussion 253
7.1 Predictable Scaling Laws 253
7.2 Generalization 255
7.3 Concluding remarks 255

License Information 257

8 Bibliography 259

Part I

Introduction

Chapter 1

Recent Advances in Machine Learning Are Driven By Scale

In recent years, the potential of large-scale deep learning models has become ever more apparent. In particular the deployment of general-purpose large language model chat bots like ChatGPT (Brown et al., 2020; Achiam et al., 2023) has impacted peoples' day-to-day lives around the world, from teachers being unable to deny language model use by students, over casual users asking for cooking recipes to programmers boosting their productivity by using coding assistant tools. While some people believe that Artificial General Intelligence (AGI) is within reach and set achieving AGI as their primary goal (Altman, 2023), other renowned experts warn about imminent dangers of missuse of such powerful technology like personalized risk predictions, influenced elections, increased power imbalances, privacy and copyright infringements concerns, a distorted perception of truth through 'deep fake' campaigns or even human extinction by rogue AIs (Kleinman and Vallance, 2023; Wang et al., 2024; Bengio, 2024; Carr, 2024; Barbera, 2025). But all of these experts share the common believe in the transformative potential of AI in science, education and industry, and many experts predict an imminent industrial revolution (Metz, 2023; Devlin, 2023; Abis and Veldkamp, 2024; Marr, 2024).

Remarkable learning abilities have already been achieved in vision (Dall-E 2 Ramesh et al., 2022), natural language (ChatGPT Brown et al., 2020; Achiam et al., 2023, Olmo 2 OLMo Team et al. (2024), LLama 3 Grattafiori et al. (2024), ...), vision-language (CLIP Radford et al., 2021), but also in the sciences including applications like protein folding (AlphaFold Jumper et al., 2021), mathematics (AlphaGeometry Trinh et al., 2024), tabular data (TabPFN Hollmann et al., 2025) as well as weather and climate prediction (GraphCast Lam et al., 2023, Aurora Bodnar et al., 2024, NeuralGCM Kochkov et al., 2024). All of these models use powerful and scalable neural network architectures, mostly Transformers (Vaswani et al., 2017), trained on vast amounts of curated data with gradient-based training procedures at gigantic scales.

A continued development of these systems around the world seems unstoppable. Thus their positive and negative potential calls for a responsible development and design of these systems before their release.

1.1 A Short History of Deep Learning

The idea of using artificial neural networks for learning dates back to Rosenblatt (1958). However, their full potential could not be realized back then due to computational limitations. Much later, efficient implementations of the theoretical tool of backpropagation enabled fast optimization of neural networks with gradient-based algorithms (Rumelhart et al., 1986b; LeCun et al., 1989, 1998). In parallel, Hochreiter and Schmidhuber (1997a) worked on deep learning for time series. It took further exponential growth of parallelized computational resources and curated benchmarks (Deng et al., 2009; Hardt, 2025), before AlexNet showed remarkable learning capabilities of deep neural networks for ImageNet-scale image classification (Krizhevsky et al., 2012). This advance required the previous development of (i) convolutional neural networks (CNNs) (LeCun and Bengio, 1995), encoding useful inductive biases toward image structure in the architecture, (ii) ReLU nonlinearities to prevent vanishing gradient signals and (iii) an efficient GPU implementation of the parallelizable computations. In parallel, the foundations of language modelling were layed by Mikolov et al. (2013) and Bahdanau et al. (2014), but earlier attention mechanisms date back to Rumelhart et al. (1986a), Schmidhuber (1992) and many more, reviewed in Niu et al. (2021). Ultimately, Vaswani et al. (2017) proposed the attention architecture, called Transformer, that dominates today's large language models (LLMs), up to minor modifications. Although the field of neural network optimization is still very active (Gupta et al., 2018; Jordan et al., 2024), the Adam optimizer (Kingma and Ba, 2014; Loshchilov and Hutter, 2019) has prevailed as a general-purpose optimization algorithm across architectures and data modalities.

Today's deep learning landscape is the result of an iterative process of incremental improvements, many heuristics and careful hyperparameter tuning. Often statistical considerations like signal propagation through the activation function, initialization variance or residual connections (Nair and Hinton, 2010; He et al., 2015, 2016) together with computational scaling enabled deep network training and laid the foundation for breakthroughs on new data modalities.

The amount of resources and attention put into deep learning research is still growing. Every year, a growing number of researchers submit more papers to leading conferences with NeurIPS 2025 exceeding 25 000 submissions. Every year more money is invested in big compute clusters. Meta AI, xAI and Microsoft/OpenAI run their own supercomputers containing at least 100 000 NVidia H100 GPUs, each equivalent to approximately $9.9 \cdot 10^{19}$ FLOP/s and a cost of $4 \cdot 10^{9}$ US dollars. Scale still appears to be a dominating factor in the arms race toward increasingly powerful multi-modal AI, and progress, novel applications as well as increase in energy consumption is projected to continue at an overstraining pace for regulators, society, individuals and Earth's limited resources.

1.2 Progress through Principled Understanding

The astronomical scales and impacts of AI research entail a big responsibility for the research community at this critical point in time. Both the responsible development of AI systems as well as their projected energy consumption in the near future will greatly impact future generations. At the same time, principled understanding

of these increasingly complex systems and training procedures is far outpaced by practical advances. But such principled understanding may yield crucial insights into which components drive which learning mechanisms, and may consequently inform principled improvements toward efficient use of resources and toward guaranteeing desired learning properties. Without provable guarantees, the adoption of these models in critical contexts such as medicine or education is lacking trustworthiness, even if predictions are accurate most of the time. With the EU AI act, regulators have begun to formalize such reliability requirements.

In this thesis, we restrict our attention towards understanding the scaling and learning properties of such large artificial neural networks from a predominantly statistical perspective. On a high level we ask:

Can we understand what and how neural networks learn from a statistical and learning theoretic perspective? And can this understanding inform improved, principled and possibly simplified, more efficient or easy-to-use learning procedures?

While deep understanding of training dynamics is still limited to 2-layer ReLU nets, (deep) linear nets or even just linear models under strong distributional assumptions (Ren et al., 2025; Kunin et al., 2024; Zhang et al., 2025; Tsigler et al., 2025), training dynamics of practical neural networks may still simplify in important aspects that may be grasped by theory. A parallel can be drawn to statistical physics (Zdeborová, 2020). There the dynamics of large complex systems of interacting particles can simplify significantly and evolve predictably at macroscopic scales as a function of few interpretable summary statistics such as temperature and pressure.

Indeed, for state of the art Transformers, Kaplan et al. (2020) report robust (sums of) scaling laws $\mathcal{L}(t, N) = \mathcal{L}_0 + C_t \cdot t^{\alpha_t} + C_N \cdot N^{\alpha_N}$, where $\mathcal{L}$ denotes the validation cross-entropy loss after training, and the scaling dimensions t and N denote training time and number of parameters. The scaling exponents α_t and α_N generally depend on the architecture and dataset (Bordelon et al., 2024a; Bachmann et al., 2023). These scaling laws inform the compute-optimal choice of model size N and training time t, given a fixed compute budget C, since gradient-based training satisfies $C \propto N \cdot t$ (Hoffmann et al., 2022; Porian et al., 2024). Finite-width loss corrections have been consistently observed to decay as $1/\mathit{width}$ for single pass training irrespective of the parameterization and task (Dyer and Gur-Ari, 2019; Bahri et al., 2024; Bordelon et al., 2024a), but non-trivial exponents emerge in multi-pass settings (Vyas et al., 2024; Kaplan et al., 2020). We show in Chapter 6 that the amount of feature learning as a function of width also follows predictable exponents, building on the seminal work by Yang and Hu (2021). Often the scaling exponents α corresponding to loss improvement are tiny. But, since models with billions of parameters have never been trained before, likely there still exist many suboptimalities in the training procedure. Even slight improvements in these exponents α or at least the constants C directly translate into significant savings in compute, money and CO_2 emissions for a given large-scale model. Although, by the Jevons paradox (Jevons, 1866), improved model capability and efficiency will likely not reduce resource use, improved models may speed up technological progress and thereby the transition to more sustainable technologies.

In this thesis, we focus on model scaling to large network width. This scaling dimension has been particularly amenable to studies in the past (Neal, 1996; Jacot et al., 2018; Lee et al., 2019; Yang and Hu, 2021), because trainable weights are typically

initialized i.i.d. Gaussian and large sums of independent variables are well understood through central limit theorems and the law of large numbers. Stochastic Gradient Descent (SGD) updates are low-rank and analytically tractable in certain cases.

While tight bounds on generalization require architecture and data-dependent analyses, which is only mathematically tractable in simplified toy settings, there are weaker objectives which allow much more general statements and thereby gain big practical impact. As an important example, fundamental signal propagation considerations are universally valid across all applications from computer vision to natural language processing: Activations and their updates in each layer should neither vanish nor explode with width. To achieve these desiderata, we can adjust the initialization variance and learning rates of each trainable weight matrix with width. We call such a width-scaling rule *parameterization*. Yang and Hu (2021) show for SGD training and Yang and Littwin (2023) for Adam that there exists a unique parameterization (up to smaller last-layer initialization) in which the updates of each trainable weight as well as their initialization have a width-independent effect on the output function. They call this prescription the *Maximal Update Parameterization* (μP). This infinite-width theory is appealing in the following sense: After formulating conditions of ideal scaling properties such as stability and layer-balanced feature learning, it allows to rule out all other parameterizations that break some of these constraints, under weak assumptions on the architecture and dataset. As a consequence, practitioners gain clarity about which network properties are well-understood and controlled. It has been empirically validated in practical settings that network scaling in μP behaves width-independently (Yang et al., 2022; Vyas et al., 2024; Noci et al., 2024b), which entails impactful consequences such as predictability and optimality at large scales. While deep learning practice typically involves many heuristics and extensive hyperparameter tuning at each model scale before the final well-performing training run, networks scaled in μP can be tuned at some small base width and optimal hyperparameters transfer from small to large scale. While networks with a single global learning rate lose feature learning in the input layer or become unstable at large model width and hence often do not monotonically improve with scale, networks in μP preserve stable feature learning at all scales and have consequently been observed to monotonically improve with scale (Yang et al., 2022; Bordelon et al., 2024a).

In the same spirit, our goal in this thesis is to develop principled and generalizable understanding of the network properties that drive and inhibit optimal learning. In this way, we uncover fundamental limitations but also find principled improvements to correct standard practice and to motivate more reliable training practices in the future.

Before discussing our own contributions in Chapter 3, we provide some background knowledge on infinite-width theory in the next section.

Chapter 2

Statistical Understanding of How and What Large Neural Networks Learn

In this section, we recapitulate the technical background on infinite-width theory of neural networks that provides the basis for our work in the subsequent sections. We first specify what exactly we mean by learning neural networks from data in Section 2.1. We then present some basic results in kernel theory in Section 2.2, which provides an invaluable tool for describing the functions learned in the infinite-width limit in certain parameterizations in Section 2.3. Lastly, we summarize the practical recommendations for width scaling from infinite-width theory in Section 2.5.

2.1 Setting: Learning with Neural Networks

Learning objective. Throughout this thesis, we consider the classical learning tasks of classification and regression. Assume labeled data points $(\xi, y) \in \mathbb{R}^{d_{\text{in}}} \times \mathbb{R}^{d_{\text{out}}}$ are drawn i.i.d. from some probability distribution $\mathbb{P}^{(\xi,y)}$ in Euclidean space.

In regression, the goal is to learn the Bayes-optimal regression function $f^*(\xi) = \mathbb{E}[y|\xi]$, based on a finite set of potentially noisy labeled training points $\{(\xi_t, y_t)\}_{t\in[T]}$. A typical approach to solve this inference task is minimizing the Mean-Square Error (MSE), $\mathcal{L} : \mathbb{R}^{d_{\text{out}}} \times \mathbb{R}^{d_{\text{out}}} \to \mathbb{R}$, $\mathcal{L}(f, y) = \|f - y\|_2^2$. This loss function is popular due to its mathematical tractability.

In classification, we assume that each label denotes the probability of the data point ξ to belong to each of the finitely many classes $[C] := \{1, 2, \ldots, C\}$. Often, one-hot labels $y = e_c$, $c \in [C]$, are used to denote deterministic membership to a particular class. In both image classification and next-token prediction in NLP, the network $\tilde{f}$ is then trained to minimize the expected cross-entropy loss $\mathcal{L} : \mathbb{R}^{d_{\text{out}}} \times \mathbb{R}^{d_{\text{out}}} \to \mathbb{R}$, $\mathcal{L}(\tilde{f}, y) = -\sum_{c=1}^{C} y_c \log(\tilde{f}_c)$. But importantly, when using `torch.nn.CrossEntropyLoss`, output logits are first passed through a softmax $\sigma(f)_i = \exp(f_i)/(\sum_{j\in[C]} \exp(f_j))$, which yields a different effective loss function $\mathcal{L}(f, y) = -\sum_{c=1}^{C} y_c \log(\sigma(f)_c)$.

There exist many procedures for approximately minimizing the expected loss. We are particularly interested in the case of iteratively optimizing neural networks.

Neural network architecture. A simple but common neural network building block is a fully connected neural network, also called multilayer perceptron (MLP), that just consists of alternating linear transformations and elementwise nonlinearity. These networks can already approximate all functions arbitrarily well at sufficient width. Thus other architectures such as convolutional neural networks (CNNs) or Transformers are only chosen because of their suitable inductive biases that guide the common optimization algorithms to learn better-generalizing solutions for certain data modalities of interest such as images or language.

Definition 1 (Multilayer Perceptron). We iteratively define a *L-hidden layer multilayer perceptron (MLP) of width n* (equivalently, a $(L+1)$-layer MLP) with trainable weight matrices $W^1 \in \mathbb{R}^{n \times d_{in}}$, $W^l \in \mathbb{R}^{n \times n}$ for $l \in [2, L]$, and $W^{L+1} \in \mathbb{R}^{d_{\text{out}} \times n}$, elementwise nonlinearity $\phi : \mathbb{R} \to \mathbb{R}$ and for inputs $\xi \in \mathbb{R}^{d_{in}}$ via

$$h^1(\xi) := W^1 \xi, \qquad x^l(\xi) := \phi(h^l(\xi)), \qquad h^{l+1}(\xi) := W^{l+1} x^l(\xi), \qquad f(\xi) := W^{L+1} x^L(\xi).$$

We call h^l *preactivations*, x^l *activations*, and $f(\xi)$ *output logits*. ◀

We use the `fan`-notation for weight matrices W of arbitrary shapes, meaning $W : \mathbb{R}^{\texttt{fan\_in}} \to \mathbb{R}^{\texttt{fan\_out}}$. Weights are predominantly initialized with He initialization (He et al., 2015) or Glorot initialization (Glorot and Bengio, 2010), which means i.i.d. Gaussian entries $\mathcal{N}(0, C/\texttt{fan\_in})$. This choice of initialization variance ensures that the expected variance in the activations remains width-independent throughout the first forward pass due to the Central Limit Theorem (CLT).

For finding a loss-minimizer f within the class of L-hidden layer MLPs, the trainable weight matrices $\{W^l\}_{l \in [L+1]}$ are usually iteratively updated with gradient-based optimization algorithms on mini batches. For example, a stochasic gradient descent (SGD) update on training point (ξ_t, y_t) with learning rate $\eta > 0$ is given by

$$W^l_{t+1} = W^l_t - \eta \cdot \nabla_{W^l} \mathcal{L}(f_t(\xi_t), y_t),$$

where f_t denotes the output logit function when evaluating the MLP with weights $\{W^l_t\}_{l \in [L+1]}$ at time t.

2.2 Kernel Theory as a Tool for Understanding Representations and Learning Dynamics

The last-layer activations x^L are often called representations. A central objective in deep learning is learning representations that transfer to other image datasets or that in some sense distill essential data structures. The function that maps inputs to last-layer activations $\xi \mapsto x^L_t(\xi)$ can be interpreted as a feature map. Fixing this feature map and only training the output layer results in kernel regression. In turn, kernel regression can be seen as linear linear regression not in the original data space, but in a possibly infinite-dimensional Hilbert space. The hope is that while the original dataset has non-linear structure, the right feature map $\phi : \mathcal{X} \to \mathcal{H}$ lifts the data to a richer space where all important structure becomes linear.

Example 2 (Linear separation through feature maps). As a toy example, assume that the data of class -1 is drawn from the uniform distribution on the unit circle

$B_1(0) \subset \mathbb{R}^2$, whereas all data of class $+1$ is drawn from the uniform distribution on the circle $B_2(0) \subset \mathbb{R}^2$ of radius 2. Any linear function $f : \mathbb{R}^2 \to \{-1, +1\}$ will misclassify exactly half of the points in each class. But the feature map $\phi : \mathbb{R}^2 \to \mathbb{R}^2$ with $\phi(\xi) = (||\xi||_2, 1)$ suffices to linearly separate the two classes, and the classifier $f(\xi) = \operatorname{sgn}(\phi_1(\xi) - 1.5 \cdot \phi_2(\xi))$, that is linear in feature space, perfectly classifies all points in the support. ◀

In the absence of handcrafted feature maps, we would like to use rich general-purpose feature maps that are applicable to many different datasets. Clearly, by adding further features like polynomials of increasing degree, increasingly intricate structure becomes linearly separable in feature space. An important insight is that common linear estimators like linear support vector machines or the solution of ordinary least squares can be purely expressed in terms of inner products between the features. When we want to use such estimators, we can circumvent explicitly using the feature map and instead directly compute their inner product evaluations $\langle \cdot, \cdot \rangle_{\mathcal{H}}$ in feature Hilbert space $\mathcal{H}$. This is called the 'kernel trick', and defines the kernel $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ related to the feature map $\phi : \mathcal{X} \to (\mathcal{H}, \langle \cdot, \cdot \rangle_{\mathcal{H}})$,

$$k(\xi_1, \xi_2) := \langle \phi(\xi_1), \phi(\xi_2) \rangle_{\mathcal{H}}.$$

This shifts the task of finding a desirable feature map ϕ to finding a desirable kernel function k, when defining (reproducing) kernels as follows.

Definition 3 (Reproducing Kernel Function). A *kernel function* is symmetric, and for all $n \in \mathbb{N}$, all $\xi_1, \ldots, \xi_n \in \mathcal{X}$, the *kernel matrix* $\boldsymbol{K} \in \mathbb{R}^{n \times n}$ with entries $K_{ij} = k(\xi_i, \xi_j)$ is positive semi-definite, that is $c^\top \boldsymbol{K} c \geq 0$ for all vectors $c \in \mathbb{R}^n$. ◀

Indeed, there is a one-to-one correspondence between feature maps to Hilbert spaces and kernel functions. A feature map immediately induces a unique kernel function, where symmetry and positive definiteness follow from the properties of scalar products. For the reverse direction, given a kernel function k, its corresponding feature map can be written as $\phi : \mathcal{X} \to \mathbb{R}^{\mathcal{X}}$, $\xi \mapsto k(\xi, \cdot)$ to the space $\mathbb{R}^{\mathcal{X}} = \{f : \mathcal{X} \to \mathbb{R}\}$ of all real-valued functions from $\mathcal{X}$ to $\mathbb{R}$. Now define the corresponding scalar product for finite linear combinations $f = \sum_i \alpha_i k(\xi_i, \cdot)$ and $g = \sum_j \beta_j k(\tilde{\xi}_j, \cdot)$, as $\langle f, g \rangle_{\mathcal{H}} = \sum_{i,j} \alpha_i \beta_j k(\xi_i, \tilde{\xi}_j)$, and finally take the topological completion of the space of all such finite linear combinations to arrive at the Hilbert space that ϕ maps to. The basic properties of RKHS were already proven by Aronszajn (1950), and the feature space interpretation was first mentioned in Aizerman (1964).

As alluded to earlier, we want to choose the kernel function such that the data structure becomes linearly separable and easy to learn in the Hilbert space. To give a precise notion of richness of a feature map or kernel function, we define universal kernels.

Definition 4 (Universal Kernels). A continuous kernel k on a compact metric space $\mathcal{X}$ is called *universal* if the RKHS $\mathcal{H}$ of k is dense in $C(\mathcal{X})$, that is $\forall g \in C(\mathcal{X}), \varepsilon > 0$ there exists $f \in \mathcal{H}$ such that $\|f - g\|_\infty \leq \varepsilon$. ◀

In this thesis, we consider universality as a minimal requirement for desirable kernel functions. The kernel should at least enable us to approximate all continuous functions. In Chapter 4, we will prove that this does not mean that common estimators

will learn well-generalizing solutions on finite noisy datasets with common universal kernels such as the Laplace kernel, Gaussian kernel or more modern neural kernels that we introduce in the next section.

Let us now turn to learning algorithms with kernels. In this thesis, we consider problems of the following form:

On arbitrary input and output spaces $\mathcal{X}, \mathcal{Y}$, with loss function $\mathcal{L} : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}$, $\lambda \geq 0$ and finite training set $\{(\xi_i, y_i)\}_{i=1,\dots,N} \subseteq \mathcal{X} \times \mathcal{Y}$, minimize

$$\min_{w \in \mathcal{H}} \frac{1}{N} \sum_{i=1}^{N} \mathcal{L}(\langle w, \phi(\xi_i) \rangle_{\mathcal{H}}, y_i) + \lambda \Omega(\|w\|_{\mathcal{H}}), \tag{2.1}$$

with $\Omega : [0, \infty) \to \mathbb{R}$ strictly monotonically increasing. We call them *kernel regression/classification*, depending on the loss function. Then the representer theorem states that a minimizer of this risk in the RKHS is spanned by the basis functions evaluated on the training points $\{k(\xi_i, \cdot)\}_{i \in [N]}$.

Theorem 5 (Representer Theorem). *Consider a regularized risk minimization problem of the form (2.1). Then there exists an optimizer $w^* \in \mathcal{H}$ of the form*

$$w^* = \sum_{i=1}^{N} \alpha_i k(\xi_i, \cdot).$$

In other words, kernel regression/classification on a finite training set is always optimized inside the span induced by the covariates $\{k(\xi_i, \cdot)\}_{i=1,\dots,N}$, and the subspace relevant to finding an optimizer becomes finite dimensional.

More concretely, kernel ridge regression with training points $(\xi_i, y_i)_{i=1,\dots,N} \subseteq \mathbb{R}^d \times \mathbb{R}$ under regularization $\lambda > 0$ aims to optimize the objective

$$\min_{f \in \mathcal{H}} \frac{1}{N} \sum_{i=1}^{N} (y_i - f(\xi_i))^2 + \lambda \|f\|_{\mathcal{H}}^2.$$

Under vanishing regularization $\lambda \to 0$, the solution to kernel 'ridgeless' regression is given by the *minimum RKHS-norm interpolant (MNI)*

$$\hat{f}_{\text{MNI}} = \arg\min_{f \in \mathcal{H}} \|f\|_{\mathcal{H}} \quad \text{s.t.} \quad y_i = f(\xi_i) \;\; \forall i \in [N].$$

When the kernel matrix $\mathbf{K}$ is invertible, the MNI takes the explicit form

$$\hat{f}_{\text{MNI}}(\xi) = (k(\xi, \xi_1), \dots, k(\xi, \xi_n)) \cdot \mathbf{K}^{-1} y.$$

Note that ordinary least squares linear regression is just the special case when using the linear kernel $k(\xi_1, \xi_2) = \langle \xi_1, \xi_2 \rangle$ and using the Moore-Penrose pseudo inverse in the non-invertible case. Linear regression could also be seen as learning a 0-hidden layer MLP or learning with the identity feature map.

For more details, we refer to Appendix B in Chapter 4, Schölkopf and Smola (2001) and Steinwart and Christmann (2008). In particular, Mercer's theorem implies that a continuous kernel function on a compact domain together with a Borel probability measure is closely related to an orthonormal basis of $L_2(\mathbb{P}_X)$ and a decaying sequence of eigenvalues. The spectral decay properties of these eigenvalues have been very useful in understanding the generalization properties of kernel methods (see Barzilai and Shamir (2024) and references therein).

2.3 Infinite-width Limits of Neural Networks

In the infinite-width limit, the learning dynamics of certain classes of neural networks can become tractable. One important ansatz stems from the realization that most neural networks just consist of element-wise nonlinearities and matrix-vector multiplications between trainable weight tensors and incoming activations. Under weak assumptions, these growing sums will converge, for example following from versions of the central limit theorem or the law of large numbers. In remarkable generality, Greg Yang et al. (Yang, 2019, 2021; Yang and Hu, 2021; Yang and Littwin, 2023; Yang et al., 2023a,b) have established a compositional, non-linear framework for such random matrix theory results that describes neural network training dynamics at excessive width. Here we will introduce the basic results step by step, starting from initialization, going over neural network parameterizations where the training dynamics reduce to kernel regression with a deterministic kernel in the limit to parameterizations that preserve feature learning in the limit and hence induce more complex but practically useful width-independent dynamics.

2.3.1 Neural networks at initialization

Recall that weight matrices $W : \mathbb{R}^{\texttt{fan\_in}} \to \mathbb{R}^{\texttt{fan\_out}}$ are typically initialized as $W_{ij} \overset{iid}{\sim} \mathcal{N}(0, C/\texttt{fan\_in})$ (He et al., 2015; Glorot and Bengio, 2010). This initialization ensures that in the first forward pass, the variance in the activations x^l neither vanishes nor explodes with increasing width and/or depth, even though the sums in the matrix-vector multiplications $W^{l+1}x^l$ contain increasingly many terms. Conditioning on each layer's input, according to the Central Limit theorem, we can expect the entries of x^l to converge to a Gaussian distribution with mean and variance that can be iteratively computed when passing forward through the network. The rescaling constant C is usually chosen depending on the activation function to approximately preserve the feature variance in forward passes through arbitrarily deep networks at initialization. The ReLU function $\phi(z) := \max(0, z)$ just cuts off all negative activations. Therefore the variance of zero-mean Gaussian preactivations h^l will be cut in half, and the correction factor $C = 2$ rescales the activations $x^l = \phi(h^l)$ to have the same variance as those in the previous layer. In the infinite-width limit, the initial output logits converge to a draw from a Gaussian process called the Neural Network Gaussian Process (NNGP). The maps $\xi \mapsto x^L(\xi)$ as well as $\xi \mapsto \nabla_\theta f(\xi; \theta_0)$ can be seen as random feature maps that become deterministic in the infinite-width-limit due to the concentration of large sums of independent random variables.

More formally, we define the neural network Gaussian process (NNGP) kernel (Neal, 1996; Lee et al., 2018; Matthews et al., 2018; Hanin and Rolnick, 2018), writing f_0^i for the i-th coordinate of the output logits at initialization, as

$$\mathcal{K}^{ij}(\xi, \xi') := \lim_{n \to \infty} f_0^i(\xi) f_0^j(\xi').$$

Then, the initial function is drawn from the NNGP with that kernel function, that is, for a data matrix $\boldsymbol{X}$ it holds that $f_0(\boldsymbol{X}) \sim \mathcal{N}(0, \mathcal{K}(\boldsymbol{X}, \boldsymbol{X}))$. Conditioned on the

inputs $\xi, \xi' \in \mathbb{R}^{d_{in}}$, the first-layer NNGP kernel is given by $\Sigma^{(0)}(\xi, \xi') = \xi^\top \xi'$. The marginal distribution of (ξ, ξ') before layer $l \geq 1$ then has the covariance

$$\mathcal{P}^{(l-1)}(\xi, \xi') := \begin{pmatrix} \Sigma^{(l-1)}(\xi, \xi) & \Sigma^{(l-1)}(\xi, \xi') \\ \Sigma^{(l-1)}(\xi', \xi) & \Sigma^{(l-1)}(\xi, \xi) \end{pmatrix}.$$

The feature covariance after the layer is then given by

$$\Sigma^{(l)}(\xi, \xi') = C_\phi \cdot \mathbb{E}_{(u,v)\sim N(0, \mathcal{P}^{(l-1)}(\xi,\xi'))}[\phi(u)\phi(v)].$$

To give a concrete example, the NNGP of a 2-layer MLP with ReLU activation function for inputs restricted to the unit sphere is a dot-product kernel $k(\xi, \xi') = \kappa(\langle \xi, \xi' \rangle)$ given by $\kappa : [-1, 1] \to \mathbb{R}$ (Bietti and Bach, 2021),

$$\kappa(u) = \pi^{-1} \left(u \cdot (\pi - \arccos(u)) + \sqrt{1 - u^2} \right).$$

For L-layer MLPs, the NNGP is then simply given by iteration, $\kappa^{L-1}(\langle \xi, \xi' \rangle)$.

2.3.2 Training Wide Neural Networks in Popular Parameterizations

To move beyond initialization, we typically consider gradient-based optimization algorithms, exploiting the fact that evaluating the gradient with a backward pass is roughly only as computationally expensive as a function evaluation, also called forward pass. In this section, we discuss L-hidden layer MLPs trained with SGD for simplicity.

Random feature models. If only the last-layer weights W^{L+1} are trained, then the expectation of the function learned by gradient flow on a finite labeled training set with square loss converges to the posterior of the NNGP given the training set (Lee et al., 2019). When using ℓ_2-regularization, finite networks can also be seen as a random feature approximation to the NNGP kernel (Rahimi and Recht, 2008), which motivates its alternative name *random feature kernel*.

For full neural network training, it is crucial to first introduce the notion of a parameterization, defined as a collection of layerwise width-dependent exponents $\{a_l, b_l, c_l\}_{l\in[L+1]}$, that effectively prescribe layerwise initialization variances and layerwise learning rates.

Definition 6 (Parameterization). A L-hidden layer MLP with trainable weights $\{W^l\}_{l\in[L+1]}$ is parameterized in the *abc-parameterization* $\{a_l, b_l, c_l\}_{l\in[L+1]}$, if, for $l \in [L+1]$ and width-independent constants $\eta, \sigma_1, \ldots, \sigma_{L+1} \geq 0$, the weights W^l

1. are multiplied by width-dependent scalars $\alpha_l = n^{-a_l}$ in the forward pass,

$$h^1(\xi) := \alpha_1 \cdot W^1 \xi, \quad h^{l+1}(\xi) := \alpha_{l+1} \cdot W^{l+1} \phi(h^l(\xi)), \quad f(\xi) := \alpha_{L+1} \cdot W^{L+1} \phi(h^L(\xi)),$$

2. are initialized with i.i.d. coordinates $W^l_{ij} \sim \mathcal{N}(0, \sigma_l \cdot n^{-2b_l})$,

3. are updated with SGD or Adam using weight-specific learning rates $\eta_l = \eta \cdot n^{-c_l}$.

◀

Weight multipliers. Arbitrary layerwise update scalings can be achieved without layerwise learning rates by adapting the architecture by introducing width-dependent weight multipliers. Replacing the original weights W^l by $n^{-a_l}W^l$ rescales the gradient in each layer separately. At initialization, clearly the effective weights scale as $n^{-a_l}W^l \sim \mathcal{N}(0, n^{-2a_l-2b_l})$. For example, $W^l \sim \mathcal{N}(0, n^{-1})$ can be replaced by $n^{-1/2}W^l$ with $W^l \sim \mathcal{N}(0,1)$. The gradient $\frac{\partial \mathcal{L}}{\partial W^l} = \frac{\partial \mathcal{L}}{f}\frac{\partial f}{\partial W^l}$ is scaled by n^{-a_l}, and in the subsequent forward pass, the effect of the weight updates $\Delta W^l = \eta n^{-c_l}\nabla_{W^l}\mathcal{L}$ on the activations $h^l = n^{-a_l}(W^l + \eta n^{-c_l}\nabla_{W^l}\mathcal{L})x^{l-1}$ is scaled by an additional factor n^{-a_l}. For SGD, this results in the following layerwise equivalence between *abc*-parameterizations: For arbitrary $C \in \mathbb{R}$,

$$(a_l, b_l, c_l) \qquad \text{is equivalent to} \qquad (a_l + C, b_l - C, c_l - 2C). \tag{2.2}$$

For Adam, the gradient is normalized entrywise so that only the factor n^{-a_l} from the forward pass survives in the updates. This results in the layerwise equivalence: For arbitrary $C \in \mathbb{R}$,

$$(a_l, b_l, c_l) \qquad \text{is equivalent to} \qquad (a_l + C, b_l - C, c_l - C). \tag{2.3}$$

Feature learning and kernel regimes. Yang and Hu (2021) show that all stable *abc*-parameterizations are either feature learning or in a kernel regime. For defining these terms, we use Bachmann-Landau notation $\mathcal{O}, \Theta, \Omega$ that purely tracks dependence on width n and omits all other dependencies. Diverging logits or width-dependent activations have traditionally been associated with training instability, although our results in Chapter 6 question this intuition.

Definition 7 (Stability). We call an *abc*-parameterization *stable* if $f_t = \mathcal{O}(1)$ and $x_t^l = \Theta(1)$ for all layers $l \in [L]$ and at all times $t \geq 0$. Otherwise, we call the parameterization *unstable*. ◀

The following definition of feature learning merely provides a necessary condition for learning useful representations: The last-layer activations x_t^L should evolve non-trivially over the course of training. This does not imply that the learned representation also generalizes well, but as we will see, several common parameterizations do not even satisfy this necessary condition.

Definition 8 (Feature learning). We say that an *abc*-parameterization is in the *feature learning* regime if for every $t \in \mathbb{N}$ there exists a width-independent constant $c > 0$, a sequence of training points $\{(\xi_t, y_t)\}$, a learning rate $\eta > 0$ and input $\xi \in \mathbb{R}^{d_{\text{in}}}$ such that $\|x_t^L(\xi) - x_0^L(\xi)\|_{RMS} := n^{-1/2}\|x_t^L(\xi) - x_0^L(\xi)\|_2 \geq c$. ◀

When SGD reduces to kernel gradient descent, the learning dynamics become tractable. In other words, such parameterizations use a fixed feature map in the infinite-width limit and lose the feature learning that makes neural networks desirable in the first place. Hence, when the goal is understanding neural networks in practical settings, the kernel limit is a degenerate limit that is qualitatively different from neural networks at moderate width (Wenger et al., 2023).

Definition 9 (Kernel regime). We say that an *abc*-parameterization is in the *kernel regime*, if there exists a kernel function k such that, in the infinite-width limit $n \to \infty$,

			Weight-multiplier version			Weight-multiplier-free version		
			Input-like	Hidden-like	Output-like	Input-like	Hidden-like	Output-like
SP	$\alpha_l \cdot W^l$,	$\alpha_l \propto$				1	1	1
	$\mathcal{N}(0, \sigma_l^2)$,	$\sigma_l \propto$		-		1	$n^{-1/2}$	$n^{-1/2}$
	$\eta_l \cdot \nabla_{W^l} \mathcal{L}$,	$\eta_l \propto$				n^{-c}	n^{-c}	n^{-c}
NTP	$\alpha_l \cdot W^l$,	$\alpha_l \propto$	1	$n^{-1/2}$	$n^{-1/2}$	1	1	1
	$\mathcal{N}(0, \sigma_l^2)$,	$\sigma_l \propto$	1	1	1	1	$n^{-1/2}$	$n^{-1/2}$
	$\eta_l \cdot \nabla_{W^l} \mathcal{L}$,	$\eta_l \propto$	1	1	1	1	n^{-1}	n^{-1}
μP	$\alpha_l \cdot W^l$,	$\alpha_l \propto$	$n^{1/2}$	1	$n^{-1/2}$	1	1	1
	$\mathcal{N}(0, \sigma_l^2)$,	$\sigma_l \propto$	$n^{-1/2}$	$n^{-1/2}$	$n^{-1/2}$	1	$n^{-1/2}$	n^{-1}
	$\eta_l \cdot \nabla_{W^l} \mathcal{L}$,	$\eta_l \propto$	1	1	1	n	1	n^{-1}

Table 2.1: **(Common *abc*-parameterizations)** Here, we collect standard parameterization (SP), neural tangent parameterization (NTP) and the maximal update parameterization (μP) for SGD in their multiplier version which purely adapts the architecture and allows width-independent global learning rates (*left*) and in their weight multiplier-free version (*right*). Parameterizations differ in their layerwise choice of width-dependent weight multipliers α_l, initialization variances σ_l and learning rates η_l. Weight multiplier-free representatives of an *abc*-equivalence class purely adapt the optimization algorithm highlighting the fact that parameterizations effectively only induce layerwise learning rates. Knowing that μP correctly scales the updates in all layers, observe that the input- and hidden-layer learning rates in NTP induce vanishing updates. The same holds in SP when choosing $c \geq 1$ as is necessary for avoiding logit blowup in the infinite-width limit.

the output logits f_t evolve under a step of SGD with training point (ξ_t, y_t) and learning rate $\eta \geq 0$ according to

$$f_{t+1}(\xi) = f_t(\xi) - \eta \cdot k(\xi, \xi_t) \cdot \mathcal{L}'(f_t(\xi_t), y_t).$$

◀

While this unifying definition of *abc*-parameterizations was only made explicit by Yang and Hu (2021), different parameterizations had been previously used implicitly to enable well-defined infinite-width limits of neural networks (Neal, 1996; Jacot et al., 2018; Mei et al., 2018). Typically parameterizations have not used layerwise learning rates, but it is useful to understand that weight multipliers effectively serve the purpose of a relative reweighting of the weight updates in different layers. We summarize three common parameterizations in Table 2.1.

Standard Parameterization. *Standard parameterization (SP)* models the standard deep learning practice in computer vision and natural language processing. It neither uses weight multipliers nor layerwise learning rates. Assuming He or Glorot initialization, it is therefore defined as $a_l = 0$, $b_l = 0.5 \cdot \mathbb{I}(l > 1)$, $c_l = c$. This parameterization is common practice but rarely studied in infinite-width theory. Choosing $c = 1$ results in a well-defined kernel limit, but also does not capture the qualitative (feature learning) properties of practical networks trained at the optimal learning rate. In Section 3.3, we discuss in more detail that practical neural networks operate in the regime $c < 1$, and how we provide the first infinite-width proxy that captures SP with large learning rates in Chapter 6.

Neural Tangent Parameterization. The *neural tangent parameterization (NTP)*, popularized by Jacot et al. (2018), uses the weight multipliers $\frac{1}{\sqrt{\texttt{fan\_in}}}W$ while preserving standard initialization, which translates to $a_l = 0.5 \cdot \mathbb{I}(l > 1)$, $b_l = 0$, $c_l = 0$. This choice of weight multipliers prevents output blowup due to exploding last-layer updates under width-independent learning rates. The learning dynamics reduce to a kernel method in the infinite-width limit and have therefore been a popular choice for theoretical studies. This so called Neural Tangent Kernel is simply defined by the inner product between gradients, denoting the vectorized concatenation of all trainable weights as $\theta = (Vec(W^1), \ldots, Vec(W^{L+1}))$,

$$\hat{k}_{NTK,t}(\xi, \xi') = \langle \nabla_\theta f(\xi; \theta_t), \nabla_\theta f(\xi'; \theta_t) \rangle.$$

While this kernel is random and evolves over the course of training at finite width, Jacot et al. (2018) first showed that the NTK becomes deterministic and constant over the course of training in the infinite-width limit, $\hat{k}_{NTK,t} \to k_{NTK}$, enabling a detailed understanding of the full learning dynamics. The NTK is also useful for studying linearized neural network learning dynamics beyond the deterministic limit in NTP (Jeffares et al., 2024).

In NTP, at sufficient width gradient descent can find a global optimum while keeping weights so close to initialization that the linearization of the neural network around its initial parameters yields an accurate approximation of the learned function over the entire course of gradient flow training. Beyond the infinite-width limit, Arora et al. (2019) show that gradient flow $\dot{\theta}_t = -\nabla \mathcal{L}(\theta_t)$ on the MSE loss is equivalent to gradient flow in with the empirical/finite-width NTK $\hat{k}_{NTK,t}$, for $i \in [|\mathcal{D}|]$,

$$\frac{df(x_i; \theta_t)}{dt} = \sum_{j \in [|\mathcal{D}|]} \hat{k}_{NTK,t}(x_i, x_j)(f(x_j; \theta_t) - y_j). \tag{2.4}$$

In words, gradient flow in parameter space translates into gradient flow in function space on the residuals with the (empirical) NTK. Lee et al. (2019) provide an iterative computation rule for the NTK similar to the NNGP kernel but with an additional first-order term that stems from updating all layers,

$$k_{NTK}^{(l)}(x, x') = \Sigma^{(l)}(x, x') + k_{NTK}^{(l-1)}(x, x') \cdot \mathbb{E}_{(u,v) \sim N(0, \mathcal{P}^{(l-1)}(x,x'))}[\phi'(u)\phi'(v)].$$

Assuming $\lambda_{min}(k_{NTK}) > 0$ and $\|x\|_2 \leq 1$, Lee et al. (2019, Theorem 2.1) shows that gradient flow in NTP induces

$$\sup_{t \geq 0} \|f_t(x) - f_t^{lin}(x)\|_2, \ \sup_{t \geq 0} \frac{\|\theta_t - \theta_0\|_2}{\sqrt{n}}, \sup_{t > 0} \|\hat{k}_{NTK,t} - \hat{k}_{NTK,0}\| \leq \mathcal{O}(n^{-1/2}),$$

where $f_t^{lin}(x) = f(x, \theta_0) + \nabla_\theta f(x, \theta_0)(\theta - \theta_0)$ denotes the network's linearization around its initial parameters. Hence, over the entire course of gradient flow training the parameters and NTK stay close to initialization at sufficient width, while the output logits approximately evolve linearly in parameter space. Note, however, that this notation omits polynomial dependencies on the dataset size $|\mathcal{D}|$ that let finite networks quickly leave this kernel regime in practice. For gradient flow with MSE loss, the ODE (2.4) has a closed form solution, which converges to the minimum-norm interpolant with respect to the NTK in the limit of infinite width and training time

when $f_0 = 0$. Early stopping has a similar regularizing effect as kernel ridge regression with positive regularization (Yao et al., 2007). In Chapter 4, we provide theory for how well networks in NTP generalize when overfitting to label noise, and a discussion of our results in Section 3.1.

Maximal Update Parameterization. A more recent popular parameterization, the so called *Maximal Update Parameterization* (μP) (Yang and Hu, 2021), ensures a width-independent effect of the weight updates in each trainable weight matrix on the output logits. Up to smaller last-layer initialization, Yang and Hu (2021) show that μP is the unique parameterization with this property. Hence neither at initialization nor over the course of training does a weight induce a vanishing or exploding effect on the output logits, and training dynamics become fully width-independent (Vyas et al., 2024; Noci et al., 2024b; Bordelon and Pehlevan, 2025). As important practical benefits, the optimal learning rate transfers across scales in μP (Yang et al., 2022), so that it can be tuned on small models and the largest model only has to be trained once with optimal hyperparameters. By preserving maximal stable feature learning in all layers, increasing model size in μP is generally observed to improve performance monotonically with scale, as opposed to large models in SP. Overall, under sufficient numerical precision, μP promises more predictable, stable training of large models than SP or NTP.

This is accomplished by reducing the last-layer gradient as under NTP, but a relatively larger scaling of hidden and input layers. The derivation requires a careful signal propagation analysis both forward and backward to arrive at a unique choice that prevents both exploding as well as vanishing updates in all layers. We discuss the Tensor Program framework that was used to derive μP in Section 2.4. μP is defined as $a_l = 0.5 \cdot \mathbb{I}(l = L+1) - 0.5 \cdot \mathbb{I}(l = 1)$, $b_l = 1/2$, $c = 0$. Note that the last-layer initialization is smaller than in NTP and SP and induces $f_0 = \mathcal{O}(n^{-1/2})$. Instead of weight multipliers, one could equivalently use small last-layer learning rates $\eta_{L+1} = \eta \cdot n^{-1}$ and large input-layer learning rates $\eta_1 = \eta \cdot n$. In Vankadara et al. (2024), we show that modern structured state space models like Mamba (Gu and Dao, 2023) require different learning rate scalings. In Chapter 5, we prove that the minimax optimization algorithm *Sharpness Aware Minimization* (SAM) also requires non-trivial layerwise perturbation scaling to recover width-independent perturbations in all layers at large width, as discussed in Section 3.2.

Mean-field Parameterization. The *mean-field parameterization* was proposed by the statistical physics community and is equivalent to μP. Similar to NTP, it is typically formulated with weight multipliers $\frac{1}{\sqrt{\texttt{fan\_in}}}W$ and $W_{ij} \sim \mathcal{N}(0,1)$, except for the factor $\frac{1}{\texttt{fan\_in}}W$ in the last layer, that promotes feature learning akin to Chizat et al. (2019). To observe non-trivial learning at width-independent time, gradient flow is then run as $\dot{W} = -N \cdot \nabla_W L$. This corresponds to the choice $a_l = 0.5 \cdot \mathbb{I}(l \in [2, L]) + 1 \cdot \mathbb{I}(l = L+1)$, $b_l = 0$, $c_l = -1$, which is equivalent to μP according to the equivalence relation (2.2). Most mean-field theory only applies to shallow (meaning 2-layer) networks (Mei et al., 2018; Chizat and Bach, 2018), and a satisfactory, rigorous extension to deep networks remains an active area of research (Nguyen and Pham, 2023; Sirignano and Spiliopoulos, 2022). A crucial insight here is to use the exchangeability of neurons and

write the neural network forward pass as an integral over the empirical measure of trainable weights $\mu_n = n^{-1} \sum_{i_1}^{n} \delta_{w_i}$,

$$f(\xi) = \frac{1}{n} \sum_{i=1}^{n} \phi(w_i, \xi) = \int_{\mathbb{R}^p} \phi(w, \xi) d\mu_n(w).$$

Gradient flow on the trainable parameters then translates into Wasserstein gradient flow in the space $\mathcal{P}_2(\mathbb{R}^p)$ of probability measures with bounded second moments (Chizat and Bach, 2020).

The dynamical mean field theory (DMFT) popularized by Bordelon et al. (Agoritsas et al., 2018; Bordelon and Pehlevan, 2022; Bordelon et al., 2024c,b) relies on heuristic physics techniques but applies to more general architectures. It allows to approximate the infinite-width limit of gradient flow training by numerically solving closed-form expressions to compute the feature kernels throughout training. While this theory appears to provide accurate approximations of the training dynamics in polynomial instead of exponential time required for the exact solution, both the exact computation of these DMFT kernels as well as perturbative approximations are still very expensive with computational complexity $\mathcal{O}(T^3 \cdot |\mathcal{D}|^3)$ and $\mathcal{O}(T \cdot |\mathcal{D}|^4)$, respectively, for T steps of GD on $|\mathcal{D}|$ training points (Bordelon and Pehlevan, 2022). In future work, it would be valuable to further reduce this complexity by more refined computation and approximation algorithms.

Edge of stability under large learning rates. While gradient flow is well-understood in many settings, large learning rates often result in better generalization and qualitatively different learning dynamics. Hence, theory with the goal of describing practical neural networks needs to accurately capture this regime. One intriguing finding here is the *edge of stability* phenomenon (Cohen et al., 2020). Over the initial course of training, the sharpness, meaning the leading singular value of the loss Hessian, progressively increases to the edge of stability $2/\eta$, above which a step in the direction of maximal sharpness would result in loss divergence. But instead of increasing further, the sharpness fluctuates around this critical threshold for a long time, before it decreases again while converging to a minimum. This long period of instability allows the parameters to drift further from initialization than under gradient flow. While there exist studies that connect lower sharpness to better generalization, such statements do not hold robustly in modern architectures, and hence generalization under large learning rates remains poorly understood (Andriushchenko et al., 2023a). More details on edge of stability dynamics can be found in Cai et al. (2024) and Damian et al. (2023) and references therein.

2.4 Tensor Program Framework

A promising framework to understand width-scaling and limiting properties of neural networks during and after training is the Tensor Programs framework by Greg Yang et al (Yang, 2019, 2021; Yang and Hu, 2021; Yang and Littwin, 2023; Yang et al., 2023b). In its most general form that also covers optimization with Adam, it is called NE$\otimes$OR$\top$programs. The main idea is to provide a compositional form of random matrix theory that allows all common computations in neural network optimization like MLPs, ResNets or Transformers trained with SGD or Adam. The

allowed computation rules are modular so that layers can be flexibly added or removed, and the impact of width-scaling properties can be rigorously tracked, even in practical settings. The full formal statements can be found in the original publications. Here, we instead aim to provide a short, accessible introduction.

The three allowed computation rules are matrix-vector products, averaging and non-linear outer products. After writing forward and backward passes of standard architectures in terms of these computation rules, the NE⊗OR⊤master theorem shows that quantities of fixed length such as the neural network output converge to deterministic constants, whereas width-scaling vectors such as intermediate activations inside the network behave as if they had i.i.d. coordinates with a coordinate distribution that can also be calculated from the infinite-width counterparts of the finite-width computation rules.

When we are merely interested in the correct width-scaling, there are effectively only two types of behaviour at excessive width. For the example of matrix-vector multiplications with approximately i.i.d. coordinates, when both are sufficiently independent and one has zero mean, their multiplication will behave according to the central limit theorem, introducing a scaling factor $n^{1/2}$ with width n. When matrix and vector are correlated, their multiplication will behave according to the law of large numbers and introduce a scaling factor of n. For vectors $v \in \mathbb{R}^d$, the RMS-norm $\|v\|_{RMS} = d^{-1/2} \cdot \|v\|_2$ naturally measures its average entry size. Going through the entire neural network forward and backward pass, the width-dependent exponents of all terms can be tracked. For μP, layerwise learning rates are chosen such that activation and logit entry updates neither vanish nor explode with width, $\|\delta h_t^l\|_{RMS} = \Theta(1)$ and $\|\delta f_t\|_{RMS} = \Theta(1)$ for all $l \in [L]$ and fixed $t \in \mathbb{N}$. As a minimal example, consider the last-layer weight updates in the first step of training with cross-entropy loss,

$$\begin{aligned} \Delta W_1^{L+1} = -\eta_{L+1} \cdot \frac{\partial \mathcal{L}(f_0(\xi_0), y_0)}{\partial W^{L+1}} &= -\eta_{L+1} \cdot \frac{\partial \mathcal{L}(f_0(\xi_0), y_0)}{\partial f} \cdot \frac{\partial f_0(\xi_0)}{\partial W^{L+1}} \\ &= -\eta_{L+1} \cdot (\sigma(f_0(\xi_0)) - y_0) \cdot x_0^L(\xi_0), \end{aligned}$$

where $\|\sigma(f_0(\xi_0)) - y_0\|_{RMS} = \Theta(1)$ generically holds and $\|x_0^L(\xi_0)\|_{RMS} = \Theta(1)$, since the initialization is chosen to be stable. To quantify the effect of the weight updates on the logits, observe that logit updates are composed of effective last-layer updates and updates propagating forward from previous layers,

$$\delta f_1(\xi) = \delta W_1^{L+1} x_1^L(\xi) + W_0^{L+1} \delta x_1^L(\xi). \tag{2.5}$$

Since last-layer activations x_0^L and x_1^L generally have non-zero correlation, the effective last-layer update scales as $\delta W_1^{L+1} x_1^L(\xi) = \Theta(\eta_{L+1} \cdot x_0^L \cdot x_1^L(\xi)) = \Theta(\eta_{L+1} \cdot n)$. Width-independent logit updates through $\|\delta W_1^{L+1} x_1^L(\xi)\|_{RMS} = \Theta(1)$ are therefore achieved with the choice $\eta_{L+1} = \Theta(n^{-1})$. Concerning the propagating update term in (2.5), W_0^{L+1} and δx_1^L are generally correlated, as $\frac{\partial \mathcal{L}}{\partial W^L}$ explicitly contains W_0^{L+1}. Thus, achieving both feature learning $\delta x_1^L = \Theta(1)$ and logit stability $\delta f_1 = O(1)$ requires small last-layer initialization variance $\|W_0^{L+1}\|_{RMS} = O(n^{-1})$.

Similarly evaluating the effective update scalings in all remaining layers yields width-scaling rules for the learning rate of each weight matrix. In short, hidden layer gradients inherit an additional term $\frac{\partial f}{\partial h^l} = \Theta(n^{-1})$ under small last-layer initialization compared to the last layer, but accumulate the same n-factor, so that the learning rate

can be chosen width-independently $\eta_l = \Theta(1)$ for $l \in [2, L]$. The input layer only accumulates sums of fixed size but inherits the same small gradients $\frac{\partial f}{\partial h^1} = \Theta(n^{-1})$, so that input layer learning rates have to be increased by a factor $\eta_1 = \Theta(n)$. Overall, these layerwise learning rates together with small last-layer initialization results in the Maximal Update Parameterization (μP) for SGD. Instead of using layerwise learning rates, gradient scaling can also be adjusted by introducing width-dependent weight multipliers in the architecture. The choice of weight multipliers $a_l = 0.5 \cdot \mathbb{I}(l = L+1) - 0.5 \cdot \mathbb{I}(l = 1)$ has the advantage of not only allowing a naive global learning rate $\eta_n = \eta \cdot n^0$ but also scaling Hessian eigenvalues and Sharpness Aware Minimization perturbations width-independently. In other words, under these weight multipliers, width independence in parameter space translates into width independence in function space. A more detailed accessible introduction into the TP framework can be found in Appendix C in Chapter 6.

2.5 Practical Considerations for Neural Network Scaling

Here we present complementary perspectives on μP, provide a tutorial for practitioners that aims to facilitate understanding the desiderata for desirable width scaling, and make practical considerations transparent that facilitate their implementation.

Spectral perspective on μP. We are ultimately interested in the correct activation and logit update scalings, but can only control the layerwise weight initialization and learning rates. While Tensor Programs allow to track the transformations of vectors like activations, Yang et al. (2023a) provide an equivalent formulation in terms of weight matrix operator norms $\|W\|_{2\to 2}$ for $W : \mathbb{R}^{\texttt{fan\_in}} \to \mathbb{R}^{\texttt{fan\_out}}$. The condition assumes that weights act linearly on activations, which is the case in standard architectures. Since we are interested in average entry size of activation updates, we rewrite this condition in terms of the operator norm with respect to the RMS-norm in the input and output spaces of the weight matrices,

$$\|W\|_{RMS\to RMS} = \max_{v\in\mathbb{R}^{\texttt{fan\_in}}} \|Av\|_{RMS} / \|v\|_{RMS},$$

which simplifies the condition to,

$$\|W_0^l\|_{RMS\to RMS} \overset{!}{=} \Theta(1) \quad \text{and} \quad \|\Delta W_t^l\|_{RMS\to RMS} \overset{!}{=} \Theta(1),$$

at all times t. Intuitively, this condition simply states that correctly scaled incoming activations $\|x_t^{l-1}\|_{RMS} = \Theta(1)$ or activation updates $\|\delta x_t^{l+1}\|_{RMS} = \Theta(1)$ should be propagated forward through the layer such that their effect on the next layer remains width-independent. Clearly, this only holds when the weights and incoming activations are sufficiently aligned to not lose a width-dependent factor, $\|\Delta W_t^l x_t^l\|_{RMS} \overset{!}{=} \Theta(\|\Delta W_t^l\|_{RMS\to RMS} \cdot \|x_t^l\|_{RMS})$. The TP framework formally guarantees this alignment at extensive width, and these conditions can always be empirically validated. In Chapter 6, we do this and find that TP scaling predictions even hold after accumulating many update steps in practical Transformer training, showing that these scaling considerations are even practically useful.

The operator norm conditions on weight initialization and updates hold under initialization variance σ_l, SGD learning rate η_l and Adam learning rate η_l^{Adam} chosen as,

$$\sigma_l = \Theta\left(\frac{1}{\sqrt{\texttt{fan\_in}}} \min\left\{1, \sqrt{\frac{\texttt{fan\_out}}{\texttt{fan\_in}}}\right\}\right),$$
$$\eta_l = \Theta\left(\frac{\texttt{fan\_out}}{\texttt{fan\_in}}\right), \qquad \eta_l^{Adam} = \Theta\left(\frac{1}{\texttt{fan\_in}}\right).$$

This generalizes μP to varying widths inside the network. In a similar direction, Bernstein and Newhouse (2024) are incorporating composable norm considerations in practical neural network optimization by understanding gradients as dual vectors in a norm-constrained optimization problem induced by appropriate choices of norms on the input and output spaces. This has led to speed ups in LLM training (Jordan et al., 2024). Beyond scaling theory, norm considerations for neural networks have a rich history for generalization bounds (Bartlett, 1996; Neyshabur et al., 2015; Bartlett et al., 2017).

However, the spectral condition can also fail when its assumptions are violated. As an important example, the modern structured state space model Mamba (Gu and Dao, 2023) requires its own detailed signal propagation analysis due to its non-standard selection mechanism and structured HiPPO matrices (Vankadara et al., 2024).

Tutorial for practitioners on how to achieve width-independent μP-scaling. Here, we aim to make all steps transparent that are necessary to successfully implement μP for training standard architectures with Adam.

In practice, it has been found that it can be necessary to tune constant weight multipliers at base width at least in the embedding, attention and output layer, as well as Adam's hyperparameters, both for achieving competitive generalization performance as well as stable hyperparameter transfer of the optimal learning rate across model scales. This partially makes up for the fact that optimizers, architectures and hyperparameters have all been fine-tuned for standard initialization by the entire research community for years. When tuning constant weight multipliers $\alpha_l \cdot W_t^l$, all parameterizations are equivalent at some fixed base width (up to smaller last-layer initialization). Thus, μP is best understood as a rule of how to rescale hyperparameters in relation to such a base width. This also allows to exactly replicate standard neural networks at base width, for backward compatibility.

We now summarize the necessary steps for adapting an existing code base that trains MLPs, CNNs or Transformers with Adam to width-independent μP-scaling. We further explain these steps below. After implementing the following changes, it should be possible to change the width without having to adapt the learning rate:

1. Initialize the output layer weights to 0.
2. Rescale the Adam learning rate of a trainable weight W by the factor $\frac{\texttt{base\_fan\_in}}{\texttt{fan\_in}}$.
3. When training a Transformer, replace the key-query normalization $d_{kq}^{-1/2}$ by d_{kq}^{-1}.
4. Introduce a constant multiplier $\alpha > 0$ in at least the output layer $\alpha \cdot f_W(x)$.

5. Tune hyperparameters such as the learning rate, initialization variance and weight multipliers on a small proxy model.

Note that transfer of the optimal hyperparameters is not guaranteed when scaling other dimensions like depth or training time. The implementation by Blake et al. (2025) reduces the necessity of multiplier tuning by introducing alternative multipliers with less redundancies and by prioritizing unit scaling considerations. Note however that they increase input layer learning rates beyond μP. The original implementation provided by Yang et al. (2022) uses a width-dependent output multiplier n^{-1}, which lets them treat input-like and output-like weights in the same way, so that they only distinguish vector-like weights with one width-dependent dimension versus matrix-like weights with two width-scaling dimensions.

Step 1 reduces finite-width biases by exactly equalizing the initial function $f_0 = 0$ at finite and infinite width. For understanding step 2, note Adam's entrywise gradient normalization under sufficiently small ε greatly simplifies scaling considerations: The product $\Delta W^{l+1}x^l$ generally accumulates a LLN-like factor `fan_in`, since normalized gradients and incoming activations are correlated. As all activations should scale width-independently, the effective update scaling simplifies to $\|\Delta W^{l+1}x^l\|_{RMS} = \Theta(\eta_{l+1} \cdot \texttt{fan\_in})$, and the accumulation from the product is corrected by the inverse learning rate scaling $\eta_{l+1} = 1/\texttt{fan\_in}$. Similarly, for step 3, the key-query inner product is generally expected to accumulate a factor proportional to the length of the sum $\|k \cdot q\|_{RMS} = \Theta(\|k\|_{RMS} \cdot \|q\|_{RMS} \cdot d_{kq})$ with $k, q \in \mathbb{R}^{d_{kq}}$ after the first update step, so that independence with respect to the scaling dimension d_{kq} requires correcting with the inverse d_{kq}^{-1}, instead of $d_{kq}^{-1/2}$ which is only correct at initialization. Step 4 and 5 allows for adjusting the amount of feature learning at fixed width (Chizat et al., 2019), and weight multipliers more generally adjust the relative learning speed of all layers (Kunin et al., 2024).

Chapter 3

Thesis Contributions

This thesis is based on the following conference papers, where * denotes equal contribution,

> **Haas***, **Moritz**, David Holzmüller*, Ulrike von Luxburg, and Ingo Steinwart. "Mind the spikes: Benign overfitting of kernels and neural networks in fixed dimension." *Advances in Neural Information Processing Systems 36* (2023): 20763–20826. https://proceedings.neurips.cc/paper_files/paper/2023/hash/421f83663c02cdaec8c3c38337709989-Abstract-Conference.html

> **Haas, Moritz**, Jin Xu, Volkan Cevher, and Leena Chennuru Vankadara. "μP^2: Effective Sharpness Aware Minimization Requires Layerwise Perturbation Scaling." *Advances in Neural Information Processing Systems 37* (2024): 38888–38959. https://proceedings.neurips.cc/paper_files/paper/2024/hash/449a016a6ce6fba3fe50d05482abf836-Abstract-Conference.html

and the following submitted preprint,

> **Haas, Moritz**, Sebastian Bordt, Ulrike von Luxburg, and Leena Chennuru Vankadara. "On the Surprising Effectiveness of Large Learning Rates under Standard Width Scaling." *Submitted to The Thirty-ninth Annual Conference on Neural Information Processing Systems* (2025). https://arxiv.org/pdf/2505.22491

Additional publications not included in this thesis. In addition to the above papers, I published the following journal paper on the statistical pitfalls of graph construction methods for climate data analysis, which is not included in this thesis because it treats questions unrelated to neural network scaling.

Haas, Moritz, Bedartha Goswami, and Ulrike von Luxburg. "Pitfalls of Climate Network Construction—A Statistical Perspective." *Journal of Climate 36, no. 10* (2023): 3321-3342. https://journals.ametsoc.org/view/journals/clim/36/10/JCLI-D-22-0549.1.xml?tab_body=pdf

During my time as a PhD student, I have also co-authored two more papers:

The following paper shows how to set the initialization variance and learning rate of each learnable weight tensor in modern structured state space models such as Mamba to achieve width-independent signal propagation, in both forward and backward passes throughout training.

Vankadara*, Leena Chennuru, Jin Xu*, **Moritz Haas**, and Volkan Cevher. "On Feature Learning in Structured State Space Models." *Advances in Neural Information Processing Systems 37* (2024): 86145-86179. https://proceedings.neurips.cc/paper_files/paper/2024/hash/9c7eeda2dc98e61baa9a5884afd231bc-Abstract-Conference.html

Finally, I co-authored the following submitted preprint. It studies the evolution equations of 2-layer networks in μP in more detail and proves that incorrect width-dependent scaling of Adam's ε or SAM's ρ introduces width dependent dynamics.

Zhu, Zhenyu, Mikolaj Boronski, **Moritz Haas**, Volkan Cevher, and Leena Chennuru Vankadara. "On the Role of Parameterization in the Dynamics of SAM and Adam." *Submitted to The Thirty-ninth Annual Conference on Neural Information Processing Systems* (2025).

I now summarize the contributions I made in the three publications in this thesis and embed them in the field of deep learning theory.

3.1 Benign Overfitting of Kernels and Extensively Wide Neural Networks

Leading question. Often huge overparameterized neural networks are trained until they nearly interpolate noisy training data, and often they still generalize near-optimally. This is surprising from a statistical perspective and raises the question:

When and why can kernel or neural network models that overfit noisy training data generalize nearly optimally?

3.1.1 Prior State of the Literature

The phenomena of *benign overfitting* and *double descent* (Figure 3.1) have surprised classical statisticians, as they stand in conflict with the classically dominant intuition

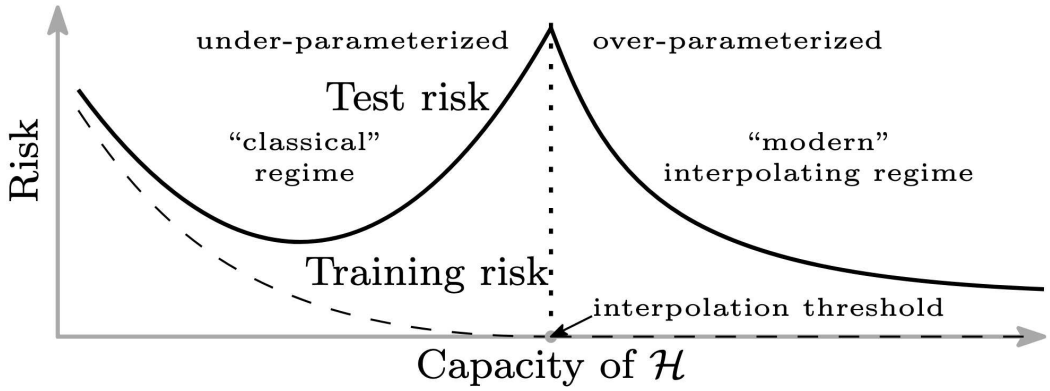


Figure 3.1: **Double Descent.** Schematic illustration of training risk (dashed line) and test risk (solid line) when capacity is measured with, for example, number of trainable parameters. The training risk monotonically decreases with increased capacity and the training set can be perfectly interpolated in the overparameterized regime. The test risk follows a classical u-shaped curve in the under-parameterized regime, but monotonically improves with increasing capacity in the overparameterized regime. This figure has been adapted from Belkin et al. (2019a).

of a u-shaped generalization curve (Hastie et al., 2022). In standard textbooks like "Elements of statistical learning" (Hastie et al., 2009, Section 2.9), statisticians were told:

> *"However with too much fitting, the model adapts itself too closely to the training data, and will not generalize well (i.e., have large test error)."*

According to this classical intuition the bias-variance trade-off induces a *u*-shaped curve: A model class should be rich enough to roughly capture the structure in the data to mitigate bias, but simple enough to avoid overfitting to the noise in the training data to mitigate variance. Hence, function classes were designed to find the sweet-spot of intermediate complexity, while not being able to perfectly interpolate the training data. This intuition was backed by uniform convergence upper bounds which guaranteed the test error to be well-approximated by the training error *on the entire function class*. But such generalization bounds can clearly only explain good generalization in cases where overfitting is avoided. On the other hand, they do not rule out the possibility that even estimators that interpolate noisy training data may generalize well.

In the underparameterized regime, the u-shaped generalization curve persists, but when increasing the number of parameters in, for example neural networks, beyond the point at which a potentially noisy training set can be perfectly interpolated, the generalization monotonically improves again and sometimes attains its optimum in the infinitely overparameterized limit. Zhang et al. (2021), Wyner et al. (2017) and Belkin et al. (2019a) drew significant attention to this phenomenon in the context of neural networks, random feature models, random forests and AdaBoost. However, the first observations of the double descent phenomenon were made much earlier in the statistical physics community. According to Loog et al. (2020), empirically Vallet et al. (1989) and theoretically Opper et al. (1990) first reported the double descent curve for binary classification with linear minimum-norm interpolants. While the implications of

the findings by Vallet et al. (1989) had not been sufficiently appreciated for over 25 years due to a lack of interconnectedness between the physical and statistical disciplines, the essential theoretical question that the benign overfitting phenomenon has motivated in the learning theory community since around 2017 is:

When and why can estimators that interpolate noisy training data still generalize well?

First, it is important to realize that, even in settings where double descent occurs, the bias-variance decomposition remains valid and useful. The interpolation peak only highlights the fact that understanding why the variance decreases in the overparameterized regime requires more estimator-specific analyses beyond classical uniform convergence bounds. Fully resolving the question clearly requires considering the data, the architecture and the optimization algorithm jointly (Zdeborová, 2020). Significant progress has been made in each of the individual components. Estimators from complex function classes can generalize well, when the effective class of functions that are learned with high probability has desirable structure. When the neural network architecture has *inductive biases* related to the data structure (Mitchell, 1980; Battaglia et al., 2018) or when the optimization algorithm induces *implicit regularization* (Soudry et al., 2018; Arora et al., 2019). This line of work suggests that plotting the double descent curve as a function of the number of parameters is misleading, as the mere number of parameters does not measure the effective complexity of the learned functions. Curth et al. (2024) observe that, at the interpolation peak, the mechanism by which the number of parameters is increased often changes. For example switching from deepening a single tree to bagging over more trees in Belkin et al. (2019a). Curth et al. (2024) instead resort back to (Hastie et al., 2009, Chapter 7.6) and propose to measure the effective number of parameters of smoothers and show that with such an appropriate complexity measure the double descent curve folds back in, resolving the initial surprise. Unfortunately many modern estimators, including deep neural networks trained with cross-entropy loss depend non-linearly on the labels y, so that it remains unclear how to measure effective complexity in general. While there are attempts at finding useful generalization bounds for neural networks (Bartlett et al., 2017; Neyshabur et al., 2017; Liang et al., 2019; Kawaguchi et al., 2022; Dziugaite and Roy, 2017; Zhou et al., 2019), this question remains largely open.

A related natural question that benign overfitting raises is:

Does the recent literature suggest that overfitting is desirable?

Indeed in some settings, interpolation is optimal in linear regression (Kobak et al., 2020; Tsigler and Bartlett, 2023) (where the optimal regularization can even be negative), kernel regression (Liang and Rakhlin, 2020) and random feature regression (Simon et al., 2024). But often, the optimal regularization is positive and removes the error peak around the interpolation threshold Hastie et al. (2022, Figures 1 and 9). The same phenomenon has therefore also been called *harmless interpolation* (Muthukumar et al., 2020; Mcrae et al., 2022), emphasizing that while overfitting does not necessarily hurt it also does not improve generalization. For minimum norm interpolants in linear and kernel regression in high dimension, it has been shown that the first principal components learn the signal and are responsible for near-optimal generalization, while the remaining dimensions can help to interpolate all training points without harming generalization (Bartlett et al., 2021). Intuitively, many similarly unimportant noise

directions can spread noise in spikes with vanishing volume, making it unlikely that an independent test point will lie in such spikes in the noise dimensions and effectively approximating the zero function (Tsigler and Bartlett, 2023). Effectively such noise dimensions add ridge regularization and hence do not improve upon explicit ridge regularization.

Benign overfitting in classification. Benign overfitting occurs much more generically in classification than in regression (Shamir, 2022). There exist regimes where the same minimum-norm interpolant generalizes well when measured in 0-1-loss, but not in mean squared error (Muthukumar et al., 2021). While regression requires estimating a correct value, classification merely requires learning the correct sign. Crucially, Friedman (1997) shows that the recovered part of the signal should be large compared to the variance while potentially allowing large bias. This has been used for proving faster non-asymptotic rates for classification versus regression (Audibert and Tsybakov, 2007; Koltchinskii and Beznosova, 2005; Devroye et al., 2013). Shamir (2022) shows that, for regression, benign overfitting rarely occurs beyond the well-specified settings from Bartlett et al. (2020), as benign overfitting on one task precludes its existence on another. For classification on the other hand, it is shown that benign overfitting can be expected in cases in which the squared hinge loss is a good surrogate for the missclassification error. In this thesis, we focus on benign overfitting in regression, as that is strictly harder to achieve.

Benign overfitting in kernel regression and wide neural networks. Often generalization performance of neural networks improves with model scale (Kaplan et al., 2020). This motivates considering studying their properties at extensive width. Networks in NTP are in the kernel regime at width polynomial in the dataset size (Arora et al., 2019).

Under very particular spectral decay of the covariance structure $\lambda_k \propto k^{-1}\log^{\alpha}(k)$, $\alpha > 1$, that is just fast enough to induce finite variance, benign overfitting also occurs in infinite-dimensional linear regression (Bartlett et al., 2020). We are not aware of an analytical expression of a kernel that would induce such a spectral decay. Even if such a kernel were found, the convergence rates may be very bad, as opposed to our results that achieve minimax-optimal convergence rates under weak assumptions. After our work, Barzilai and Shamir (2024) have derived unifying spectral bounds for kernel regression, and show that the strong Gaussian, independent feature assumptions in non-rigorous statistical physics papers (Mallinar et al., 2022) or those on linear models (Bartlett et al., 2020) result in overly optimistic variance bounds for kernel regression, and are hence uninformative in practical settings. What matters and is worse for common kernels is what they call *concentration coefficient*

$$\rho_{k,n} = \frac{\|\Sigma_{>k}\| + \lambda_1(\frac{1}{n}K) + \gamma_n}{\lambda_n(\frac{1}{n}K) + \gamma_n},$$

that quantifies how much the tail-component of the kernel matrix $K_{>k}$ differs from the exact effective ridge $\gamma\mathbb{I}$. Intuitively, their results suggest that benign overfitting can be expected when few principal eigendirections $\mu_i(\frac{1}{n}K)$ of $\frac{1}{n}K$ are large and they approximate the corresponding population quantities λ_i of the kernel, and when the tail eigenvalues concentrate well and result in similar size γ_n much larger than $\lambda_i = \mathcal{O}(1/i)$. This very much resembles the intuition for linear regression by Bartlett

et al. (2020), but crucially common kernels have worse concentration properties and hence result in worse generalization in realistic settings.

Before our work, kernel regression in fixed dimension seemed to generically overfit harmfully (Rakhlin and Zhai, 2019; Buchholz, 2022). Hence recent works have mostly considered high-dimensional limits $d \to \infty$, using tools from random matrix theory. In such high-dimensional limits, there exist fundamental approximation-theoretic limitations that require a bounded $\mathcal{H}$-norm assumption (Ghorbani et al., 2021; Donhauser et al., 2021). In other words, rotation invariant kernels can only learn a very limited class of functions, and benign overfitting will only occur under favorable distributional assumptions. For example, Montanari and Zhong (2022); Ghorbani et al. (2021) show that random feature and neural tangent kernel regression effectively perform ridge regression with polynomial features up to degree $l \in \mathbb{N}$, when the number of observations is large compared to d^l but small compared to d^{l+1}. Thus, without extensively many observations, the minimum RKHS-norm interpolant will only learn a low-degree polynomial part of the target function. Liang et al. (2020) observe and prove a related multiple descent phenomenon with peaks when $d \approx n^\alpha$ with $\alpha = 1/l$ for every $l \in \mathbb{N}$, and better generalization in between these peaks. Intuitively, at these peaks, degree-l polynomial parts start to be fitted, but initially with high variance. Therefore the best rates are achieved in between the peaks at $d = n^{1/(l+1/2)}$. Technically, a restricted lower isometry property is shown for truncated Taylor expansions of high-dimensional dot-product kernels.

3.1.2 Summary and Contributions

We consider the classical kernel regression setting where an increasing amount of data points is drawn i.i.d. from the same underlying distribution with noisy labels $\mathrm{Var}(y|x) \geq \varepsilon > 0$. Previous work (Rakhlin and Zhai, 2019; Buchholz, 2022) had suggested that overfitting is harmful in this setting, by showing that minimum norm interpolants (MNI) of classical kernels such as the Laplace kernel with RKHS equivalent to some Sobolev space do not converge to the optimal regression function.

Our work in Chapter 4 uncovers a more nuanced picture. First, we extend the previous inconsistency results separately to all estimators with RKHS norm comparable to that of the MNI that overfit by at least a constant fraction, to more general noise distributions, and to more kernels, including modern neural kernels like (deep) ReLU NTKs and NNGPs on the sphere. This essentially shows that overfitting with all common kernel estimators such as kernel ridge regression or gradient flow training will not generalize well in fixed dimension under weak distributional assumptions. Intuitively, standard smooth inductive biases will accumulate too much error around noisy training points. But we also show that by constructing estimators with a correctly balanced spiky-smooth inductive bias, akin to Wyner et al. (2017) or Belkin et al. (2019b), MNIs of spiky-smooth kernel sequences as well as wide neural networks with adapted activation functions can be consistent with minimax-optimal convergence rates under weak distributional assumptions. Intuitively, by adding a sharp spike of height $\lambda > 0$ to the kernel function, we mimick kernel ridge regression with regularization λ on most of the support, while interpolating the noisy training points. Remarkably, adding a Gaussian kernel with small bandwidth to the NTK or to the NNGP approximately translates into adding a high-frequency low-amplitude shifted sin-curve to the activation function. The additive structure of the spiky-smooth

kernels translates into approximate additivity in the NTK activation functions, so that the smooth regularized component and the spiky component of the learned function can be disentangled.

Overall our work shows that harmful overfitting with common estimators is a generic phenomenon for kernel regression, but that, with the right choice of estimator, overfitting is neither intrinsically helpful nor harmful for generalization in arbitrary dimension.

3.2 Width-independent Training Dynamics for Sharpness Aware Minimization

Leading question. Sharpness Aware Minimization (SAM) and increasing model scale have been observed to improve generalization across datasets and architectures (Chen et al., 2021; Kaddour et al., 2022; Kaplan et al., 2020; Yang et al., 2022). But before our work, the scaling properties of SAM were poorly understood. Our main question can be summarized as:

Can we understand how to optimally scale SAM training with width in a principled way?

3.2.1 Prior State of the Literature

Sharpness Aware Minimization. Sharpness Aware Minimization (SAM) (Foret et al., 2021) and its variants (Kwon et al., 2021; Müller et al., 2024) have been observed to consistently improve generalization across vision datasets and architectures (Chen et al., 2021; Kaddour et al., 2022). SAM is motivated through a minimax optimization objective within a ball of perturbation radius ρ, which provably reduces properties of the Hessian that are related to sharpness in simple settings (Bartlett et al., 2023; Wen et al., 2023; Monzio Compagnoni et al., 2023). The optimization algorithm evaluates the weight gradients $\nabla_W \mathcal{L}$ not on the current weights, but first perturbs the current weights upward the gradient direction before evaluating the gradient on these perturbed weights. This perturbed gradient can then be plugged into any base optimizer of interest like SGD or Adam. In this way, SAM can be seen as an orthogonal addition to the optimizer for improving generalization. A systematic understanding of why SAM improves generalization in modern architectures remains elusive. Correlations between sharpness and generalization have been questioned (Andriushchenko et al., 2023a; Wen et al., 2024) and perturbing only the normalization layers seems to suffice, despite increasing sharpness (Müller et al., 2024). SAM's main drawbacks are a doubled computation cost per update and an additional hyperparameter that needs to be tuned: the perturbation radius. Many variants of SAM have recently been proposed with the purpose of reducing SAM's computational and memory complexity or further improving generalization. We consider two variants of Adaptive SAM (Kwon et al., 2021) which are motivated by sharpness measures that are invariant to parameter rescaling symmetries and achieve the strongest results in the independent evaluation by Müller et al. (2024). We do not make direct statements about generalization. Instead, our goal is to achieve improved generalization and computational efficiency indirectly through width scaling desiderata akin to μP, which preserves maximal stable feature learning across widths.

Signal propagation desiderata. Signal propagation theory suggests that initialization scaling should be chosen to preserve width-independent and depth-independent variance in the activations in the first forward pass. This has been established by Glorot and Bengio (2010) and He et al. (2015), and has been widely accepted by the deep learning community. The main practical contribution of μP is its correction of signal propagation not only at initialization but to also correctly scale and balance the updates of all trainable weights. Width-independent update scaling entails several practical benefits. Fully width-independent training dynamics do not only enable HP transfer across model scales, but also preserve feature learning and hence improved generalization at large scale (Yang et al., 2022). Bordelon et al. (2024a) indeed quantify that feature learning networks obtain improved scaling exponents in test loss given power law structured teacher data. Both vanishing and diverging effects of weight updates on the logits potentially harm predictability across model scales. Additionally, diverging updates potentially induce training instability, and vanishing updates induce weaker feature learning with increasing width. Hence we consider width-independent effects of weight updates in each layer on output logits to be a reasonable desideratum for optimal width-scaling, both for optimal generalization at scale as well as for predictability across scales. While networks in NTP have been observed to require extensive width to approach their kernel limit (Wenger et al., 2023), the infinite-width limit of SGD and Adam in μP has generally been observed to accurately approximate finite-width networks in μP, over the course of training (Vyas et al., 2024; Noci et al., 2024b; Bordelon et al., 2024b; Bordelon and Pehlevan, 2025).

The formal derivation of μP is based on the Tensor Program framework, that we discuss in Section 2.4, which serves as a general framework for studying width-scaling in practical neural networks. It covers many common architectures like MLPs, convolutional neural networks and Transformers. For all of these architectures, maximal stable feature learning is achieved by the same μP initialization and learning rate scalings. This motivates the question:

Is μP a one-fits-all solution to neural network width scaling?

There are two aspects to this question. The first asks whether width-independent dynamics are always desirable, the second one asks whether width independence is always induced by the same layerwise initialization variance and learning rate scalings. For addressing these questions, we consider non-standard structured state-space model architectures like Mamba (Gu and Dao, 2023) in Vankadara et al. (2024), and non-standard optimization algorithms like Sharpness Aware Minimization (SAM) in Chapter 5. In this thesis, we focus on the latter.

Depth scaling. For scaling neural networks to infinite depth, residual connections have been found to be beneficial for stabilizing signal propagation while retaining expressivity. Without such residual connections, non-linearities in the network are only allowed to be perturbations of the identity that vanish with increasing depth for recovering a non-degenerate feature covariance (Hanin and Zlokapa, 2024; Li et al., 2022). Similarly to width scaling desiderata, the goal here is to retain a depth-independent effect of the increasing number of residual blocks on the output logits. The simple $\frac{1}{\sqrt{L}}$-scaling allows depth-scaling in ResNets where each block consists of a single layer, and unlocks hyperparameter transfer across depths (Hayou et al., 2021; Li et al., 2021; Bordelon et al., 2024c; Yang et al., 2023b). In this case, at initialization,

the infinite width and depth limits commute (Hayou and Yang, 2023). Noci et al. (2022, 2024a) provide infinite width and depth analyses for Transformers with the goal of preventing rank collapse and attaining a limit that has behaviour consistent with that of moderately large networks. Bordelon et al. (2024b) find that attention blocks in Transformers require the scaling L^{-1} for non-trivial feature learning in the infinite-depth limit under fixed width. Dey et al. (2025) validate this scaling in large-scale Transformer experiments. We expect SAM to require the same depth-scaling as SGD as each residual stream should still contribute the same update and perturbation size to the output.

3.2.2 Summary and Contributions

As discussed above, we consider optimal width scaling to mean fully width-independent training dynamics, and subsequently we evaluate empirically whether the optimality and predictability benefits observed under SGD and Adam are also induced by fully width-independent SAM training. But while μP achieves width-independent dynamics by correcting layerwise update scalings for SGD and Adam, SAM involves an additional intermediate perturbation step that potentially induces width dependence. Indeed, we show that in μP a global perturbation radius is not transferred across model scales and SAM's improved generalization is lost in MLPs, because effectively only the last layer is perturbed and the perturbations in all other layers vanish with width. Thus, recovering non-trivial perturbations in these earlier layers requires layerwise perturbation scaling. Hence, we characterize SAM training with arbitrary layerwise initialization variance, learning rate and perturbation scaling parameterizations in the infinite-width limit into four perturbation regimes: unstable, vanishing, non-trivial and effective perturbations in all layers. We find that there is a unique stable choice of layerwise perturbation radii that both effectively updates and effectively perturbs all layers, and hence induces fully width-independent training dynamics. We call this parameterization the *Maximal Update and Perturbation Parameterization* (μP^2). In MLPs and ResNets on CIFAR-10 and Vision Transformers on Imagenet-1K, we find that SAM training in μP^2 jointly transfers the optimal learning rate and perturbation radius from small to large scales, improves generalization and training stability over μP with a global perturbation radius. For finding the correct layerwise scaling for other gradient-based perturbation rules, a crucial insight is that gradients are generally low rank and correlated with the incoming activations, so that the average entry size in weight updates and weight perturbations should always share the same width-scaling exponent in each layer. We show that this heuristic even induces joint HP transfer in (η, ρ) for elementwise Adaptive SAM, which can not formally be written as a Tensor Program. For the proof for standard SAM, we write out all computations over the course of SAM training using the Tensor Programs framework. In addition to the forward and backward pass for weight updates, this requires an intermediate forward and backward pass for the weight perturbations. The joint gradient normalization typically implemented in practice (see the GitHub repositories provided by Foret et al., 2021; Kwon et al., 2021; Andriushchenko and Flammarion, 2022; Müller et al., 2024) and shown to speed up escaping saddle points (Monzio Compagnoni et al., 2023), complicates the analysis by coupling all layers. Ishikawa and Karakida (2024) analyse width scaling of the second-order optimization algorithms K-FAC and Shampoo and arrive at similar conclusions: The additional

damping hyperparameters in the preconditioning matrices require a particular width-dependent scaling in order to induce width-independent activation updates and HP transfer. In Vankadara et al. (2024), we find that achieving maximal stable feature learning in modern structured state space models such as Mamba requires a width-scaling rule that differs from μP in standard architectures. Due to the structured HiPPO matrices in the architecture, its derivation requires a detailed forward and backward signal propagation analysis that necessarily goes beyond the Tensor Program framework.

Overall, our work reinforces the practical desirability of width-independent training dynamics for optimality and predictability properties at scale across architectures, optimizers and datasets. It underlines the predictiveness of the Tensor Program framework for width dependent exponents of activation and logit updates in standard architectures even in practical settings, and its usefulness for deriving layerwise hyperparameter scaling rules that achieve width-independent dynamics. But we also show that the concrete initialization variance and learning rate scaling rule that achieves width-independent dynamics is not a one-fits all solution. Different optimization algorithms require their own detailed signal propagation analysis. For SAM, the additional perturbation radius induces width dependence without the correct layerwise rescaling. SAM in μP^2 not only recovers the bonus of HP transfer across scales, but layerwise perturbation scaling is necessary to recover SAM's beneficial implicit bias, even when allowing to tune hyperparameters at each scale.

3.3 Understanding the Effectiveness of Standard Width Scaling

Leading question. Existing infinite-width theory suggests that standard parameterization (SP) enters a kernel regime under small learning rates $\eta_n = O(n^{-1})$ and that output logits and the NTK diverge under larger learning rates $\eta_n = \omega(n^{-1})$ in the infinite-width limit, which may cause training instability (Sohl-Dickstein et al., 2020; Yang and Hu, 2021). In our experiments, however, we consistently observe slowly decaying optimal learning rates around $\eta_n \approx \Theta(n^{-1/2})$ in deep networks trained with SGD, and feature learning is preserved even at large width. This raises the question:

Can infinite-width theory be useful for understanding finite neural networks as they are initialized and trained in practice?

3.3.1 Prior State of the Literature

The dominant practice of training large scale neural networks is standard parameterization (SP) which we define as He initialization (He et al., 2015) and a single global learning rate for all trainable parameters tuned at every model scale, without width-dependent weight multipliers. The speed at which the learning rate decays then determines the qualitative behaviour in the infinite-width limit. Before our work, small learning rates $\eta_n = O(n^{-1})$ have been shown to induce vanishing feature learning similar to NTP, and larger learning rates $\eta_n = \omega(n^{-1})$ have been mostly thought to induce instability in the infinite-width limit (Yang and Hu, 2021; Sohl-Dickstein et al., 2020).

Infinite-width theory. A primary goal of infinite-width theory has always been developing a tractable model for understanding practical neural networks. First studies showed convergence of (deep) neural networks at initialization to the Neural Network Gaussian Process (NNGP) with increasing width (Neal, 1996; Daniely et al., 2016; Matthews et al., 2018; Lee et al., 2018). But this model does not sufficiently account for the fact that the weights in all layers evolve over the course of training. A second important step was the finding by Jacot et al. (2018) that the entire training dynamics evolve as a function of a deterministic kernel, they called the Neural Tangent Kernel (NTK). However, the authors acknowledge that this result is only possible due to a non-standard choice of weight multipliers, we introduce as the neural tangent parameterization (NTP) in Section 2.3.2. In NTP, Jacot et al. (2018) and Lee et al. (2019) show that this NTK is fixed over the entire course of training, and the training dynamics are well-approximated by the model linearized around the initial parameters at sufficient width. But this result can also be seen as a negative result, when the goal is understanding practical neural networks trained to convergence: While updates of the input and hidden layers non-vanishingly influence the learned function in NTP, input and hidden layer activations are updated to a vanishing degree and the feature learning that makes neural networks so attractive in the first place is lost in the infinite-width limit. Consequently, particularly convolutional networks have been observed to perform better than their limiting NTK solution (Lee et al., 2020; Novak et al., 2019; Arora et al., 2019) as the finite NTK evolves early in training (Fort et al., 2020), and the loss scaling exponents with dataset size are often better in finite networks (Vyas et al., 2022). On the other hand, when the data structure aligns with the NTK, Ortiz-Jiménez et al. (2021) show that the infinite-width limit can perform better than finite networks, which further reinforces that finite networks significantly differ from their kernel limit. Yang and Hu (2021) characterize the more general class of *abc*-parameterizations and find a unique parameterization μP (up to smaller last-layer initialization) that preserves stable, non-vanishing feature learning in all layers. As discussed in the previous section, this parameterization possesses several practically desirable properties, but this limit is again qualitatively different from the limit of neural networks as they are initialized and trained in practice. In SP, irrespective of the choice of learning rate, input layer updates affect the output logits with a smaller width-dependent exponent than hidden layer updates under both SGD and Adam. In μP in contrast, the updates of all layers are balanced as a function of width. This indicates that the limit in μP is also not ideal for understanding practical neural networks in SP. Most previous work however focuses on stable parameterizations in the sense that activations remain width-independent over the course of training and output logits remain bounded with increasing width. Most notably, Golikov (2020) has first studied the infinite-width limit of parameterizations with diverging logits under CE loss, showing that training does not necessarily diverge in this regime. They propose a parameterization called 'sym-default' as a more theoretically tractable proxy model for SP that is again not equivalent to SP with large learning rates. In addition, they only consider 2-layer networks and assume for simplicity that the gradients for the different weight matrices are estimated using different inputs, which is not done in practice.

Large learning rate dynamics at finite width. In stark contrast to existing infinite-width theory, we often observe that optimal learning rates in SP decay slowly with exponents around $\eta_n = \Theta(n^{-1/2})$ in our experiments of training deep networks with

SGD in SP. Much attention has been devoted to studying the training dynamics of finite networks under large learning rates. It has been widely observed that while small learning rates induce monotonic convergence to a local minimum, large learning rates induce better generalization after sufficiently long training. Cohen et al. (2021, 2022) observe that early in training sharpness, meaning the leading eigenvalue of the loss Hessian, typically increases and then hovers just above the edge of stability threshold $2/\eta$ above which the classical the descent lemma (Nesterov et al., 2018) does not guarantee loss decrease. In this second stage, the loss oscillates on short timescales but continually decreases over long timescales. Over the course of long training, the trainable parameters escape the initial region and drift toward a region of lower sharpness. Similarly, Lewkowycz et al. (2020) describe a catapult effect at large learning rates. Specifically they show for a 2-layer linear network in NTP that for learning rates above the edge of stability $2/\eta$ just below the instability threshold where training diverges, the loss increases in the first steps of training, while the sharpness decreases. Once the sharpness lies below the edge of stability $2/\eta$, training converges monotonically. While lower sharpness has often been connected to better generalization (Hochreiter and Schmidhuber, 1997b; Barrett and Dherin, 2021; Tsuzuku et al., 2020; Bartlett et al., 2023), it does not entirely explain the benefits of large learning rates, as there exist counterexamples, even when accounting for reparameterization symmetries (Dinh et al., 2017; Andriushchenko et al., 2023a; Müller et al., 2024). Other proposed partial explanations for the superiority of large learning rates include a different order in which patterns are learned (Li et al., 2019), enhanced SGD noise (Keskar et al., 2017), and implicit biases toward sparse solutions (Andriushchenko et al., 2023b; Qiao et al., 2024).

Chizat et al. (2019) and Woodworth et al. (2020) identify the output multiplier as an important hyperparameter for controlling the amount of feature learning. Kunin et al. (2024) study arbitrary relative scale of initialization variances and learning rates between layers at constant width. They find fundamental differences between the optimal layer balance in linear versus nonlinear networks, and that nonlinear networks benefit from faster learning in earlier layers with rapid feature learning and reduced sample complexity on hierarchical data. Atanasov et al. (2025) study width scaling in SP and find that the catapult regime as a function of learning rate and last-layer weight multiplier is larger under CE loss than under MSE loss at finite width. As another potential explanation for the difference between networks at finite and infinite width, Everett et al. (2024) propose that at finite width and over the course of long training alignment between weights and incoming activations may accumulate a smaller width-dependent exponent which could allow larger learning rates. However they only measure the confounded alignment between W_t^{l+1} and x_t^l as opposed to the alignment between $(\Delta W_t^{l+1}, x_t^l)$ and $(W_0^{l+1}, \Delta x_t^l)$ separately, which is necessary for evaluating the scaling predictions by Yang and Hu (2021) and Yang and Littwin (2023).

To conclude, there remains an apparent gap in the existing literature between finite feature-learning networks in SP under large learning rates and infinite-width theory which suggests instability due to logit divergence (Yang and Hu, 2021) and a diverging NTK (Sohl-Dickstein et al., 2020) in SP under large learning rates, and vanishing feature learning under small learning rates.

3.3.2 Summary and Contributions

In Chapter 6, we fundamentally resolve apparent contradictions between networks at finite versus infinite width, and therefore demonstrate that infinite-width theory *can* be useful for understanding the large learning rate regime under SP. We first show that finite-width effects such as catapults (Lewkowycz et al., 2020) and a lack of alignment (Everett et al., 2024) do not suffice for understanding why training remains stable under large learning rate exponents. Instead, we identify that the loss function is the essential component that allows to close the gap between theory and practice. Opposed to MSE loss, we prove that under `torch.nn.CrossEntropyLoss` a *controlled divergence* regime emerges where gradient, activation and loss dynamics remain stable in the infinite-width limit despite logit divergence. Under large learning rates $\eta_n = \Theta(n^{-1/2})$ for SGD in SP and $\eta_n = \Theta(n^{-1})$ for Adam in SP at the upper edge of this controlled divergence regime, hidden-layer feature learning is recovered even in the limit, which partially explains the good performance of SP in practice. We empirically observe that Tensor Program scaling arguments are surprisingly accurate at moderate widths, when evaluated with refined coordinate checks of the form (2.5). In extensive experiments across architectures, data modalities and optimizers, we find that our width-scaling stability and feature learning predictions are surprisingly predictive of optimal learning rate exponents in SP, in particular for deep, fragile networks such as state-of-the-art Transformers. Our controlled divergence regime also covers standard initialization with layerwise learning rates (SP-full-align) recently proposed by Everett et al. (2024), and explains its favorable scaling properties.

To summarize, we provide the first infinite-width theory that exactly covers how neural networks are initialized and trained in practice, and consequently find that this theory is more descriptive of finite neural network scaling properties than previous infinite-width theory, even after long training.

Part II

Publications

Chapter 4

Mind the Spikes: Benign Overfitting of Kernels and Neural Networks in Fixed Dimension

Mind the spikes: Benign overfitting of kernels and neural networks in fixed dimension

Moritz Haas[1]* **David Holzmüller**[2]* **Ulrike von Luxburg**[1] **Ingo Steinwart**[2]
[1]University of Tübingen and Tübingen AI Center, Germany
[2]Institute for Stochastics and Applications, University of Stuttgart, Germany
{mo.haas,ulrike.luxburg}@uni-tuebingen.de
{david.holzmueller,ingo.steinwart}@mathematik.uni-stuttgart.de

Abstract

The success of over-parameterized neural networks trained to near-zero training error has caused great interest in the phenomenon of benign overfitting, where estimators are statistically consistent even though they interpolate noisy training data. While benign overfitting in fixed dimension has been established for some learning methods, current literature suggests that for regression with typical kernel methods and wide neural networks, benign overfitting requires a high-dimensional setting where the dimension grows with the sample size. In this paper, we show that the smoothness of the estimators, and not the dimension, is the key: benign overfitting is possible if and only if the estimator's derivatives are large enough. We generalize existing inconsistency results to non-interpolating models and more kernels to show that benign overfitting with moderate derivatives is impossible in fixed dimension. Conversely, we show that rate-optimal benign overfitting is possible for regression with a sequence of spiky-smooth kernels with large derivatives. Using neural tangent kernels, we translate our results to wide neural networks. We prove that while infinite-width networks do not overfit benignly with the ReLU activation, this can be fixed by adding small high-frequency fluctuations to the activation function. Our experiments verify that such neural networks, while overfitting, can indeed generalize well even on low-dimensional data sets.

1 Introduction

While neural networks have shown great practical success, our theoretical understanding of their generalization properties is still limited. A promising line of work considers the phenomenon of benign overfitting, where researchers try to understand when and how models that interpolate noisy training data can generalize (Zhang et al., 2021, Belkin et al., 2018, 2019). In the high-dimensional regime, where the dimension grows with the number of sample points, consistency of minimum-norm interpolants has been established for linear models and kernel regression (Hastie et al., 2022, Bartlett et al., 2020, Liang and Rakhlin, 2020, Bartlett et al., 2021). In fixed dimension, minimum-norm interpolation with standard kernels is inconsistent (Rakhlin and Zhai, 2019, Buchholz, 2022).

In this paper, we shed a differentiated light on benign overfitting with kernels and neural networks. We argue that the dimension-dependent perspective does not capture the full picture of benign overfitting. In particular, we show that harmless interpolation with kernel methods and neural networks is possible, even in small fixed dimension, with adequately designed kernels and activation functions. The key is to properly design estimators of the form 'signal+spike'. While minimum-norm criteria have widely been considered a useful inductive bias, we demonstrate that designing unusual norms can resolve the shortcomings of standard norms. For wide neural networks, harmless interpolation can be

*Equal contribution.

37th Conference on Neural Information Processing Systems (NeurIPS 2023).

realized by adding tiny fluctuations to the activation function. Such networks do not require explicit regularization and can simply be trained to overfit (Figure 1).

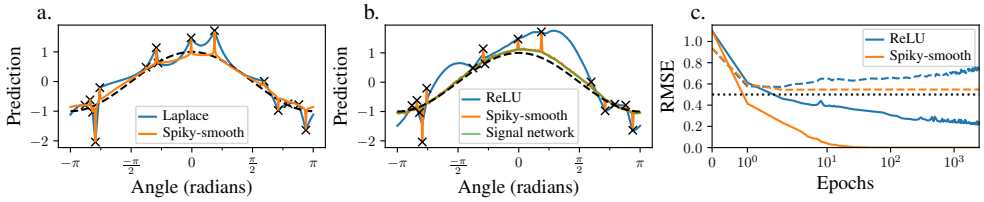


Figure 1: **Spiky-smooth overfitting in 2 dimensions. a.** We plot the predicted function for ridgeless kernel regression with the Laplace kernel (blue) versus our spiky-smooth kernel (4) with Laplace components (orange) on $\mathbb{S}^1$. The dashed black line shows the true regression function, black 'x' denote noisy training points. Further details can be found in Section 6.2. **b.** The predicted function of a trained 2-layer neural network with ReLU activation (blue) versus ReLU plus shifted high-frequency sin-function (8) (orange). Using the weights learned with the spiky-smooth activation function in a ReLU network (green) disentangles the spike component from the signal component. **c.** Training error (solid lines) and test error (dashed lines) over the course of training for b. evaluated on 10^4 test points. The dotted black line shows the optimal test error. The spiky-smooth activation function does not require regularization and can simply be trained to overfit.

On a technical level, we additionally prove that overfitting in kernel regression can only be consistent if the estimators have large derivatives. Using neural tangent kernels or neural network Gaussian process kernels, we can translate our results from kernel regression to the world of neural networks (Neal, 1996, Jacot et al., 2018). In particular, our results enable the design of activation functions that induce benign overfitting in fixed dimension: the spikes in kernels can be translated into infinitesimal fluctuations that can be added to an activation function to achieve harmless interpolation with neural networks. Such small high frequency oscillations can fit noisy observations without affecting the smooth component too much. Training finite neural networks with gradient descent shows that spiky-smooth activation functions can indeed achieve good generalization even when interpolating small, low-dimensional data sets (Figure 1 b,c).

Thanks to new technical contributions, our inconsistency results significantly extend existing ones. We use a novel noise concentration argument (Lemma D.6) to generalize existing inconsistency results on minimum-norm interpolants to the much more realistic regime of overfitting estimators with comparable Sobolev norm scaling, which includes training via gradient flow and gradient descent with "late stopping" as well as low levels of ridge regularization. Moreover, a novel connection to eigenvalue concentration results for kernel matrices (Proposition 5) allows us to relax the smoothness assumption and to treat heteroscedastic noise in Theorem 6. Lastly, our Lemma E.1 translates inconsistency results from bounded open subsets of $\mathbb{R}^d$ to the sphere $\mathbb{S}^d$, which leads to results for the neural tangent kernel and neural network Gaussian processes.

2 Setup and prerequisites

General approach. We consider a general regression problem on $\mathbb{R}^d$ with an arbitrary, fixed dimension d and analyze kernel-based approaches to solve this problem: kernel ridge regression, kernel gradient flow and gradient descent, minimum-norm interpolation, and more generally, overfitting norm-bounded estimators. We then translate our results to neural networks via the neural network Gaussian process and the neural tangent kernel. Let us now introduce the formal framework.

Notation. We denote scalars by lowercase letters x, vectors by bold lowercase letters $\boldsymbol{x}$ and matrices by bold uppercase letters $\boldsymbol{X}$. We denote the eigenvalues of $\boldsymbol{A}$ as $\lambda_1(\boldsymbol{A}) \geq \ldots \geq \lambda_n(\boldsymbol{A})$ and the Moore-Penrose pseudo-inverse by $\boldsymbol{A}^+$. We say that a probability distribution P has lower and upper bounded density if its density p satisfies $0 < c < p(\boldsymbol{x}) < C$ for suitable constants c, C and all $\boldsymbol{x}$ on a given domain.

Regression setup. We consider a data set $D = ((\boldsymbol{x}_1, y_1), \ldots, (\boldsymbol{x}_n, y_n)) \in (\mathbb{R}^d \times \mathbb{R})^n$ with i.i.d. samples $(\boldsymbol{x}_i, y_i) \sim P$, written as $D \sim P^n$, where P is a probability distribution on $\mathbb{R}^d \times \mathbb{R}$. We define $\boldsymbol{X} := (\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n)$ and $\boldsymbol{y} := (y_1, \ldots, y_n)^\top \in \mathbb{R}^n$. Random variables $(\boldsymbol{x}, y) \sim P$ denote

test points independent of D, and P_X denotes the probability distribution of $\boldsymbol{x}$. The (least squares) *empirical risk* R_D and *population risk* R_P of a function $f : \mathbb{R}^d \to \mathbb{R}$ are defined as

$$R_D(f) \coloneqq \frac{1}{n}\sum_{i=1}^{n}(y_i - f(x_i))^2, \qquad R_P(f) \coloneqq \mathbb{E}_{\boldsymbol{x},y}[(y - f(\boldsymbol{x}))^2] \,.$$

We assume $\mathrm{Var}(y|\boldsymbol{x}) < \infty$ for all $\boldsymbol{x}$. Then, R_P is minimized by the target function $f_P^*(\boldsymbol{x}) = \mathbb{E}[y|\boldsymbol{x}]$, and the *excess risk* of a function f is given by

$$R_P(f) - R_P(f_P^*) = \mathbb{E}_{\boldsymbol{x}}(f_P^*(\boldsymbol{x}) - f(\boldsymbol{x}))^2 \,.$$

We call a data-dependent estimator f_D *consistent for* P if its excess risk converges to 0 in probability, that is, for all $\varepsilon > 0$, $\lim_{n\to\infty} P^n\left(D \in (\mathbb{R}^d \times \mathbb{R})^n \mid \; R_P(f_D) - R_P(f_P^*) \geq \varepsilon\right) = 0$. We call f_D *consistent in expectation for* P if $\lim_{n\to\infty} \mathbb{E}_D R_P(f_D) - R_P(f_P^*) = 0$. We call f_D *universally consistent* if is it consistent for all Borel probability measures P on $\mathbb{R}^d \times \mathbb{R}$.

Solutions by kernel regression. Recall that a kernel k induces a reproducing kernel Hilbert space $\mathcal{H}_k$, abbreviated RKHS (more details in Appendix B). For $f \in \mathcal{H}_k$, we consider the objective

$$\mathcal{L}_\rho(f) \coloneqq \frac{1}{n}\sum_{i=1}^{n}(y_i - f(\boldsymbol{x}_i))^2 + \rho\|f\|_{\mathcal{H}_k}^2$$

with regularization parameter $\rho \geq 0$. Denote by $f_{t,\rho}$ the solution to this problem that is obtained by optimizing on $\mathcal{L}_\rho$ in $\mathcal{H}_k$ with gradient flow until time $t \in [0, \infty]$, using fixed a regularization constant $\rho > 0$, and initializing at $f = 0 \in \mathcal{H}_k$. We show in Appendix C.1 that it is given as

$$f_{t,\rho}(\boldsymbol{x}) \coloneqq k(\boldsymbol{x}, \boldsymbol{X})\left(\boldsymbol{I}_n - e^{-\frac{2}{n}t(k(\boldsymbol{X},\boldsymbol{X})+\rho n \boldsymbol{I}_n)}\right)(k(\boldsymbol{X},\boldsymbol{X}) + \rho n \boldsymbol{I}_n)^{-1}\boldsymbol{y} \,, \tag{1}$$

where $k(\boldsymbol{x}, \boldsymbol{X})$ denotes the row vector $(k(\boldsymbol{x}, \boldsymbol{x}_i))_{i\in[n]}$ and $k(\boldsymbol{X}, \boldsymbol{X}) = (k(\boldsymbol{x}_i, \boldsymbol{x}_j))_{i,j\in[n]}$ the kernel matrix. $f_{t,\rho}$ elegantly subsumes several popular kernel regression estimators as special cases: (i) classical kernel ridge regression for $t \to \infty$, (ii) gradient flow on the unregularized objective for $\rho \searrow 0$, and (iii) kernel "ridgeless" regression $f_{\infty,0}(\boldsymbol{x}) = k(\boldsymbol{x}, \boldsymbol{X})k(\boldsymbol{X}, \boldsymbol{X})^+\boldsymbol{y}$ in the joint limit of $\rho \to 0$ and $t \to \infty$. If $k(\boldsymbol{X}, \boldsymbol{X})$ is invertible, $f_{\infty,0}$ is the interpolating function $f \in \mathcal{H}_k$ with the smallest $\mathcal{H}_k$-norm.

From kernels to neural networks: the neural tangent kernel (NTK) and the neural network Gaussian process (NNGP) . Denote the output of a NN with parameters $\boldsymbol{\theta}$ on input $\boldsymbol{x}$ by $f_{\boldsymbol{\theta}}(\boldsymbol{x})$. It is known that for suitable random initializations $\boldsymbol{\theta}_0$, in the infinite-width limit the random initial function $f_{\boldsymbol{\theta}_0}$ converges in distribution to a Gaussian Process with the so-called Neural Network Gaussian Process (NNGP) kernel (Neal, 1996, Lee et al., 2018, Matthews et al., 2018). In Bayesian inference, the posterior mean function is then of the form $f_{\infty,\rho}$. With minor modifications (Arora et al., 2019, Zhang et al., 2020), training infinitely wide NNs with gradient flow corresponds to learning the function $f_{t,0}$ with the neural tangent kernel (NTK) (Jacot et al., 2018, Lee et al., 2019). If only the last layer is trained, the NNGP kernel should be used instead (Daniely et al., 2016). For ReLU activation functions, the RKHS of the infinite-width NNGP and NTK on the sphere $\mathbb{S}^d$ is typically a Sobolev space (Bietti and Bach, 2021, Chen and Xu, 2021), see Appendix B.4. Using other parametrizations induces feature learning infinite-width limits for neural networks (Yang and Hu, 2021); an analysis of such neural network algorithms is left for future work.

3 Related work

We here provide a short summary of related work. A more detailed account is provided in Appendix A.

Kernel regression. With appropriate regularization, kernel ridge regularization with typical universal kernels like the Gauss, Matérn, and Laplace kernels is universally consistent (Steinwart and Christmann, 2008, Chapter 9). Optimal rates in Sobolev RKHS can also be achieved using cross-validation of the regularization ρ (Steinwart et al., 2009) or early stopping rules (Yao et al., 2007, Raskutti et al., 2014, Wei et al., 2017). The above kernels as well as NTKs and NNGPs of standard fully-connected neural networks are rotationally invariant. In the high-dimensional regime, the class of functions that is learnable with rotation-invariant kernels is quite limited (Donhauser et al., 2021, Ghorbani et al., 2021, Liang et al., 2020).

Inconsistency results. Besides Rakhlin and Zhai (2019) and Buchholz (2022), Beaglehole et al. (2023) derive inconsistency results for ridgeless kernel regression given assumptions on the spectral tail in the Fourier basis, and contemporaneously propose a special case of our spiky-smooth kernel sequence to mimic kernel ridge regression without providing any quantitative statements. Li et al. (2023) show that polynomial convergence is impossible for common kernels including ReLU NTKs. Mallinar et al. (2022) conjecture inconsistency for interpolation with ReLU NTKs based on their semi-rigorous result , which essentially assumes that the eigenfunctions can be replaced by structureless Gaussian random variables. Lai et al. (2023) show an inconsistency-type result for overfitting two-layer ReLU NNs with $d = 1$, but for fixed inputs $\boldsymbol{X}$. They also note that an earlier inconsistency result by Hu et al. (2021) relies on an unproven result. Mücke and Steinwart (2019) show that global minima of NNs can overfit both benignly and harmfully, but their result does not apply to gradient descent training. Overfitting with typical linear models around the interpolation peak is inconsistent (Ghosh and Belkin, 2022, Holzmüller, 2021).

Classification. For binary classification, benign overfitting is a more generic phenomenon than for regression (Muthukumar et al., 2021, Shamir, 2022), and consistency has been shown under linear separability assumptions (Montanari et al., 2019, Chatterji and Long, 2021, Frei et al., 2022), through complexity bounds for reference classes (Cao and Gu, 2019, Chen et al., 2021) or as long as the total variation distance of the class conditionals is sufficiently large and $f^*(\boldsymbol{x}) = \mathbb{E}[y|\boldsymbol{x}]$ lies in the RKHS with bounded norm (Liang and Recht, 2023). Chapter 8 of Steinwart and Christmann (2008) discusses how the overlap of the two classes may influence learning rates under positive regularization.

4 Inconsistency of overfitting with common kernel estimators

We consider a regression problem on $\mathbb{R}^d$ in arbitrary, fixed dimension d that is solved by kernel regression. In this section, we derive several new results, stating that overfitting estimators with moderate Sobolev norm are inconsistent, in a variety of settings. In the next section, we establish the other direction: overfitting estimators can be consistent when we adapt the norm that is minimized.

4.1 Beyond minimum-norm interpolants: general overfitting estimators with bounded norm

Existing generalization bounds often consider the perfect minimum norm interpolant. This is a rather theoretical construction; estimators obtained by training with gradient descent algorithms merely overfit and, in the best case, approximate interpolants with small norm. In this section, we extend existing bounds to arbitrary overfitting estimators whose norm does not grow faster than the minimum norm that would be required to interpolate the training data. Before we can state the theorem, we need to establish some technical assumptions.

Assumptions on the data generating process. The following assumptions (as in Buchholz (2022)) allow for quite general domains and distributions. They are standard in nonparametric statistics.

(D1) Let P_X be a distribution on a bounded open Lipschitz domain $\Omega \subseteq \mathbb{R}^d$ with lower and upper bounded Lebesgue density. Consider data sets $D = \{(\boldsymbol{x}_1, y_1), \dots, (\boldsymbol{x}_n, y_n)\}$, where $\boldsymbol{x}_i \sim P_X$ i.i.d. and $y_i = f^*(\boldsymbol{x}_i) + \varepsilon_i$, where ε_i is i.i.d. Gaussian noise with positive variance $\sigma^2 > 0$ and $f^* \in C_c^\infty(\Omega) \backslash \{0\}$ denotes a smooth function with compact support.

Assumptions on the kernel. Our assumption on the kernel is that its RKHS is equivalent to a Sobolev space. For integers $s \in \mathbb{N}$, the norm of a Sobolev space $H^s(\Omega)$ can be defined as

$$\|f\|_{H^s(\Omega)}^2 \coloneqq \sum_{0 \leq |\alpha| \leq s} \|D^\alpha f\|_{L_2(\Omega)}^2,$$

where D^α denotes partial derivatives in multi-index notation for α. It measures the magnitude of derivatives up to some order s. For general $s > 0$, $H^s(\Omega)$ is (equivalent to) an RKHS if and only if $s > d/2$. For example, Laplace and Matérn kernels (Kanagawa et al., 2018, Example 2.6) have Sobolev RKHSs. The RKHS of the Gaussian kernel $\mathcal{H}^{\text{Gauss}}$ is contained in every Sobolev space, $\mathcal{H}^{\text{Gauss}} \subsetneq H^s$ for all $s \geq 0$ (Steinwart and Christmann, 2008, Corollary 4.36). Due to its smoothness, the Gaussian kernel is potentially even more prone to harmful overfitting than Sobolev kernels (Mallinar et al., 2022). We make the following assumption on the kernel:

(K) Let k be a positive definite kernel function whose RKHS $\mathcal{H}_k$ is equivalent to the Sobolev space H^s for some $s \in (\frac{d}{2}, \frac{3d}{4})$.

Now we are ready to state the main result of this section:

Theorem 1 (Overfitting estimators with small norms are inconsistent). *Let assumptions (D1) and (K) hold. Let $c_{\text{fit}} \in (0, 1]$ and $C_{\text{norm}} > 0$. Then, there exist $c > 0$ and $n_0 \in \mathbb{N}$ such that the following holds for all $n \geq n_0$ with probability $1 - O(1/n)$ over the draw of the data set D with n samples: Every function $f \in \mathcal{H}_k$ that satisfies the follwing two conditions*

(O) $\frac{1}{n}\sum_{i=1}^{n}(f(x_i) - y_i)^2 \leq (1 - c_{\text{fit}}) \cdot \sigma^2$ *(training error of f is below Bayes risk)*
(N) $\|f\|_{\mathcal{H}_k} \leq C_{\text{norm}}\|f_{\infty,0}\|_{\mathcal{H}_k}$ *(norm comparable to minimum-norm interpolant (1)),*

has an excess risk that satisfies

$$R_P(f) - R_P(f^*) \geq c\sigma^2 > 0\,. \tag{2}$$

In words: In fixed dimension d, every differentiable function f that overfits the training data and is not much "spikier" than the minimum RKHS-norm interpolant is inconsistent!

Proof idea. Our proof follows a similar approach as Rakhlin and Zhai (2019), Buchholz (2022), and also holds for kernels with adaptive bandwidths. For small bandwidths, $\|f_{\infty,0}\|_{L_2(P_X)} \ll \|f^*\|_{L_2(P_X)}$ because $f_{\infty,0}$ decays to 0 between the training points, which shows that purely 'spiky' estimators are inconsistent. In this case, the lower bound is independent of σ^2. For all other bandwidths, interpolating $\Theta(n)$ many noisy labels y_i incurs $\Theta(1)$ error in an area of volume $\Omega(1/n)$ around $\Theta(n)$ data points with high probability, which accumulates to a total error $\Omega(1)$. Our observation is that the same logic holds when overfitting by a constant fraction. Formally, we show that f^* and f must then be separated by a constant on a constant fraction of training points, with high probability, by using the fact that a constant fraction of the total noise cannot concentrate on less than $\Theta(n)$ noise variables, with high probability (Lemma D.6). The full proof can be found in Appendix D. □

Assumption (O) is necessary in Theorem 1, because optimally regularized kernel ridge regression fulfills all other assumptions of Theorem 1 while achieving consistency with minimax optimal convergence rates (see Section 3). The necessity of Assumption (N) is demonstrated by Section 5.

The following proposition establishes that Theorem 1 covers the entire overfitting regime of the popular (regularized) gradient flow estimators $f_{t,\rho}$ for all times $t \in [0, \infty]$ and any regularization $\rho \geq 0$. The proof in Appendix C.2 also covers gradient descent.

Proposition 2 (Popular estimators fulfill the norm bound (N)). *For arbitrary $t \in [0, \infty]$ and $\rho \in [0, \infty)$, $f_{t,\rho}$ as defined in (1) fulfills Assumption (N) with $C_{\text{norm}} = 1$.*

Remark 3 (Dimension dependency). Some works argue that for specific sequences of kernels $(k_d)_{d\in\mathbb{N}}$, the constant c in Theorem 1 decreases with increasing dimension d (Liang et al., 2020, Liang and Rakhlin, 2020, Mallinar et al., 2022). In Theorem 1, if the equivalence constants in Assumption (K) are uniformly bounded in d, the behavior in d might still depend on the definition of the Sobolev norms. Overall, similar to Rakhlin and Zhai (2019) and Buchholz (2022), our proof techniques do not allow to easily obtain a dependence on d. ◀

4.2 Inconsistency of overfitting with neural kernels

We would now like to apply the above results to neural kernels, which would allow us to translate our inconsistency results from the kernel domain to neural networks. However, to achieve this, we need to take one more technical hurdle: the equivalence results for NTKs and NNGPs only hold for probability distributions on the sphere $\mathbb{S}^d$ (detailed summary in Appendix B.4). Lemma E.1 provides the missing technical link: It establishes a smooth correspondence between the respective kernels, Sobolev spaces, and probability distributions. The inconsistency of overfitting with (deep) ReLU NTKs and NNGP kernels then immediately follows from adapting Theorem 1 via Lemma E.1.

Theorem 4 (Overfitting with neural network kernels in fixed dimension is inconsistent). *Let $c \in (0, 1)$, and let P be a probability distribution with lower and upper bounded Lebesgue density on an arbitrary spherical cap $T \coloneqq \{\boldsymbol{x} \in \mathbb{S}^d \mid x_{d+1} < v\} \subseteq \mathbb{S}^d$, $v \in (-1, 1)$. Let k either be*

(i) the fully-connected ReLU NTK with 0-initialized biases of any fixed depth $L \geq 2$, and $d \geq 2$, or
(ii) the fully-connected ReLU NNGP kernel without biases of any fixed depth $L \geq 3$, and $d \geq 6$.

Then, if $f_{t,\rho}$ fulfills Assumption (O) with probability at least c over the draw of the data set D, $f_{t,\rho}$ is inconsistent for P.

Theorem 4 also holds for more general estimators as in Theorem 1, cf. the proof in Appendix E.

Mallinar et al. (2022) already observed empirically that overfitting common network architectures yields suboptimal generalization performance on large data sets in fixed dimension. Theorem 4 now provides a rigorous proof for this phenomenon since sufficiently wide trained neural networks and the corresponding NTKs have a similar generalization behavior (e.g. (Arora et al., 2019, Theorem 3.2)).

4.3 Relaxing smoothness and noise assumptions via spectral concentration bounds

In this section, we consider a different approach to derive lower bounds for the generalization error of overfitting kernel regression: through concentration results for the eigenvalues of kernel matrices. On a high level, we obtain similar results as in the last section. The novelty of this section is on the technical side, and we suggest that non-technical readers skip this section in their first reading.

We define the convolution kernel of a given kernel k as $k_*(\boldsymbol{x}, \boldsymbol{x}') \coloneqq \int k(\boldsymbol{x}, \boldsymbol{x}'')k(\boldsymbol{x}'', \boldsymbol{x}') \,\mathrm{d}P_X(\boldsymbol{x}'')$, which is possible whenever $k(\boldsymbol{x}, \cdot) \in L_2(P_X)$ for all $\boldsymbol{x}$. The latter condition is satisfied for bounded kernels. Our starting point is the following new lower bound:

Proposition 5 (Spectral lower bound). *Assume that the kernel matrix $k(\boldsymbol{X}, \boldsymbol{X})$ is almost surely positive definite, and that $\mathrm{Var}(y|\boldsymbol{x}) \geq \sigma^2$ for P_X-almost all $\boldsymbol{x}$. Then, the expected excess risk satisfies*

$$\mathbb{E}_D R_P(f_{t,\rho}) - R_P^* \geq \frac{\sigma^2}{n} \sum_{i=1}^{n} \mathbb{E}_{\boldsymbol{X}} \frac{\lambda_i(k_*(\boldsymbol{X}, \boldsymbol{X})/n) \left(1 - e^{-2t(\lambda_i(k(\boldsymbol{X}, \boldsymbol{X})/n)+\rho)}\right)^2}{(\lambda_i(k(\boldsymbol{X}, \boldsymbol{X})/n) + \rho)^2} \,. \tag{3}$$

Using concentration inequalities for kernel matrices and the relation between the integral operators of k and k_*, it can be seen that for $t = \infty$ and $\rho = 0$, every term in the sum in Eq. (3) should converge to 1 as $n \to \infty$. However, since the number of terms in the sum increases with n and the convergence may not be uniform, this is not sufficient to show inconsistency in expectation. Instead, relative concentration bounds that are even stronger than the ones by Valdivia (2018) would be required to show inconsistency in expectation. However, by combining multiple weaker bounds and further arguments on kernel equivalences, we can still show inconsistency in expectation for a class of dot-product kernels on the sphere, including certain NTK and NNGP kernels (Appendix B.4):

Theorem 6 (Inconsistency for Sobolev dot-product kernels on the sphere). *Let k be a dot-product kernel on $\mathbb{S}^d$, i.e., a kernel of the form $k(\boldsymbol{x}, \boldsymbol{x}') = \kappa(\langle \boldsymbol{x}, \boldsymbol{x}' \rangle)$, such that its RKHS $\mathcal{H}_k$ is equivalent to a Sobolev space $H^s(\mathbb{S}^d)$, $s > d/2$. Moreover, let P be a distribution on $\mathbb{S}^d \times \mathbb{R}$ such that P_X has a lower and upper bounded density w.r.t. the uniform distribution $\mathcal{U}(\mathbb{S}^d)$, and such that $\mathrm{Var}(y|\boldsymbol{x}) \geq \sigma^2 > 0$ for P_X-almost all $\boldsymbol{x} \in \mathbb{S}^d$. Then, for every $C > 0$, there exists $c > 0$ independent of σ^2 such that for all $n \geq 1$, $t \in (C^{-1} n^{2s/d}, \infty]$, and $\rho \in [0, C n^{-2s/d})$, the expected excess risk satisfies*

$$\mathbb{E}_D R_P(f_{t,\rho}) - R_P^* \geq c\sigma^2 > 0 \,.$$

The assumptions of Theorem 6 and Theorem 4 differ in several ways. Theorem 6 applies to arbitrarily high smoothness s and therefore to ReLU NTKs and NNGPs in arbitrary dimension d. Moreover, it applies to distributions on the whole sphere and allows more general noise distributions. On the flip side, it only shows inconsistency in expectation, which we believe could be extended to inconsistency for Gaussian noise. Moreover, it only applies to functions of the form $f_{t,\rho}$ but provides an explicit bound on t and ρ to get inconsistency. For $t = \infty$, the bound $\rho = O(n^{-2s/d})$ appears to be tight, as larger ρ yield consistency for comparable Sobolev kernels on $\mathbb{R}^d$ (Steinwart et al., 2009, Corollary 3).

We only prove Theorem 6 for dot-product kernels on the sphere since we can show for these kernels that $\mathcal{H}_{k_*}$ is equivalent to a Sobolev space (Lemma F.13), while this is not true for open domains Ω (Schaback, 2018). However, an improved understanding of $\mathcal{H}_{k_*}$ for such Ω could potentially allow to extend our proof to the non-spherical case.

The spectral lower bounds in Theorem F.2 show that our approach can directly benefit from developing better kernel matrix concentration inequalities. Conversely, the investigation of consistent kernel interpolation might provide information about where such concentration inequalities do not hold.

5 Consistency via spiky-smooth estimators – even in fixed dimension

In Section 4, we have seen that when common kernel estimators overfit, they are inconsistent for many kernels and a wide variety of distributions. We now design consistent interpolating kernel estimators. The key is to violate Assumption (N) for every fixed Sobolev RKHS norm $\|\cdot\|_{\mathcal{H}_k}$ and introduce an inductive bias towards learning spiky-smooth functions.

5.1 Almost universal consistency of spiky-smooth ridgeless kernel regression

In high dimensional regimes (where the dimension d is supposed to grow with the number of data points), benign overfitting of linear and kernel regression has been understood by an additive decomposition of the minimum-norm interpolant into a smooth regularized component that is responsible for good generalization, and a spiky component that interpolates the noisy data points while not harming generalization (Bartlett et al., 2021). This inspires us to enforce such a decomposition in arbitrary fixed dimension by adding a sharp kernel spike $\rho\check{k}_{\gamma_n}$ to a common kernel $\tilde{k}$. In this way, we can still generate any Sobolev RKHS (see Appendix G.2).

Definition 7 (Spiky-smooth kernel). Let $\tilde{k}$ denote any universal kernel function on $\mathbb{R}^d$. We call it the smooth component. Consider a second, translation invariant kernel $\check{k}_\gamma$ of the form $k_\gamma(\boldsymbol{x},\boldsymbol{y}) = q(\frac{\boldsymbol{x}-\boldsymbol{y}}{\gamma})$, for some function $q : \mathbb{R}^d \to \mathbb{R}$. We call it the spiky component. Then we define the *ρ-regularized spiky-smooth kernel with spike bandwidth γ* as

$$k_{\rho,\gamma}(\boldsymbol{x},\boldsymbol{y}) = \tilde{k}(\boldsymbol{x},\boldsymbol{y}) + \rho\cdot\check{k}_\gamma(\boldsymbol{x},\boldsymbol{y}), \qquad \boldsymbol{x},\boldsymbol{y}\in\mathbb{R}^d. \quad (4)$$

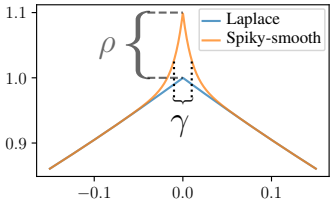


Figure 2: The spiky-smooth kernel with Laplace components (orange) consists of a Laplace kernel (blue) plus a Laplace kernel of height ρ and small bandwidth γ.

We now show that the minimum-norm interpolant of the spiky-smooth kernel sequence with properly chosen $\rho_n, \gamma_n \to 0$ is consistent for a large class of distributions, on a space with fixed (possibly small) dimension d. We establish our result under the following assumption (as in Mücke and Steinwart (2019)), which is weaker than our previous Assumption (D1).

(D2) There exists a constant $\beta_X > 0$ and a continuous function $\phi : [0,\infty) \to [0,1]$ with $\phi(0) = 0$ such that the data generating probability distribution satisfies $P_X(B_t(\boldsymbol{x})) \le \phi(t) = O(t^{\beta_X})$ for all $\boldsymbol{x}\in\Omega$ and all $t \ge 0$ (here $B_t(\boldsymbol{x})$ denotes the Euclidean ball of radius t around $\boldsymbol{x}$).

Theorem 8 (Consistency of spiky-smooth ridgeless kernel regression). *Assume that the training set D consists of n i.i.d. pairs $(\boldsymbol{x},y) \sim P$ such that the marginal P_X fulfills (D2) and $\mathbb{E}y^2 < \infty$. Let the kernel components satisfy:*

- *$\tilde{k}$ is a universal kernel, and $\rho_n \to 0$ and $n\rho_n^4 \to \infty$.*
- *$\check{k}_{\gamma_n}$ denotes the Laplace kernel with a sequence of positive bandwidths (γ_n) fulfilling $\gamma_n \le n^{-\frac{2+\alpha}{d}}\left((\frac{9}{4}+\frac{\alpha}{2})\ln n\right)^{-1}$, where $\alpha > 0$ arbitrary.*

Then the minimum-norm interpolant of the ρ_n-regularized spiky-smooth kernel sequence $k_n := k_{\rho_n,\gamma_n}$ is consistent for P.

Remark 9 (Benign overfitting with optimal convergence rates). Suppose that we have a Sobolev target function $f^* \in H^{s^*}(\Omega)\backslash\{0\}$, that the noise satisfies a moment condition and that P_X has an upper- and lower-bounded density on a Lipschitz domain or the sphere. Then, we show in Theorem G.5 that, by using smooth components $\tilde{k}$ whose RKHS is equivalent to a Sobolev space H^s, $s > \max(s^*, d/2)$, and choosing the spike components $\check{k}_{\gamma_n}$ as in Theorem 8, the minimum-norm interpolant of $k_n := k_{\rho_n,\gamma_n}$ achieves the convergence rate $n^{-\frac{s^*}{(s^*+d/2)}}\log^2(n)$ when choosing the quasi-regularization ρ_n properly. Moreover, for $s^* > d/2$, this rate is known to be optimal up to the factor $\log^2(n)$ (Remark G.6). Since optimal rates can be achieved both with optimal regularization and with interpolation, our results show that in Sobolev RKHSs, overfitting is neither intrinsically helpful nor harmful for generalization with the right choice of kernel function. ◀

Proof idea. With sharp spikes $\gamma \to 0$, it holds that $\check{k}_\gamma(\boldsymbol{X}, \boldsymbol{X}) \approx \boldsymbol{I}_n$, with high probability. Hence, ridgeless kernel regression with the spiky-smooth kernel interpolates the training set while approximating kernel ridge regression with the smooth component $\tilde{k}$ and regularization ρ. □

The theorem even holds under much weaker assumptions on the decay behavior of the spike component $\check{k}_{\gamma_n}$, including Gaussian and Matérn kernels. The full version of the theorem and its proof can be found in Appendix G. It also applies to kernels and distributions on the sphere $\mathbb{S}^d$.

Remark 10 (Interplay between smoothness and dimensionality). Irrespective of the dimension d, we achieve benign overfitting with estimators in RKHS of arbitrary degrees of smoothness. With increasing d, for the Laplace kernel the spike bandwidth is allowed to be chosen as $\gamma_n = \Omega(n^{-(2+\alpha)/d})$, $\alpha > 0$, for covariate distributions with upper bounded Lebesgue density (see Remark G.2). Hence the magnitude of derivatives of the spikes is allowed to scale less aggressively with increasing dimension. ◀

5.2 From spiky-smooth kernels to spiky-smooth activation functions

So far, our discussion revolved around the properties of kernels and whether they lead to estimators that are consistent. We now turn our attention to the neural network side. The big question is whether it is possible to specifically design activation functions that enable benign overfitting in fixed, possibly small dimension. We will see that the answer is yes: similarly to adding sharp spikes to a kernel, we add tiny fluctuations to the activation function. Concretely, we exploit (Simon et al., 2022, Theorem 3.1). It states that any dot-product kernel on the sphere that is a dot-product kernel in every dimension d can be written as an NNGP kernel or an NTK of two-layer fully-connected networks with a specifically chosen activation function. Further details can be found in Appendix H.

Theorem 11 (Connecting kernels and activation functions (Simon et al., 2022)). *Let $\kappa : [-1, 1] \to \mathbb{R}$ be a function such that $k_d : \mathbb{S}^d \times \mathbb{S}^d \to \mathbb{R}, k_d(\boldsymbol{x}, \boldsymbol{x}') = \kappa(\langle \boldsymbol{x}, \boldsymbol{x}' \rangle)$ is a kernel for every $d \geq 1$. Then, there exist $b_i \geq 0$ with $\sum_{i=0}^\infty b_i < \infty$ such that $\kappa(t) = \sum_{i=0}^\infty b_i t^i$, and for any choice of signs $(s_i)_{i \in \mathbb{N}_0} \subseteq \{-1, +1\}$, the kernel k_d can be realized as the NNGP kernel or NTK of a two-layer fully-connected network without biases and with activation function*

$$\phi^{k_d}_{NNGP}(x) = \sum_{i=0}^\infty s_i (b_i)^{1/2} h_i(x), \qquad \phi^{k_d}_{NTK}(x) = \sum_{i=0}^\infty s_i \left(\frac{b_i}{i+1}\right)^{1/2} h_i(x). \tag{5}$$

Here, h_i denotes the i-th Probabilist's Hermite polynomial normalized such that $\|h_i\|_{L_2(\mathcal{N}(0,1))} = 1$.

The following proposition justifies the approach of adding spikes $\rho^{1/2}\phi^{\check{k}_\gamma}$ to an activation function to enable harmless interpolation with wide neural networks. Here we state the result for the case of the NTK; an analogous result holds for induced NNGP activation functions.

Proposition 12 (Additive decomposition of spiky-smooth activation functions). *Fix $\tilde{\gamma}, \rho > 0$ arbitrary. Let $k = \tilde{k} + \rho \check{k}_\gamma$ denote the spiky-smooth kernel where $\tilde{k}$ and $\check{k}_\gamma$ are Gaussian kernels of bandwidth $\tilde{\gamma}$ and γ, respectively. Assume that we choose signs $\{s_i\}_{i \in \mathbb{N}}$ and then the activation functions ϕ^k_{NTK}, $\phi^{\tilde{k}}_{NTK}$ and $\phi^{\check{k}_\gamma}_{NTK}$ as in Theorem 11. Then, for $\gamma > 0$ small enough, it holds that*

$$\|\phi^k_{NTK} - (\phi^{\tilde{k}}_{NTK} + \sqrt{\rho} \cdot \phi^{\check{k}_\gamma}_{NTK})\|^2_{L_2(\mathcal{N}(0,1))} \leq 2^{1/2} \rho \gamma^{3/2} \exp\left(-\frac{1}{\gamma}\right) + \frac{4\pi(1+\tilde{\gamma})\gamma}{\tilde{\gamma}}.$$

Proof idea. When the spikes are sharp enough (γ small enough), the smooth and the spiky component of the activation function are approximately orthogonal in $L_2(\mathcal{N}(0, 1))$ (Figure 3c), so that the spiky-smooth activation function can be approximately additively decomposed into the smooth activation component $\phi^{\tilde{k}}$ and the spike component $\phi^{\check{k}}$ responsible for interpolation. □

To motivate why the added spike functions $\rho^{1/2}\phi^{\check{k}_\gamma}$ should have small amplitudes, observe that Gaussian activation components $\phi^{\check{k}_\gamma}$ satisfy

$$\|\phi^{\check{k}_\gamma}_{NNGP}\|^2_{L_2(\mathcal{N}(0,1))} = 1, \qquad \|\phi^{\check{k}_\gamma}_{NTK}\|^2_{L_2(\mathcal{N}(0,1))} = \frac{\gamma}{2}\left(1 - \exp\left(-\frac{2}{\gamma}\right)\right). \tag{6}$$

Hence, the average amplitude of NNGP spike activation components $\rho^{1/2}\phi^{\check{k}_\gamma}$ does not depend on γ, while the average amplitude of NTK spike components decays to 0 with $\gamma \to 0$. Since consistency requires the quasi-regularization $\rho \to 0$, the spiky component of induced NTK as well as NNGP activation functions should vanish for large data sets $n \to \infty$ to achieve consistency.

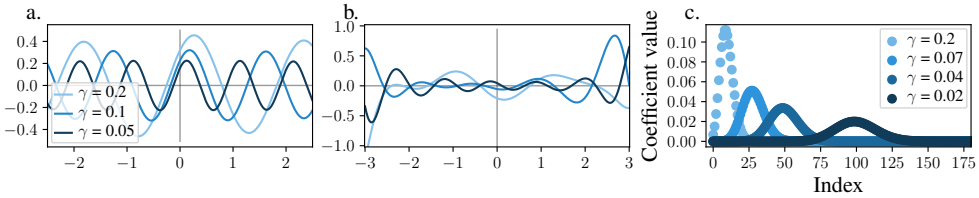


Figure 3: **a., b.** Gaussian NTK activation components $\phi_{NTK}^{\check{k}_\gamma}$ defined via (5) induced by the Gaussian kernel with varying bandwidth $\gamma \in [0.2, 0.1, 0.05]$ (the darker, the smaller γ) for **a.** bi-alternating signs $s_i = +1$ iff $\lfloor i/2 \rfloor$ even, and **b.** randomly iid chosen signs $s_i \sim \mathcal{U}(\{-1, +1\})$. **c.** Coefficients of the Hermite series of a Gaussian NTK activation component with varying bandwidth γ. Observe peaks at $2/\gamma$. For reliable approximations of activation functions use a truncation $\geq 4/\gamma$. The sum of squares of the coefficients follows Eq. (6). Figure I.8 visualizes NNGP activation components.

6 Experiments

Now we explore how appropriate spiky-smooth activation functions might look like and whether they indeed enable harmless interpolation for trained networks of finite width on finite data sets. Further experimental results are reported in Appendix I.

6.1 What do common activation functions lack in order to achieve harmless interpolation?

To understand which properties we have to introduce into activation functions to enable harmless interpolation, we plot NTK spike components $\phi^{\check{k}_\gamma}$ induced by the Gaussian kernel (Figure 3a,b) as well as their Hermite series coefficients (Figure 3c). Remarkably, the spike components $\phi^{\check{k}_\gamma}$ approximately correspond to a shifted, high-frequency $\sin$-curve, when choosing the signs s_i in (5) to alternate every second i, that is $s_i = +1$ iff $\lfloor i/2 \rfloor$ even (Figure 3a). Proposition H.1 shows that the NNGP activation functions correspond to the fluctuation function

$$\omega_{\mathrm{NNGP}}(x; \gamma) \coloneqq \sqrt{2} \cdot \sin\left(\sqrt{2/\gamma} \cdot x + \pi/4\right) = \sin\left(\sqrt{2/\gamma} \cdot x\right) + \cos\left(\sqrt{2/\gamma} \cdot x\right), \tag{7}$$

where the last equation follows from the trigonometric addition theorem. For small bandwidths γ, the NTK activation functions are increasingly well approximated (Appendix I.6) by

$$\omega_{\mathrm{NTK}}(x; \gamma) \coloneqq \sqrt{\gamma} \cdot \sin\left(\sqrt{2/\gamma} \cdot x + \pi/4\right) = \sqrt{\gamma/2}\left(\sin\left(\sqrt{2/\gamma} \cdot x\right) + \cos\left(\sqrt{2/\gamma} \cdot x\right)\right). \tag{8}$$

With decreasing bandwidth $\gamma \to 0$ the frequency increases, while the amplitude decreases for the NTK and remains constant for the NNGP (see Eq. (6)). Plotting equivalent spike components $\phi^{\check{k}_\gamma}$ with different choices of the signs s_i (Figure 3b and Appendix I.5) suggests that harmless interpolation requires activation functions that contain **small high-frequency oscillations** or that **explode at large** $|x|$, which only affects few neurons. The Hermite series expansion of suitable activation functions should contain **non-negligible weight spread across high-order coefficients** (Figure 3c). While Simon et al. (2022) already truncate the Hermite series of induced activation functions at order 5, Figure 3c shows that an accurate approximation of spiky-smooth activation functions requires the truncation index to be larger than $2/\gamma$. Only a careful implementation allows us to capture the high-order fluctuations in the Hermite series of the spiky activation functions. Our implementation can be found at `https://github.com/moritzhaas/mind-the-spikes`.

6.2 Training neural networks to achieve harmless interpolation in low dimension

In Figure 1, we plot the results of (a) ridgeless kernel regression and (b) trained 2-layer neural networks with standard choices of kernels and activation functions (blue) as well as our spiky-smooth alternatives (orange). We trained on 15 points sampled i.i.d. from $x = (x_1, x_2) \sim \mathcal{U}(\mathbb{S}^1)$ and $y = x_1 + \varepsilon$ with $\varepsilon \sim \mathcal{N}(0, 0.25)$. The figure shows that both the Laplace kernel and standard ReLU networks interpolate the training data too smoothly in low dimension, and do not generalize well. However, our spiky-smooth kernel and neural networks with spiky-smooth activation functions achieve close to optimal generalization while interpolating the training data with sharp spikes.

We achieve this by using the adjusted activation function with high-frequency oscillations $x \mapsto \mathrm{ReLU}(x) + \omega_{\mathrm{NTK}}(x; \frac{1}{5000})$ as defined in Eq. (8). With this choice, we avoid activation functions with exploding behavior, which would induce exploding gradients. Other choices of amplitude and frequency in Eq. (8) perform worse. To bring our neural networks close to the kernel regime, we use the neural tangent parameterization (Jacot et al., 2018) and make the networks very wide (20000 hidden neurons). To ensure that the initial function is identically zero, we use the antisymmetric initialization trick (Zhang et al., 2020). Over the course of training (Figure 1c), the standard ReLU network exhibits harmful overfitting, whereas the NN with a spiky-smooth activation function quickly interpolates the training set with nearly optimal generalization. Training details and hyperparameter choices can be found in Appendix I.1. Although the high-frequency oscillations perturb the gradients, the NN with spiky smooth activation has a stable training trajectory using gradient descent with a large learning rate of 0.4 or stochastic gradient descent with a learning rate of 0.04. Since our activation function is the sum of two terms, we can additively decompose the network into its ReLU-component and its ω_{NTK}-component. Figure 1b and Appendix I.2 demonstrate that our interpretation of the ω_{NTK}-component as 'spiky' is accurate: The oscillations in the hidden neurons induced by ω_{NTK} interfere constructively to interpolate the noise in the training points and regress to 0 between training points. This entails immediate access to the signal component of the trained neural network in form of its ReLU-component.

7 Conclusion

Conceptually, our work shows that inconsistency of overfitting is quite a generic phenomenon for regression in fixed dimension. However, particular spiky-smooth estimators enable benign overfitting, even in fixed dimension. We translate the spikes that lead to benign overfitting in kernel regression into infinitesimal fluctuations that can be added to activation functions to consistently interpolate with wide neural networks. Our experiments verify that neural networks with spiky-smooth activation functions can exhibit benign overfitting even on small, low-dimensional data sets.

Technically, our inconsistency results cover many distributions, Sobolev spaces of arbitrary order, and arbitrary RKHS-norm-bounded overfitting estimators. Lemma E.1 serves as a generic tool to extend generalization bounds to the sphere $\mathbb{S}^d$, allowing us to cover (deep) ReLU NTKs and ReLU NNGPs.

Future work. While our experiments serve as a promising proof of concept, it remains unclear how to design activation functions that enable harmless interpolation of more complex neural network architectures and data sets. As another interesting insight, our consistent kernel sequence shows that although kernels may have equivalent RKHS (see Appendix G.2), their generalization error can differ arbitrarily much; the constants of the equivalence matter and the narrative that depth does not matter in the NTK regime as in Bietti and Bach (2021) is too simplified. More promisingly, analyses that extend our analysis in the infinite-width limit to a joint scaling of width and depth could help us to understand the influence of depth (Fort et al., 2020, Li et al., 2021, Seleznova and Kutyniok, 2022). Finite-sample analyses of moderate-width neural networks with feature learning parametrizations (Yang and Hu, 2021) and other initializations could enable to understand how to induce a spiky-smooth inductive bias in feature learning neural architectures.

Acknowledgments and Disclosure of Funding

Funded by Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) under Germany's Excellence Strategy - EXC 2075 - 390740016 and EXC 2064/1 - Project 390727645, as well as the DFG Priority Program 2298/1, project STE 1074/5-1. The authors thank the International Max Planck Research School for Intelligent Systems (IMPRS-IS) for supporting Moritz Haas and David Holzmüller. We want to thank Tizian Wenzel and Václav Voráček for interesting discussions. We also thank Nadine Große, Jens Wirth, and Daniel Winkle for helpful comments on Sobolev spaces, Max Schölpple for pointing out an error in Lemma B.2, and Nilotpal Sinha for pointing us to Laurent series.

References

Robert A. Adams and John J.F. Fournier. *Sobolev Spaces*. Elsevier Science, 2003.

Michael Aerni, Marco Milanta, Konstantin Donhauser, and Fanny Yang. Strong inductive biases provably prevent harmless interpolation. In *International Conference on Learning Representations (ICLR)*, 2023.

Mikhail S Agranovich. *Sobolev spaces, their generalizations and elliptic problems in smooth and Lipschitz domains*. Springer, 2015.

Ludovic Arnould, Claire Boyer, and Erwan Scornet. Is interpolation benign for random forest regression? In *International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2023.

Sanjeev Arora, Simon S. Du, Wei Hu, Zhiyuan Li, Russ R. Salakhutdinov, and Ruosong Wang. On exact computation with an infinitely wide neural net. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2019.

Yaroslav Averyanov and Alain Celisse. Early stopping and polynomial smoothing in regression with reproducing kernels. *arXiv:2007.06827*, 2020.

Peter L. Bartlett, Philip M. Long, Gábor Lugosi, and Alexander Tsigler. Benign overfitting in linear regression. *Proceedings of the National Academy of Sciences*, 117(48):30063–30070, 2020.

Peter L. Bartlett, Andrea Montanari, and Alexander Rakhlin. Deep learning: a statistical viewpoint. *Acta Numerica*, 30:87–201, 2021.

Daniel Beaglehole, Mikhail Belkin, and Parthe Pandit. On the inconsistency of kernel ridgeless regression in fixed dimensions. *SIAM Journal on Mathematics of Data Science*, 5(4):854–872, 2023.

Mikhail Belkin, Siyuan Ma, and Soumik Mandal. To understand deep learning we need to understand kernel learning. In *International Conference on Machine Learning (ICML)*, 2018.

Mikhail Belkin, Alexander Rakhlin, and Alexandre B. Tsybakov. Does data interpolation contradict statistical optimality? In *International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2019.

Jordan Bell. The singular value decomposition of compact operators on Hilbert spaces, 2014.

Alberto Bietti and Francis Bach. Deep equals shallow for ReLU networks in kernel regimes. In *International Conference on Learning Representations (ICLR)*, 2021.

Simon Buchholz. Kernel interpolation in sobolev spaces is not consistent in low dimensions. In *Conference on Learning Theory (COLT)*, 2022.

Yuan Cao and Quanquan Gu. Generalization bounds of stochastic gradient descent for wide and deep neural networks. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2019.

Bernd Carl and Irmtraud Stephani. *Entropy, Compactness and the Approximation of Operators*. Cambridge University Press, Cambridge, 1990.

Niladri S. Chatterji and Philip M. Long. Finite-sample analysis of interpolating linear classifiers in the overparameterized regime. *Journal of Machine Learning Research (JMLR)*, 2021.

Lin Chen and Sheng Xu. Deep neural tangent kernel and laplace kernel have the same rkhs. In *International Conference on Learning Representations (ICLR)*, 2021.

Zixiang Chen, Yuan Cao, Difan Zou, and Quanquan Gu. How much over-parameterization is sufficient to learn deep ReLU networks? In *International Conference on Learning Representations (ICLR)*, 2021.

Lénaïc Chizat, Edouard Oyallon, and Francis Bach. On lazy training in differentiable programming. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2019.

Fan Chung and Linyuan Lu. Concentration inequalities and martingale inequalities: a survey. *Internet Mathematics*, 3(1):79 – 127, 2006.

Amit Daniely, Roy Frostig, and Yoram Singer. Toward deeper understanding of neural networks: The power of initialization and a dual view on expressivity. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2016.

Ernesto De Vito, Nicole Mücke, and Lorenzo Rosasco. Reproducing kernel hilbert spaces on manifolds: Sobolev and diffusion spaces. *Analysis and Applications*, 19(03):363–396, 2021.

Ronald A DeVore and Robert C Sharpley. Besov spaces on domains in $\mathbb{R}^d$. *Transactions of the American Mathematical Society*, 335(2):843–864, 1993.

Luc Devroye, Laszlo Györfi, and Adam Krzyżak. The Hilbert kernel regression estimate. *Journal of Multivariate Analysis*, 65(2):209–227, 1998.

Eleonora Di Nezza, Giampiero Palatucci, and Enrico Valdinoci. Hitchhiker's guide to the fractional Sobolev spaces. *Bulletin des Sciences Mathematiques*, 136(5):521–573, 2012.

Konstantin Donhauser, Mingqi Wu, and Fanny Yang. How rotational invariance of common kernels prevents generalization in high dimensions. In *International Conference on Machine Learning (ICML)*, 2021.

Konstantin Donhauser, Nicolò Ruggeri, Stefan Stojanovic, and Fanny Yang. Fast rates for noisy interpolation require rethinking the effect of inductive bias. In *International Conference on Machine Learning (ICML)*, 2022.

Lutz Duembgen. Bounding standard gaussian tail probabilities. *arXiv:1012.2063*, 2010.

David E. Edmunds and Hans Triebel. *Function Spaces, Entropy Numbers, Differential Operators*. Cambridge University Press, Cambridge, 1996.

Simon Fischer and Ingo Steinwart. Sobolev norm learning rates for regularized least-squares algorithms. *Journal of Machine Learning Research (JMLR)*, 2020.

Stanislav Fort, Gintare Karolina Dziugaite, Mansheej Paul, Sepideh Kharaghani, Daniel M Roy, and Surya Ganguli. Deep learning versus kernel learning: an empirical study of loss landscape geometry and the time evolution of the neural tangent kernel. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

Spencer Frei, Niladri S Chatterji, and Peter Bartlett. Benign overfitting without linearity: Neural network classifiers trained by gradient descent for noisy linear data. In *Conference on Learning Theory (COLT)*, 2022.

Semjon Gerschgorin. Über die Abgrenzung der Eigenwerte einer Matrix. *Izv. Akad. Nauk. USSR Otd. Fiz.-Mat. Nauk*, 1931.

Behrooz Ghorbani, Song Mei, Theodor Misiakiewicz, and Andrea Montanari. Linearized two-layers neural networks in high dimension. *The Annals of Statistics*, 49(2):1029 – 1054, 2021.

Nikhil Ghosh and Mikhail Belkin. A universal trade-off between the model size, test loss, and training loss of linear predictors. *arXiv:2207.11621*, 2022.

Tilmann Gneiting. Strictly and non-strictly positive definite functions on spheres. *Bernoulli*, 19(4): 1327–1349, 2013.

Trevor Hastie, Andrea Montanari, Saharon Rosset, and Ryan J. Tibshirani. Surprises in high-dimensional ridgeless least squares interpolation. *The Annals of Statistics*, 50(2):949 – 986, 2022.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *IEEE international conference on computer vision (ICCV)*, 2015.

David Holzmüller. On the universality of the double descent peak in ridgeless regression. In *International Conference on Learning Representations (ICLR)*, 2021.

David Holzmüller and Ingo Steinwart. Training two-layer relu networks with gradient descent is inconsistent. *Journal of Machine Learning Research (JMLR)*, 23(181):1–82, 2022.

Roger A. Horn and Charles R. Johnson. *Matrix Analysis*. Cambridge University Press, 2013.

Tianyang Hu, Wenjia Wang, Cong Lin, and Guang Cheng. Regularization matters: A nonparametric perspective on overparametrized neural network. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2021.

Simon Hubbert, Quoc Le Gia, and Tanya Morton. *Spherical Radial Basis Functions, Theory and Applications*. Springer International Publishing, 2015.

Simon Hubbert, Emilio Porcu, Chris J. Oates, and Mark Girolami. Sobolev spaces, kernels and discrepancies over hyperspheres. *Transactions on Machine Learning Research (TMLR)*, 2023.

Arthur Jacot, Franck Gabriel, and Clément Hongler. Neural Tangent Kernel: Convergence and generalization in neural networks. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2018.

Ziwei Ji, Justin D. Li, and Matus Telgarsky. Early-stopped neural networks are consistent. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2021.

Fredrik Johansson et al. *mpmath: a Python library for arbitrary-precision floating-point arithmetic (version 1.3.0)*, 2023. `https://mpmath.org/`.

Motonobu Kanagawa, Philipp Hennig, Dino Sejdinovic, and Bharath K. Sriperumbudur. Gaussian processes and kernel methods: A review on connections and equivalences. *arXiv:1805.08845v1*, 2018.

Jianfa Lai, Manyun Xu, Rui Chen, and Qian Lin. Generalization ability of wide neural networks on $\mathbb{R}$. *arXiv:2302.05933*, 2023.

Béatrice Laurent and Pascal Massart. Adaptive estimation of a quadratic functional by model selection. *The Annals of Statistics*, 28(5):1302 – 1338, 2000.

Jaehoon Lee, Yasaman Bahri, Roman Novak, Samuel S. Schoenholz, Jeffrey Pennington, and Jascha Sohl-Dickstein. Deep neural networks as Gaussian processes. In *International Conference on Learning Representations (ICLR)*, 2018.

Jaehoon Lee, Lechao Xiao, Samuel Schoenholz, Yasaman Bahri, Roman Novak, Jascha Sohl-Dickstein, and Jeffrey Pennington. Wide neural networks of any depth evolve as linear models under gradient descent. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2019.

Mufan Li, Mihai Nica, and Dan Roy. The future is log-gaussian: Resnets and their infinite-depth-and-width limit at initialization. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2021.

Yicheng Li, Haobo Zhang, and Qian Lin. Kernel interpolation generalizes poorly. *Biometrika*, 2023.

Tengyuan Liang and Alexander Rakhlin. Just interpolate: Kernel "ridgeless" regression can generalize. *Annals of Statistics*, 48(3):1329–1347, 2020.

Tengyuan Liang and Benjamin Recht. Interpolating classifiers make few mistakes. *Journal of Machine Learning Research (JMLR)*, 24(20):1–27, 2023.

Tengyuan Liang, Alexander Rakhlin, and Xiyu Zhai. On the multiple descent of minimum-norm interpolants and restricted lower isometry of kernels. In *Conference on Learning Theory (COLT)*, 2020.

Jacques Louis Lions and Enrico Magenes. *Non-homogeneous boundary value problems and applications: Vol. 1*. Springer Science & Business Media, 2012.

Neil Rohit Mallinar, James B Simon, Amirhesam Abedsoltan, Parthe Pandit, Misha Belkin, and Preetum Nakkiran. Benign, tempered, or catastrophic: Toward a refined taxonomy of overfitting. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2022.

Alexander G. de G. Matthews, Jiri Hron, Mark Rowland, Richard E. Turner, and Zoubin Ghahramani. Gaussian process behaviour in wide deep neural networks. In *International Conference on Learning Representations (ICLR)*, 2018.

Andrew D. Mcrae, Santhosh Karnik, Mark Davenport, and Vidya K. Muthukumar. Harmless interpolation in regression and classification with structured features. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2022.

Robert E. Megginson. *An Introduction to Banach Space Theory*. Springer-Verlag, New York, 1998.

Leon Mirsky. On the trace of matrix products. *Mathematische Nachrichten*, 20(3-6):171–174, 1959.

Andrea Montanari, Feng Ruan, Youngtak Sohn, and Jun Yan. The generalization error of max-margin linear classifiers: High-dimensional asymptotics in the overparametrized regime. *arXiv:1911.01544*, 2019.

Vidya Muthukumar, Adhyyan Narang, Vignesh Subramanian, Mikhail Belkin, Daniel Hsu, and Anant Sahai. Classification vs regression in overparameterized regimes: Does the loss function matter? *JMLR*, 22(1):10104–10172, 2021.

Nicole Mücke and Ingo Steinwart. Global minima of DNNs: The plenty pantry. *arXiv:1905.10686*, 2019.

Radford M. Neal. *Priors for Infinite Networks*. Springer New York, 1996.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2019.

Albrecht Pietsch. *Eigenvalues and s-Numbers*. Geest & Portig K.-G., Leipzig, 1987.

Iosif Pinelis. Exact lower and upper bounds on the incomplete gamma function. *Mathematical Inequalities & Applications*, 23(4):1261–1278, 2020.

Alexander Rakhlin and Xiyu Zhai. Consistency of interpolation with Laplace kernels is a high-dimensional phenomenon. In *Conference on Learning Theory (COLT)*, 2019.

Akshay Rangamani, Lorenzo Rosasco, and Tomaso Poggio. For interpolating kernel machines, minimizing the norm of the erm solution maximizes stability. *Analysis and Applications*, 21:193 – 215, 2023.

Garvesh Raskutti, Martin J. Wainwright, and Bin Yu. Early stopping and non-parametric regression: An optimal data-dependent stopping rule. *Journal of Machine Learning Research (JMLR)*, 15(11): 335–366, 2014.

Carl Edward Rasmussen and Christopher K. I. Williams. *Gaussian Processes for Machine Learning*. MIT Press, 2005.

Hans Richter. Zur Abschätzung von Matrizennormen. *Mathematische Nachrichten*, 18(1-6):178–187, 1958.

Robert Schaback. Superconvergence of kernel-based interpolation. *Journal of Approximation Theory*, 235:1–19, 2018.

Cornelia Schneider and Nadine Große. Sobolev spaces on Riemannian manifolds with bounded geometry: General coordinates traces. *Mathematische Nachrichten*, 286(16), 2013.

Mariia Seleznova and Gitta Kutyniok. Neural tangent kernel beyond the infinite-width limit: Effects of depth and initialization. In *International Conference on Machine Learning (ICML)*, 2022.

Ohad Shamir. The implicit bias of benign overfitting. In *Conference on Learning Theory (COLT)*, 2022.

James Benjamin Simon, Sajant Anand, and Mike Deweese. Reverse engineering the neural tangent kernel. In *International Conference on Machine Learning (ICML)*, 2022.

Ingo Steinwart. Consistency of support vector machines and other regularized kernel machines. *IEEE Transactions on Information Theory*, 2001.

Ingo Steinwart. A short note on the comparison of interpolation widths, entropy numbers, and Kolmogorov widths. *J. Approx. Theory*, 215:13–27, 2017.

Ingo Steinwart and Andreas Christmann. *Support Vector Machines*. Springer New York, 2008.

Ingo Steinwart and Clint Scovel. Mercer's theorem on general domains: on the interaction between measures, kernels, and RKHSs. *Constr. Approx.*, 35:363–417, 2012.

Ingo Steinwart, Don R. Hush, and Clint Scovel. Optimal rates for regularized least squares regression. In *Conference on Learning Theory (COLT)*, 2009.

Daniel W. Stroock et al. *Essentials of integration theory for analysis*. Springer, 2011.

Sattar Vakili, Michael Bromberg, Jezabel Garcia, Da-shan Shiu, and Alberto Bernacchia. Uniform generalization bounds for overparameterized neural networks. *arXiv:2109.06099*, 2021.

Ernesto Araya Valdivia. Relative concentration bounds for the spectrum of kernel matrices. *arXiv:1812.02108*, 2018.

Roman Vershynin. *High-dimensional probability: An introduction with applications in data science*. Cambridge University Press, 2018.

Guorong Wang, Yimin Wei, Sanzheng Qiao, Peng Lin, and Yuzhuo Chen. *Generalized Inverses: Theory and Computations*. Springer, 2018.

Yutong Wang and Clayton Scott. Consistent interpolating ensembles via the manifold-Hilbert kernel. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2022.

Yuting Wei, Fanny Yang, and Martin J Wainwright. Early stopping for kernel boosting algorithms: A general analysis with localized complexities. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2017.

Holger Wendland. *Scattered Data Approximation*. Cambridge University Press, 2005.

Xingyu Xu and Yuantao Gu. Benign overfitting of non-smooth neural networks beyond lazy training. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2023.

Greg Yang and Edward J. Hu. Tensor programs iv: Feature learning in infinite-width neural networks. In *International Conference on Machine Learning (ICML)*, 2021.

Yuan Yao, Lorenzo Rosasco, and Andrea Caponnetto. On early stopping in gradient descent learning. *Constructive Approximation*, 26:289–315, 2007.

Chiyuan Zhang, Samy Bengio, Moritz Hardt, Benjamin Recht, and Oriol Vinyals. Understanding deep learning (still) requires rethinking generalization. *Communications of the ACM*, 64(3):107–115, 2021.

Yaoyu Zhang, Zhi-Qin John Xu, Tao Luo, and Zheng Ma. A type of generalization error induced by initialization in deep neural networks. In *Proceedings of The First Mathematical and Scientific Machine Learning Conference*, 2020.

Appendices

Appendix Contents.

A Detailed related work **17**

B Kernels and Sobolev spaces on the sphere **19**

B.1 Background on Sobolev spaces 19

B.2 General kernel theory and notation 19

B.3 Dot-product kernels on the sphere 20

B.4 Neural kernels 21

C Gradient flow and gradient descent with kernels **22**

C.1 Derivation of gradient flow and gradient descent 22

C.2 Gradient flow and gradient descent initialized at 0 have monotonically growing $\mathcal{H}$-norm 23

D Proof of Theorem 1 **24**

D.1 Auxiliary results for the proof of Theorem 1 28

E Translating between $\mathbb{R}^d$ and $\mathbb{S}^d$ **32**

F Spectral lower bound **35**

F.1 General lower bounds 35

F.2 Equivalences of norms and eigenvalues 38

F.3 Kernel matrix eigenvalue bounds 41

F.4 Spectral lower bound for dot-product kernels on the sphere 43

G Proof of Theorem 8 **45**

G.1 Auxiliary results for the proof of Theorem 8 50

G.2 RKHS norm bounds 51

H Spiky-smooth activation functions induced by Gaussian components **52**

I Additional experimental results **56**

I.1 Experimental details of Figure 1 56

I.2 Disentangling signal from noise in neural networks with spiky-smooth activation functions 56

I.3 Repeating the finite-sample experiments 58

I.4 Spiky-smooth activation functions 58

I.5 Isolated spike activation functions 60

I.6 Additive decomposition and sin-fit 62

I.7 Spiky-smooth kernel hyper-parameter selection 63

A Detailed related work

Motivated by Zhang et al. (2021) and Belkin et al. (2018), an abundance of papers have tried to grasp when and how benign overfitting occurs in different settings. Rigorous understanding is mainly restricted to linear (Bartlett et al., 2020), feature (Hastie et al., 2022) and kernel regression (Liang and Rakhlin, 2020) under restrictive distributional assumptions. In the well-specified linear setting under additional assumptions, the minimum-norm interpolant is consistent if and only if $k \ll n \ll d$, the top-k eigendirections of the covariate covariance matrix align with the signal, followed by sufficiently many 'quasi-isotropic' directions with eigenvalues of similar magnitude (Bartlett et al., 2020).

Kernel methods. The analysis of kernel methods is more nuanced and depends on the interplay between the chosen kernel, the choice of regularization and the data distribution. L_2-generalization error bounds can be derived in the eigenbasis of the kernel's integral operator (Mcrae et al., 2022), where upper bounds of the form $\sqrt{\boldsymbol{y}^\top k(\boldsymbol{X},\boldsymbol{X})^{-1}\boldsymbol{y}/n}$ promise good generalization when the regression function f^* is aligned with the dominant eigendirections of the kernel, or in other words, when $\|f^*\|_{\mathcal{H}}$ is small. Most recent work focuses on high-dimensional limits, where the data dimensionality $d \to \infty$. For $d \to \infty$, the Hilbert space and its norm change, so that consistency results that demand bounded Hilbert norm of rotation-invariant kernels do not even include simple functions like sparse products (Donhauser et al., 2021, Lemma 2.1). In the regime $d^{l+\delta} \leq n \leq d^{l+1-\delta}$, rotation-invariant (neural) kernel methods (Ghorbani et al., 2021, Donhauser et al., 2021) can in fact only learn the polynomial parts up to order l of the regression function f^*, and fully-connected NTKs do so. Liang et al. (2020) uncover a related multiple descent phenomenon in kernel regression, where the risk vanishes for most $n \to \infty$, but peaks at $n = d^i$ for all $i \in \mathbb{N}$. The slower d grows, the slower the optimal rate $n^{-\frac{1}{2i+1}}$ between the peaks. Note, however, that these bounds are only upper bounds, and whether they are optimal remains an open question to the best of our knowledge. Another recent line of work analyzes how different inductive biases, measured in $\|\cdot\|_p$-norm minimization, $p \in [1,2]$, (Donhauser et al., 2022) or in the filter size of convolutional kernels (Aerni et al., 2023), affects the generalization properties of minimum-norm interpolants. While the risk on noiseless training samples (bias) decreases with decreasing p or small filter size, the sensitivity to noise in the training data (variance) increases. Hence only 'weak inductive biases', that is large p or large filter sizes, enable harmless interpolation. Our results suggest that to achieve harmless interpolation in fixed dimension one has to construct and minimize more unusual norms than $\|\cdot\|_p$-norms.

Regularised kernel regression achieves optimal rates. With appropriate regularization, kernel ridge regularization with typical universal kernels like the Gauss, Matérn, and Laplace kernels is universally consistent (Steinwart and Christmann, 2008, Chapter 9). Steinwart et al. (2009, Corollary 6) even implies minimax optimal nonparametric rates for clipped kernel ridge regression with Sobolev kernels and $f^* \in H^\beta$ where $d/2 < \beta \leq s$ for the choice $\rho_n = n^{-2s/(2\beta+d)}$. Although f^* is not necessarily in the RKHS, KRR is adaptive and can still achieve optimal learning rates. Lower smoothness β of f^* as well as higher smoothness of the kernel should be met with faster decay of ρ_n. Optimal rates in Sobolev RKHS can also be achieved using cross-validation of the regularization ρ (Steinwart et al., 2009), early stopping rules based on empirical localized Rademacher (Raskutti et al., 2014) or Gaussian complexity (Wei et al., 2017) or smoothing of the empirical risk via kernel matrix eigenvalues (Averyanov and Celisse, 2020).

Lower bounds for kernel regression. Besides Rakhlin and Zhai (2019) and Buchholz (2022), Beaglehole et al. (2023) derive inconsistency results for kernel ridgeless regression given assumptions on the spectral tail in the Fourier basis. Mallinar et al. (2022) provide a characterization of kernel ridge regression into benign, tempered and catastrophic overfitting using a *heuristic* approximation of the risk via the kernel's eigenspectrum, essentially assuming that the eigenfunctions can be replaced by structureless Gaussian random variables. A general lower bound for ridgeless linear regression Holzmüller (2021) predicts bad generalization near the "interpolation threshold", where the dimension of the feature space is close to n, also known as the *double descent* phenomenon. In this regime, Ghosh and Belkin (2022) also consider overfitting by a fraction beyond the noise level and derive a lower bound for linear models.

Benign overfitting in fixed dimension. Only few works have established consistency results for interpolating models in fixed dimension. The first statistical guarantees for Nadaraya-Watson kernel

smoothing with singular kernels were given by Devroye et al. (1998). Optimal non-asymptotic results have only been established more recently. Belkin et al. (2019) show that Nadaraya-Watson kernel smoothing achieves minimax optimal convergence rates for $a \in (0, d/2)$ under smoothness assumptions on f^*, when using singular kernels such as truncated Hilbert kernels $K(u) = \|u\|_2^a \mathbb{1}_{\|u\| \leq 1}$, which do not induce RKHS that only contain weakly differentiable functions (as our results do). By thresholding the kernel they can adjust the amount of overfitting without affecting the generalization bound. To the best of our knowledge, rigorously proving or disproving analogous bounds for kernel ridge regression remains an open question. Arnould et al. (2023) show that median random forests are able to interpolate consistently in fixed dimension because of an averaging effect introduced through feature randomization. They conjecture consistent interpolation for Breiman random forests based on numerical experiments.

Classification. For binary classification tasks, benign overfitting is a more generic phenomenon than for regression tasks (Muthukumar et al., 2021, Shamir, 2022). Consistency has been shown under linear separability assumptions (Montanari et al., 2019, Chatterji and Long, 2021, Frei et al., 2022) and through complexity bounds with respect to reference classes like the 'Neural Tangent Random Feature' model (Cao and Gu, 2019, Chen et al., 2021). Most recently, Liang and Recht (2023) have shown that the 0-1-generalization error of minimum RKHS-norm interpolants $\hat{f}_0$ is upper bounded by $\frac{\|\hat{f}_0\|_{\mathcal{H}}^2}{n}$ and analogously that kernel ridge regression $\hat{f}_\rho$ generalizes as $\frac{\boldsymbol{y}^\top (k(\boldsymbol{X},\boldsymbol{X})+\rho \boldsymbol{I})^{-1} \boldsymbol{y}}{n}$, where the numerator upper bounds $\|\hat{f}_\rho\|_{\mathcal{H}}^2$. Their bounds imply consistency as long as the total variation distance between the class conditionals is sufficiently large and the regression function has bounded RKHS-norm, and their Lemma 7 shows that the upper bound is rate optimal. Under a noise condition on the regression function $f^*(\boldsymbol{x}) = \mathbb{E}[y|\boldsymbol{x}]$ for binary classification and bounded $\|f^*\|_{\mathcal{H}}$, our results together with Liang and Recht (2023) reiterate the distinction between benign overfitting for binary classification and inconsistent overfitting for least squares regression for a large class of distributions in kernel regression over Sobolev RKHS. Chapter 8 of Steinwart and Christmann (2008) discusses how the overlap of the two classes may influence learning rates under positive regularization. Using Nadaraya-Watson kernel smoothing, Wang and Scott (2022) offer the first consistency result for a simple interpolating ensemble method with data-independent base classifiers.

Connection to neural networks. It is known that neural networks can behave like kernel methods in certain infinite-width limits. For example, the function represented by a randomly initialized NN behaves like a Gaussian process with the NN Gaussian process (NNGP) kernel, which depends on details such as the activation function and depth of the NN (Neal, 1996, Lee et al., 2018, Matthews et al., 2018). Hence, Bayesian inference in infinitely wide NNs is GP regression, whose posterior predictive mean function is of the form $f_{\infty,\rho}$, where ρ depends on the assumed noise variance. Moreover, gradient flow training of certain infinitely wide NNs is similar to gradient flow training with the so-called *neural tangent kernel* (NTK) (Jacot et al., 2018, Lee et al., 2019, Arora et al., 2019), and the correspondence can be made exact using small modifications to the NN to remove the stochastic effect of the random initial function (Arora et al., 2019, Zhang et al., 2020). In other words, certain infinitely wide NNs trained with gradient flow learn functions of the form $f_{t,0}$.

When considering the sphere $\Omega = \mathbb{S}^d$, the NTK and NNGP kernels of fully-connected NNs are dot-product kernels, i.e., $k(\boldsymbol{x}, \boldsymbol{x}') = \kappa(\langle \boldsymbol{x}, \boldsymbol{x}' \rangle)$ for some function $\kappa : [-1, 1] \to \mathbb{R}$. Moreover, from Bietti and Bach (2021) and Chen and Xu (2021) it follows that the RKHSs of typical NTK and NNGP kernels for the ReLU activation function are equivalent to the Sobolev spaces $H^{(d+1)/2}(\mathbb{S}^d)$ and $H^{(d+3)/2}(\mathbb{S}^d)$, respectively, cf Appendix B.4.

Regarding consistency, Ji et al. (2021) use the NTK correspondence to show that early-stopped wide NNs for classification are universally consistent under some assumptions. On the other hand, Holzmüller and Steinwart (2022) show that zero-initialized biases can prevent certain two-layer ReLU NNs from being universally consistent. Lai et al. (2023) show an inconsistency-type result for overfitting two-layer ReLU NNs with $d = 1$, but for fixed inputs $\boldsymbol{X}$. They also note that an earlier inconsistency result by Hu et al. (2021) relies on an unproven result. Li et al. (2023) show that consistency with polynomial convergence rates is impossible for minimum-norm interpolants of common kernels including ReLU NTKs. Mallinar et al. (2022) conjecture tempered overfitting and therefore inconsistency for interpolation with ReLU NTKs based on their semi-rigorous result and the results of Bietti and Bach (2021) and Chen and Xu (2021). Xu and Gu (2023) establish consistency of overfitting wide 2-layer neural networks beyond the NTK regime for binary classification in very

high dimension $d = \Omega(n^2)$ and for a quite restricted class of distributions (the mean difference μ of the class conditionals needs to fulfill $\mu = \Omega((d/n)^{1/4} \log^{1/4}(md/n))$ and $\mu = O((d/n)^{1/2})$).

B Kernels and Sobolev spaces on the sphere

B.1 Background on Sobolev spaces

We say that two Hilbert spaces $\mathcal{H}_1, \mathcal{H}_2$ are equivalent, written as $\mathcal{H}_1 \cong \mathcal{H}_2$, if they are equal as sets and the corresponding norms $\|\cdot\|_{\mathcal{H}_1}$ and $\|\cdot\|_{\mathcal{H}_2}$ are equivalent.

Let Ω be an open set with C^∞ boundary. In this paper, we will mainly consider ℓ_2-balls for Ω. There are multiple equivalent ways to define a (fractional) Sobolev space $H^s(\Omega)$, $s \in \mathbb{R}_{\geq 0}$, these are equivalent in the sense that the resulting Hilbert spaces will be equivalent. For example, $H^s(\Omega)$ can be defined through restrictions of functions from $H^s(\mathbb{R}^d)$, through interpolation spaces, or through Sobolev-Slobodetski norms (see e.g. Chapter 5 and 14 in Agranovich, 2015 and Chapters 7–10 in Lions and Magenes, 2012). Some requirements on Ω can be relaxed, for example to Lipschitz domains, by using more general extension operators (e.g. DeVore and Sharpley, 1993). Since our results are based on equivalent norms and not specific norms, we do not care which of these definitions is used. Further background on Sobolev spaces can be found in Adams and Fournier (2003), Wendland (2005) and Di Nezza et al. (2012).

B.2 General kernel theory and notation

There is a one-to-one correspondence between kernel functions k and the corresponding reproducing kernel Hilbert spaces (RKHS) $\mathcal{H}_k$. Mercer's theorem (Steinwart and Christmann, 2008, Theorem 4.49) states that for compact Ω, continuous k and a Borel probability measure P_X on Ω whose support is Ω, the integral operator $T_{k,P_X} : L_2(P_X) \to L_2(P_X)$ given by

$$T_{k,P_X} f(\boldsymbol{x}) = \int_\Omega f(\boldsymbol{x}') k(\boldsymbol{x}, \boldsymbol{x}') dP_X(\boldsymbol{x}'),$$

can be decomposed into an orthonormal basis $(e_i)_{i \in I}$ of $L_2(P_X)$ and corresponding eigenvalues $(\lambda_i)_{i \in I} \geq 0$, $\lambda_l \searrow 0$, such that

$$T_{k,P_X} f = \sum_{i \in I} \lambda_i \langle f, e_i \rangle e_i, \qquad f \in L_2(P_X).$$

We write $\lambda_i(T_{k,P_X}) \coloneqq \lambda_i$. Moreover, $k(\boldsymbol{x}, \boldsymbol{x}') = \sum_{i \in I} \lambda_i e_i(\boldsymbol{x}) e_i(\boldsymbol{x}')$ converges absolutely and uniformly, and the RKHS is given by

$$\mathcal{H}_k = \left\{ \sum_{i \in I} a_i \sqrt{\lambda_i} e_i \;\middle|\; \sum_{i \in I} a_i^2 < \infty \right\}. \tag{B.1}$$

The corresponding inner product between $f = \sum_{i \in I} a_i \sqrt{\lambda_i} e_i \in \mathcal{H}$ and $g = \sum_{i \in I} b_i \sqrt{\lambda_i} e_i \in \mathcal{H}$ can then be written as

$$\langle f, g \rangle_{\mathcal{H}} = \sum_{i \in I} a_i b_i. \tag{B.2}$$

We use asymptotic notation O, Ω, Θ for integers n in the following way: We write

$$\begin{aligned}
&f(n) = O(g(n)) \Leftrightarrow \exists C > 0 \forall n : f(n) \leq C g(n) \\
&f(n) = \Omega(g(n)) \Leftrightarrow g(n) = O(f(n)) \\
&f(n) = \Theta(g(n)) \Leftrightarrow f(n) = O(g(n)) \text{ and } g(n) = O(f(n)) \,.
\end{aligned}$$

Above, we require that the inequality $f(n) \leq C g(n)$ holds for all n and not only for $n \geq n_0$. This implies that if $f(n) = \Omega(g(n))$, then f must be nonzero whenever g is nonzero. This is an important detail when arguing about equivalence of RKHSs, since it allows the following statement: If we have two kernels $k, \tilde{k}$ with Mercer representations

$$k(\boldsymbol{x}, \boldsymbol{x}') = \sum_{i \in I} \lambda_i e_i(\boldsymbol{x}) e_i(\boldsymbol{x}')$$

$$\tilde{k}(\boldsymbol{x}, \boldsymbol{x}') = \sum_{i \in I} \tilde{\lambda}_i e_i(\boldsymbol{x}) e_i(\boldsymbol{x}')$$

with identical eigenfunctions e_i and eigenvalues satisfying $\lambda_i = \Theta(\tilde{\lambda}_i)$, then the associated RKHSs are equivalent by (B.1) and (B.2).

B.3 Dot-product kernels on the sphere

A kernel of the form $k(\boldsymbol{x}, \boldsymbol{x}') = \kappa(\langle \boldsymbol{x}, \boldsymbol{x}' \rangle)$ for some function κ is called *dot-product kernel*. Dot-product kernels are rotationally invariant. Especially, NTKs and NNGPs of fully-connected NNs restricted to the sphere $\mathbb{S}^d$ are dot-product kernels. Moreover, kernels like the Laplace, Matérn, and Gaussian kernels that only depend on the distance between their inputs are also dot-product kernels when restricted to the sphere $\mathbb{S}^d$. Therefore, in this section, we will assume that $k : \mathbb{S}^d \times \mathbb{S}^d \to \mathbb{R}$ is a dot-product kernel.

We can then leverage some convenient results from the theory of dot-product kernels on the sphere, which are summarized in more detail by Hubbert et al. (2023). Let $\{Y_{l,1}, \ldots, Y_{l,N_{l,d}}\}$ be a real orthonormal basis for the space of spherical harmonics of degree l within $L_2(\mathbb{S}^d)$. Moreover, let ω_d be the surface area of $\mathbb{S}^d$, then the $\tilde{Y}_{l,i} \coloneqq \sqrt{\omega_d} Y_{l,i}$ form a corresponding orthonormal basis w.r.t. the uniform distribution $\mathcal{U}(\mathbb{S}^d)$. Then, a Mercer representation of k is given by

$$k(\boldsymbol{x}, \boldsymbol{x}') = \sum_{l=0}^{\infty} \mu_l \sum_{i=1}^{N_{l,d}} Y_{l,i}(\boldsymbol{x}) Y_{l,i}(\boldsymbol{x}') = \sum_{l=0}^{\infty} \tilde{\mu}_l \sum_{i=1}^{N_{l,d}} \tilde{Y}_{l,i}(\boldsymbol{x}) \tilde{Y}_{l,i}(\boldsymbol{x}') \,,$$

with $\tilde{\mu}_l = \mu_l / \omega_d$. Especially, the integral operator $T_{k,\mathcal{U}(\mathbb{S}^d)}$ for the uniform distribution $\mathcal{U}(\mathbb{S}^d)$ has eigenvalues $\tilde{\mu}_l$ with multiplicity $N_{l,d}$ and eigenfunctions $\tilde{Y}_{l,i}$. The RKHS of k is then given by

$$\mathcal{H}_k = \left\{ \sum_{l=0}^{\infty} \sqrt{\mu_l} \sum_{i=1}^{N_{l,d}} a_{l,i} Y_{l,i} \;\middle|\; \sum_{l=0}^{\infty} \sum_{i=1}^{N_{l,d}} a_{l,i}^2 < \infty \right\} \,.$$

Since the index l can be zero, we will denote decay asymptotics for l in the form $\Theta((l+1)^{-q})$ and not $\Theta(l^{-q})$, cf. our definition of Θ notation in Appendix B.2.

Lemma B.1 (Sobolev dot-product kernels on the sphere)**.** *For a dot-product kernel k on $\mathbb{S}^d$ as above, the RKHS $\mathcal{H}_k$ is equivalent to the Sobolev space $H^s(\mathbb{S}^d)$, $s > d/2$, if and only if $\mu_l = \Theta((l+1)^{-2s})$. In this case, we have*

$$\lambda_i(T_{k,\mathcal{U}(\mathbb{S}^d)}) = \Theta(i^{-2s/d}) \,.$$

Proof. **Step 0: Equivalence.** If $\mu_l = \Theta((l+1)^{-2s})$, it is stated in Section 3 in Hubbert et al. (2023) that $\mathcal{H}_k \cong H^s(\mathbb{S}^d)$. On the other hand, if $\mu_l \neq \Theta((l+1)^{-2s})$, it is easy to see that $\mathcal{H}_k$ is not equivalent to the RKHS of a kernel with $\mu_l = \Theta((l+1)^{-2s})$. It remains to show $\lambda_i(T_{k,\mathcal{U}(\mathbb{S}^d)}) = \Theta(i^{-2s/d})$.

Step 1: Ordering the eigenvalues. Consider a permutation $\pi : \mathbb{N}_0 \to \mathbb{N}_0$ such that

$$\tilde{\mu}_{\pi(0)} \geq \tilde{\mu}_{\pi(1)} \geq \cdots$$

We can then define the partial sums

$$S_l \coloneqq \sum_{i=0}^{l} N_{\pi(i),d} \,.$$

For $S_{l-1} < i \leq S_l$, we then have $\lambda_i(T_{k,\mathcal{U}(\mathbb{S}^d)}) = \tilde{\mu}_{\pi(l)}$.

Step 2: Show $\pi(i) = \Theta(i)$. Let $c, C > 0$ such that $c(l+1)^{-2s} \leq \tilde{\mu}_l \leq C(l+1)^{-2s}$ for all $l \in \mathbb{N}_0$. For indices $i, j \in \mathbb{N}_0$, we have the implications

$$\begin{aligned} i > j &\Rightarrow c(\pi(i)+1)^{-2s} \leq \tilde{\mu}_{\pi(i)} \leq \tilde{\mu}_{\pi(j)} \leq C(\pi(j)+1)^{-2s} \\ &\Rightarrow \pi(i) + 1 \geq \left(\frac{c}{C}\right)^{1/(2s)} (\pi(j)+1) \,. \end{aligned}$$

Therefore, for $i \geq 1$ and $j \geq 0$,

$$\pi(i) + 1 \geq \left(\frac{c}{C}\right)^{1/(2s)} \max_{i'<i}(\pi(i') + 1) \geq \left(\frac{c}{C}\right)^{1/(2s)} ((i-1)+1) \geq \Omega(i+1) ,$$
$$\pi(j) + 1 \leq \left(\frac{C}{c}\right)^{1/(2s)} \min_{j'>j}(\pi(j') + 1) \leq \left(\frac{C}{c}\right)^{1/(2s)} ((j+1)+1) \leq O(j+1) .$$

We can thus conclude that $\pi(i) + 1 = \Theta(i+1)$.

Step 3: Individual Eigenvalue decay. As explained in Section 2.1 in Hubbert et al. (2023), we have $N_{l,d} = \Theta((l+1)^{d-1})$. Therefore,

$$S_l = \sum_{i=0}^{l} \Theta((\pi(i)+1)^{d-1}) = \sum_{i=0}^{l} \Theta((i+1)^{d-1}) = \Theta((l+1)^d) .$$

Now, let $i \geq 1$ and let $l \in \mathbb{N}_0$ such that $S_{l-1} < i \leq S_l$. We have $i \geq \Omega(l^d)$, and $i \leq O((l+1)^d)$, which implies $i = \Theta((l+1)^d)$ since $i \geq 1$. Therefore,

$$\lambda_i = \tilde{\mu}_{\pi(l)} = \Theta((\pi(l)+1)^{-2s}) = \Theta((l+1)^{-2s}) = \Theta\left(i^{-2s/d}\right) . \qquad \square$$

B.4 Neural kernels

Several NTK and NNGP kernels have RKHSs that are equivalent to Sobolev spaces on $\mathbb{S}^d$. In the following cases, we can deduct this from known results:

- Consider fully-connected NNs with $L \geq 3$ layers without biases and the activation function $\varphi(x) = \max\{0, x\}^m$, $m \in \mathbb{N}_0$. Especially, the case $m = 1$ corresponds to the ReLU activation. Vakili et al. (2021) generalize the result by Bietti and Bach (2021) from $m = 1$ to $m \geq 1$, showing that the NTK-RKHS is equivalent to $H^s(\mathbb{S}^d)$ for $s = (d + 2m - 1)/2$ and the NNGP-RKHS is equivalent to $H^s(\mathbb{S}^d)$ for $s = (d + 2m + 1)/2$. For $m = 0$, Bietti and Bach (2021) essentially show that the NNGP-RKHS is equivalent to $H^s(\mathbb{S}^d)$ for $s = (d + 2^{2-L})/2$. However, all of the aforementioned results have the problem that the main theorem by Bietti and Bach (2021) allows for the possibility that finitely many μ_l are zero, which can change the RKHS. Using our Lemma B.2 below, it follows that all μ_l are in fact nonzero for NNGPs and NTKs since they are kernels in every dimension d using the same function κ independent of the dimension. Hence, the equivalences to Sobolev spaces stated before are correct.
- Chen and Xu (2021) prove that the RKHS of the NTK corresponding to fully-connected ReLU NNs with zero-initialized biases and $L \geq 2$ (as opposed to no biases and $L \geq 3$ above) layers is equivalent to the RKHS of the Laplace kernel on the sphere. Since the Laplace kernel is a Matérn kernel of order $\nu = 1/2$ (see e.g. Section 4.2 in Rasmussen and Williams (2005)), we can use Proposition 5.2 of Hubbert et al. (2023) to obtain equivalence to $H^s(\mathbb{S}^d)$ with $s = (d+1)/2$. Alternatively, we can obtain the RKHS of the Laplace kernel from Bietti and Bach (2021) combined with Lemma B.2.

Bietti and Bach (2021) also show that under an integrability condition on the derivatives, C^∞ activations induce NTK and NNGP kernels whose RKHSs are smaller than every Sobolev space.

Lemma B.2 (Guaranteeing non-zero eigenvalues)**.** *Let $\kappa : [-1, 1] \to \mathbb{R}$ be continuous, let $d \geq 1$, and let*

$$k_d : \mathbb{S}^d \times \mathbb{S}^d \to \mathbb{R}, k_d(\boldsymbol{x}, \boldsymbol{x}') \coloneqq \kappa(\langle \boldsymbol{x}, \boldsymbol{x}' \rangle)$$
$$k_{d+2} : \mathbb{S}^{d+2} \times \mathbb{S}^{d+2} \to \mathbb{R}, k_{d+2}(\boldsymbol{x}, \boldsymbol{x}') \coloneqq \kappa(\langle \boldsymbol{x}, \boldsymbol{x}' \rangle) .$$

Suppose that k_{d+2} is a kernel. Then, k_d is a kernel. Moreover, if the corresponding eigenvalues μ_l of k_d satisfy $\mu_l > 0$ for infinitely many even and infinitely many odd l, then they satisfy $\mu_l > 0$ for all $l \in \mathbb{N}_0$.

Proof. The fact that k_d is a kernel follows directly from the inclusion $\Phi_{d+2} \subseteq \Phi_d$ mentioned in Gneiting (2013). For $D \in \{d, d+2\}$, let $\mu_{l,d}$ be the sequence of eigenvalues μ_l associated with k_D. Then, as mentioned for example by Hubbert et al. (2023), the Schoenberg coefficients $b_{l,d}$ satisfy

$$b_{l,d} = \frac{\Gamma\left(\frac{d+1}{2}\right) N_{m,d} \mu_{l,d}}{2\pi^{(d+1)/2}} .$$

Especially, the Schoenberg coefficients $b_{l,d}$ have the same sign as the eigenvalues $\mu_{l,d}$. We use

$$0 \leq b_{l,d+2} = \begin{cases} b_{l,d} - \frac{1}{2} b_{l+2,d} & , l = 0 \text{ and } d = 1 \\ \frac{1}{2}(l+1)(b_{l,d} - b_{l+2,d}) & , l \geq 1 \text{ and } d = 1 \\ \frac{(l+d-1)(l+d)}{d(2l+d-1)} b_{l,d} - \frac{(l+1)(l+2)}{d(2l+d+3)} b_{l+2,d} & , d \geq 2 \, , \end{cases}$$

where the inequality follows from the fact that k_{d+2} is a kernel and the equality is the statement of Corollary 3 by Gneiting (2013). In any of the three cases, $b_{l+2,d} > 0$ implies $b_{l,d} > 0$. Hence, if $b_{l,d} > 0$ for infinitely many even and infinitely many odd l, then $b_{l,d} > 0$ for all l, which implies $\mu_{l,d} > 0$ for all l. □

C Gradient flow and gradient descent with kernels

C.1 Derivation of gradient flow and gradient descent

Here, we derive expressions for gradient flow and gradient descent in the RKHS for the regularized loss

$$L(f) \coloneqq \frac{1}{n} \sum_{i=1}^{n} (y_i - f(\boldsymbol{x}_i))^2 + \rho \|f\|_{\mathcal{H}_k}^2 = \frac{1}{n} \sum_{i=1}^{n} (y_i - \langle k(\boldsymbol{x}_i, \cdot), f \rangle_{\mathcal{H}_k})^2 + \rho \langle f, f \rangle_{\mathcal{H}_k}^2 \, .$$

Note that we will take derivatives in the RKHS with respect to f, which is different from taking derivatives w.r.t. the coefficients $\boldsymbol{c}$ in a model $f(\boldsymbol{x}) = \boldsymbol{c}^\top k(\boldsymbol{X}, \boldsymbol{x})$.

In the RKHS-Norm, the Fréchet derivative of L is

$$\frac{\partial L(f)}{f} = \frac{2}{n} \sum_{i=1}^{n} (f(\boldsymbol{x}_i) - y_i) \langle k(\boldsymbol{x}_i, \cdot), \cdot \rangle_{\mathcal{H}_k} + 2\rho \langle f, \cdot \rangle_{\mathcal{H}_k} \, ,$$

which is represented in $\mathcal{H}_k$ by

$$L'(f) = \frac{2}{n} \sum_{i=1}^{n} (f(\boldsymbol{x}_i) - y_i) k(\boldsymbol{x}_i, \cdot) + 2\rho f \, .$$

Now assume that $f = \sum_{i=1}^{n} a_i k(\boldsymbol{x}_i, \cdot) = \boldsymbol{a}^\top k(\boldsymbol{X}, \cdot)$. Then,

$$\begin{aligned} L'(f) &= \frac{2}{n} \sum_{i=1}^{n} (\boldsymbol{a}^\top k(\boldsymbol{X}, \boldsymbol{x}_i) - y_i) k(\boldsymbol{x}_i, \cdot) + 2\rho \boldsymbol{a}^\top k(\boldsymbol{X}, \cdot) \\ &= \frac{2}{n} \left(\boldsymbol{a}^\top k(\boldsymbol{X}, \boldsymbol{X}) k(\boldsymbol{X}, \cdot) - \boldsymbol{y}^\top k(\boldsymbol{X}, \cdot) + \rho n \boldsymbol{a}^\top k(\boldsymbol{X}, \cdot) \right) \\ &= \frac{2}{n} \left((k(\boldsymbol{X}, \boldsymbol{X}) + \rho n \boldsymbol{I}_n) \boldsymbol{a} - \boldsymbol{y} \right)^\top k(\boldsymbol{X}, \cdot) \, . \end{aligned}$$

Especially, under gradient flow of f, the coefficients $\boldsymbol{a}$ follow the dynamics

$$\dot{\boldsymbol{a}}(t) = \frac{2}{n} \left(\boldsymbol{y} - (k(\boldsymbol{X}, \boldsymbol{X}) + \rho n \boldsymbol{I}_n) \boldsymbol{a}(t) \right) \, ,$$

which is solved for $\boldsymbol{a}(0) = 0$ by

$$\boldsymbol{a}(t) = \left(\boldsymbol{I}_n - e^{-\frac{2}{n} t (k(\boldsymbol{X}, \boldsymbol{X}) + \rho n \boldsymbol{I}_n)} \right) (k(\boldsymbol{X}, \boldsymbol{X}) + \rho n \boldsymbol{I}_n)^{-1} \boldsymbol{y} \, ,$$

which is the closed form expression (1) of $f_{t,\rho}$.

For gradient descent, assuming that $f_{t,\rho}^{\mathrm{GD}} = \boldsymbol{c}_{t,\rho}^\top k(\boldsymbol{X}, \cdot)$, we have

$$\begin{aligned} f_{t+1,\rho}^{\mathrm{GD}} &= f_{t,\rho}^{\mathrm{GD}} - \eta_t L'(f_{t,\rho}^{\mathrm{GD}}) = \boldsymbol{c}_{t,\rho}^\top k(\boldsymbol{X}, \cdot) - \eta_t \frac{2}{n} \left((k(\boldsymbol{X}, \boldsymbol{X}) + \rho n \boldsymbol{I}_n) \boldsymbol{c}_{t,\rho} - \boldsymbol{y} \right)^\top k(\boldsymbol{X}, \cdot) \\ &= \left(\boldsymbol{c}_{t,\rho} + \eta_t \frac{2}{n} \left(\boldsymbol{y} - (k(\boldsymbol{X}, \boldsymbol{X}) + \rho n \boldsymbol{I}_n) \boldsymbol{c}_{t,\rho} \right) \right)^\top k(\boldsymbol{X}, \cdot) \end{aligned}$$

If $f_{0,\rho}^{\mathrm{GD}} \equiv 0$, the coefficients evolve as $\boldsymbol{c}_0 = \boldsymbol{0}$ and

$$\boldsymbol{c}_{t+1,\rho} = \boldsymbol{c}_{t,\rho} + \eta_t \frac{2}{n} \left(\boldsymbol{y} - (k(\boldsymbol{X}, \boldsymbol{X}) + \rho n \boldsymbol{I}_n) \boldsymbol{c}_{t,\rho} \right) \, .$$

For an analysis of gradient descent for kernel regression with $\rho = 0$, we refer to, e.g., Yao et al. (2007).

C.2 Gradient flow and gradient descent initialized at 0 have monotonically growing $\mathcal{H}$-norm

In the following proposition we show that under gradient flow and gradient descent with sufficiently small learning rates initialized at 0, the RKHS norm grows monotonically with time t. This immediately implies that Assumption (N) with $C_{\mathrm{norm}} = 1$ holds for all estimators $f_{t,\rho}$ from (1).

Proposition C.1.

(i) For any $t \in [0, \infty]$ and any $\rho \geq 0$, $f_{t,\rho}$ from (1) fulfills Assumption (N) with $C_{\mathrm{norm}} = 1$.
(ii) For any $t \in \mathbb{N}_0 \cup \{\infty\}$ and any $\rho \geq 0$, with sufficiently small fixed learning rate $0 \leq \eta \leq \frac{1}{2(\rho + \lambda_{\max}(k(\boldsymbol{X},\boldsymbol{X}))/n)}$, $f^{\mathrm{GD}}_{t,\rho}$ fulfills Assumption (N) with $C_{\mathrm{norm}} = 1$.

Proof. **Proof of (i):**

We write $f_{t,\rho}(\boldsymbol{x}) = k(\boldsymbol{x}, \boldsymbol{X})\boldsymbol{c}_{t,\rho}$, where $\boldsymbol{c}_{t,\rho} \coloneqq A_{t,\rho}(\boldsymbol{X})\boldsymbol{y}$. We now show that the RKHS-norm of $f_{t,\rho}$ grows monotonically in t, by using the eigendecomposition $k(\boldsymbol{X}, \boldsymbol{X}) = \boldsymbol{U}\Lambda\boldsymbol{U}^\top$, where $\Lambda = \mathrm{diag}(\lambda_1, \ldots, \lambda_n) \in \mathbb{R}^{n\times n}$ is diagonal and $\boldsymbol{U} \in \mathbb{R}^{n\times n}$ is orthonormal, and writing $\tilde{\boldsymbol{y}} \coloneqq \boldsymbol{U}^\top \boldsymbol{y}$. Then it holds that

$$\begin{aligned}
\|f_{t,\rho}\|^2_{\mathcal{H}} &= (\boldsymbol{c}_{t,\rho})^\top k(\boldsymbol{X}, \boldsymbol{X})\boldsymbol{c}_{t,\rho} = \tilde{\boldsymbol{y}}^\top (\Lambda + \rho n \boldsymbol{I}_n)^{-1} \left(\boldsymbol{I}_n - \exp\left(-\frac{2t}{n}(\Lambda + \rho n \boldsymbol{I}_n)\right)\right) \Lambda \cdot \\
&\quad \left(\boldsymbol{I}_n - \exp\left(-\frac{2t}{n}(\Lambda + \rho n \boldsymbol{I}_n)\right)\right) (\Lambda + \rho n \boldsymbol{I}_n)^{-1} \tilde{\boldsymbol{y}} \\
&= \sum_{k=1, \lambda_k + \rho n > 0}^{n} \tilde{y}_k^2 \underbrace{\frac{\lambda_k}{(\lambda_k + \rho n)^2}}_{\leq 1/\lambda_k} \underbrace{\left(1 - \exp\left(-\frac{2t}{n}(\lambda_k + \rho n)\right)\right)}_{\leq 1} \\
&\leq \sum_{k=1, \lambda_k > 0}^{n} \tilde{y}_k^2 \frac{1}{\lambda_k} = \|f_{\infty,0}\|^2_{\mathcal{H}}.
\end{aligned}$$

Proof of (ii):

Expanding the iteration in the definition of $\boldsymbol{c}_{t,\rho}$ yields

$$\boldsymbol{c}_{t+1,\rho} = \sum_{i=0}^{t} \prod_{j=0}^{t-i-1} \left(\boldsymbol{I} - \frac{2\eta_{t-j}}{n}(k(\boldsymbol{X}, \boldsymbol{X}) + \rho n \boldsymbol{I})\right) \frac{2\eta_i}{n} \boldsymbol{y}.$$

We again use the eigendecomposition $k(\boldsymbol{X}, \boldsymbol{X}) = \boldsymbol{U}\Lambda\boldsymbol{U}^\top$, where $\Lambda = \mathrm{diag}(\lambda_1, \ldots, \lambda_n) \in \mathbb{R}^{n\times n}$ is diagonal and $\boldsymbol{U} \in \mathbb{R}^{n\times n}$ is orthonormal, and write $\tilde{\boldsymbol{y}} \coloneqq \boldsymbol{U}^\top \boldsymbol{y}$. Then, using sufficiently small learning rates $0 \leq \eta_t \leq \frac{1}{2(\rho + \lambda_{\max}(k(\boldsymbol{X},\boldsymbol{X}))/n)}$ in all time steps $t \in \mathbb{N}$, it holds that

$$\begin{aligned}
&\|f^{\mathrm{GD}}_{t,\rho}\|^2_{\mathcal{H}} \\
&= (\boldsymbol{c}_{t,\rho})^\top k(\boldsymbol{X}, \boldsymbol{X})\boldsymbol{c}_{t,\rho} \\
&= \tilde{\boldsymbol{y}}^\top \left(\sum_{i=0}^{t} \frac{2\eta_i}{n} \prod_{j=0}^{t-i-1} ((1 - 2\eta_{t-j}\rho)\boldsymbol{I} - \frac{2\eta_{t-j}}{n}\Lambda)\right) \Lambda \left(\sum_{i=0}^{t} \frac{2\eta_i}{n} \prod_{j=0}^{t-i-1} ((1 - 2\eta_{t-j}\rho)\boldsymbol{I} - \frac{2\eta_{t-j}}{n}\Lambda)\right) \tilde{\boldsymbol{y}} \\
&= \sum_{k=1}^{n} \underbrace{\tilde{y}_k^2 \lambda_k}_{\geq 0} \left(\sum_{i=0}^{t} \frac{2\eta_i}{n} \prod_{j=0}^{t-i-1} \underbrace{(1 - 2\eta_{t-j}(\rho + \lambda_k/n))}_{\in [0,1]}\right)^2 . \qquad \text{(C.1)}
\end{aligned}$$

The last display shows that $\|f^{\mathrm{GD}}_{t,\rho}\|^2_{\mathcal{H}}$ grows monotonically in t, strictly monotonically if $\eta_t \in (0, \frac{1}{2(\rho + \lambda_{\max}(k(\boldsymbol{X},\boldsymbol{X}))/n)})$ holds for all t. It also shows that if $\rho' \geq \rho$ then $\|f^{\mathrm{GD}}_{t,\rho'}\|_{\mathcal{H}} \leq \|f^{\mathrm{GD}}_{t,\rho}\|_{\mathcal{H}}$ for any $t \in \mathbb{N} \cup \{\infty\}$. To see that $\|f^{\mathrm{GD}}_{t,\rho}\|^2_{\mathcal{H}} \leq \|f_{\infty,0}\|^2_{\mathcal{H}}$ for all $t \in \mathbb{N} \cup \{\infty\}$ and all $\rho \geq 0$, observe that with fixed learning rates $\eta_t = \eta \in (0, \frac{1}{2(\rho + \lambda_{\max}(k(\boldsymbol{X},\boldsymbol{X}))/n)}) \subseteq (0, \frac{1}{2\lambda_{\max}(k(\boldsymbol{X},\boldsymbol{X}))/n})$, for all $t \in \mathbb{N} \cup \{\infty\}$ it holds that

$$\sum_{i=0}^{t} \frac{2\eta_i}{n} \prod_{j=0}^{t-i-1} (1 - 2\eta_{t-j}\lambda_k/n) = \frac{2\eta}{n} \sum_{i=0}^{t} (1 - 2\eta\lambda_k/n)^{t-i}$$

$$= \frac{2\eta}{n}\sum_{i=0}^{t}(1-2\eta\lambda_k/n)^i = \frac{2\eta}{n}\frac{1-(1-2\eta\lambda_k/n)^{t+1}}{2\eta\lambda_k/n} \leq \frac{1}{\lambda_k}.$$

Since it suffices to consider the case $\rho \to 0$, using the above derivation in (C.1) yields $\|f_{t,\rho}^{\mathrm{GD}}\|_{\mathcal{H}}^2 \leq \|f_{\infty,0}\|_{\mathcal{H}}^2$ for all $t \in \mathbb{N}$, which concludes the proof. □

D Proof of Theorem 1

Our goal in this section is to prove Theorem D.1, which can be seen as a generalization of Theorem 1 to varying bandwidths. To be able to speak of bandwidths, we need to consider translation-invariant kernels. Although Theorem 1 is formulated for general kernels with Sobolev RKHS, it follows from Theorem D.1 since we can always find, for a fixed bandwidth, a translation-invariant kernel with equivalent RKHS, such that only the constant C_{norm} changes in the theorem statement.

To generate the RKHS H^s, Buchholz (2022) uses the translation-invariant kernel $k^B(\boldsymbol{x},\boldsymbol{y}) = u^B(\boldsymbol{x}-\boldsymbol{y})$ defined via its Fourier transform $\hat{u}^B(\boldsymbol{\xi}) = (1+|\boldsymbol{\xi}|^2)^{-s}$. Adapting the bandwidth, the kernel is then normalized in the usual L_1-sense,

$$k_\gamma^B(\boldsymbol{x},\boldsymbol{y}) = \gamma^{-d}u^B((\boldsymbol{x}-\boldsymbol{y})/\gamma). \tag{D.1}$$

Theorem D.1 (Inconsistency of overfitting estimators). *Let assumptions (D1) and (K) hold. Let $c_{\mathrm{fit}} \in (0,1]$ and $C_{\mathrm{norm}} > 0$. Then, there exist $c > 0$ and $n_0 \in \mathbb{N}$ such that the following holds for all $n \geq n_0$ with probability $1 - O(1/n)$ over the draw of the data set D with n samples: For every function $f \in \mathcal{H}_k$ with*

(O) $\frac{1}{n}\sum_{i=1}^n (f(x_i)-y_i)^2 \leq (1-c_{\mathrm{fit}})\cdot\sigma^2$ *(training error below Bayes risk) and*
(N) $\|f\|_{\mathcal{H}_k} \leq C_{\mathrm{norm}}\|f_{\infty,0}\|_{\mathcal{H}_k}$ *(norm comparable to minimum-norm interpolant (1)),*

the excess risk satisfies

$$R_P(f) - R_P(f^*) \geq c > 0\,. \tag{D.2}$$

If k_γ denotes a L_1-normalized translation-invariant kernel with bandwidth $\gamma > 0$, i.e. there exists a $q : \mathbb{R}^d \to \mathbb{R}$ such that $k_\gamma(x,y) = \gamma^{-d}q(\frac{x-y}{\gamma})$, then inequality (D.2) holds with c independent of the sequence of bandwidths $(\gamma_n)_{n\in\mathbb{N}} \subseteq (0,1)$, as long as f_D fulfills (N) for the sequence $(\mathcal{H}_{\gamma_n})_{n\in\mathbb{N}}$ with constant $C_{\mathrm{norm}} > 0$.

Proof. By assumption, the RKHS norm $\|\cdot\|_{\mathcal{H}_k}$ induced by the kernel k (or k_γ if we allow bandwidth adaptation) is equivalent to the RKHS norm $\|\cdot\|_{\mathcal{H}_\gamma}$ induced by a kernel of the form (D.1) with an arbitrary but fixed choice of bandwidth $\gamma \in (0,1)$, which means that there exists a constant $C_\gamma > 0$ such that $\frac{1}{C_\gamma}\|f\|_{\mathcal{H}_\gamma} \leq \|f\|_{\mathcal{H}_k} \leq C_\gamma\|f\|_{\mathcal{H}_\gamma}$ for all $f \in \mathcal{H}_k$. Hence the minimum-norm interpolant $g_{D,\gamma}$ in $\mathcal{H}_\gamma$ satisfies

$$\|f_D\|_{\mathcal{H}_\gamma} \leq C_\gamma\|f_D\|_{\mathcal{H}_k} \leq C_\gamma C_{\mathrm{norm}}\|g_D\|_{\mathcal{H}_k} \leq C_\gamma C_{\mathrm{norm}}\|g_{D,\gamma}\|_{\mathcal{H}_k} \leq C_\gamma^2 C_{\mathrm{norm}}\|g_{D,\gamma}\|_{\mathcal{H}_\gamma},$$

where $\|g_D\|_{\mathcal{H}_k} \leq \|g_{D,\gamma}\|_{\mathcal{H}_k}$ because, g_D is the minimum-norm interpolant in $\mathcal{H}_k$.

Now consider the RKHS norm $\|\cdot\|_{\tilde{\mathcal{H}}_\gamma}$ of a translation-invariant kernel k_γ. Then the functions $\{h_p(x) = e^{ip\cdot x}\}_{p\in\mathbb{R}^d}$ are eigenfunctions of the kernel's integral operator, so that the RKHS norm can be written as (Rakhlin and Zhai, 2019)

$$\|f\|_{\tilde{\mathcal{H}}_\gamma}^2 = \frac{1}{(2\pi)^d}\int_{\mathbb{R}^d}\frac{|\hat{f}(\omega)|^2}{\hat{q}(\omega)}d\omega,$$

where $\hat{f}$ denotes the Fourier transform of f.

By assumption we know that there exists a $C_{\gamma_0} > 0$ such that $\frac{1}{C_{\gamma_0}}\|f\|_{\mathcal{H}_{\gamma_0}} \leq \|f\|_{\tilde{\mathcal{H}}_{\gamma_0}} \leq C_{\gamma_0}\|f\|_{\mathcal{H}_{\gamma_0}}$ holds for some fixed bandwidth $\gamma_0 > 0$, then substituting by $\tilde{\omega} = \frac{\gamma}{\gamma_0}\omega$ yields

$$\|f\|_{\tilde{\mathcal{H}}_\gamma} = \frac{1}{(2\pi)^d}\int_{R^d}\frac{|\hat{f}(\omega)|^2}{\hat{q}_1(\gamma\omega)}d\omega = \frac{1}{(2\pi)^d}\int_{R^d}\frac{|\hat{f}(\frac{\gamma_0}{\gamma}\tilde{\omega})|^2}{\hat{q}_1(\gamma_0\tilde{\omega})}\left(\frac{\gamma_0}{\gamma}\right)^d d\tilde{\omega} = \|\tilde{f}\|_{\tilde{\mathcal{H}}_{\gamma_0}}$$

$$\leq C_{\gamma_0}\|\tilde{f}\|_{\mathcal{H}_{\gamma_0}} = \frac{C_{\gamma_0}}{(2\pi)^d}\int_{R^d}\frac{|\hat{f}(\frac{\gamma_0}{\gamma}\tilde{\omega})|^2}{\hat{q}_2(\gamma_0\tilde{\omega})}\left(\frac{\gamma_0}{\gamma}\right)^d d\tilde{\omega} = \frac{C_{\gamma_0}}{(2\pi)^d}\int_{R^d}\frac{|\hat{f}(\omega)|^2}{\hat{q}_2(\gamma\omega)}d\omega = C_{\gamma_0}\|f\|_{\mathcal{H}_\gamma}.$$

In the same way we get $\frac{1}{C_{\gamma_0}}\|f\|_{\mathcal{H}_\gamma} \leq \|f\|_{\tilde{\mathcal{H}}_\gamma}$ for arbitrary $\gamma \in (0,1)$. This shows that the constant C_{γ_0}, that quantifies the equivalence between $\|\cdot\|_{\mathcal{H}_\gamma}$ and $\|\cdot\|_{\tilde{\mathcal{H}}_\gamma}$ does not depend on the bandwidth γ. Finally Proposition D.4, Proposition D.2 and Remark D.3 together yield the result. □

The following proposition generalizes the inconsistency result for large bandwidths, Proposition 4 in Buchholz (2022), beyond interpolating estimators to estimators that overfit at least an arbitrary constant fraction beyond the Bayes risk and whose RKHS norm is at most a constant factor larger than the RKHS norm of the minimum-norm interpolant. Compared to Rakhlin and Zhai (2019), Buchholz gets a statement in probability over the draw of a training set D and less restrictive assumptions on the domain Ω and dimension d.

Proposition D.2 (Inconsistency for large bandwidths). *Let $c_{\text{fit}} \in (0,1]$ and $C_{\text{norm}} > 0$. Let the data set $D = \{(\boldsymbol{x}_1, y_1), \ldots, (\boldsymbol{x}_n, y_n)\}$ be drawn i.i.d. from a distribution P that fulfills Assumption (D1), let $g_{D,\gamma}$ be the minimum-norm interpolant in $\mathcal{H} := \mathcal{H}_\gamma$ with respect to the kernel (D.1) for a bandwidth $\gamma > 0$. Then, for every $A > 0$, there exist $c > 0$ and $n_0 \in \mathbb{N}$ such that the following holds for all $n \geq n_0$ with probability $1 - O(1/n)$ over the draw of the data set D with n samples:*

For every function $f \in \mathcal{H}$ that fulfills Assumption (O) with c_{fit} and Assumption (N) with C_{norm} the excess risk satisfies

$$\mathbb{E}_{\boldsymbol{x}}(f(\boldsymbol{x}) - f^*(\boldsymbol{x}))^2 \geq c > 0,$$

where c depends neither on n nor on $1 > \gamma > An^{-1/d} > 0$.

Remark D.3. Proposition D.2 holds for any kernel that fulfills Assumption (K). The reason is that any kernel k that fulfills assumption (K) and the kernel defined in (D.1) have the same RKHS and equivalent norms. Therefore every function $f \in \mathcal{H}_k = \mathcal{H}_\gamma$ (equality as sets) that fulfills Assumptions (O) and (N) for the kernel k also fulfills Assumptions (O) and (N) with an adapted constant C_{norm} for the kernel (D.1). ◀

Proof. **Step 1: Generalizing the procedure in Buchholz (2022).**

We write $[n] = \{1, \ldots, n\}$ and follow the proof of Proposition 4 in Buchholz (2022). Define $u(\boldsymbol{x}) = f(\boldsymbol{x}) - f^*(\boldsymbol{x})$. We need to show that with probability at least $1 - O(n^{-1})$ over the draw of D it holds that $\|u\|_{L^2(P_X)} \geq c > 0$, where c depends neither on n nor on γ.

For this purpose we show that with probability at least $1 - 3n^{-1}$ over the draw of D there exist a constants $c'', \kappa'' > 0$ depending only on c_{fit} and a subset $\mathcal{P}'' \subseteq [n]$ with $|\mathcal{P}''| \geq \lfloor \kappa'' \cdot n \rfloor$ such that

$$|f(\boldsymbol{x}_i) - f^*(\boldsymbol{x}_i)| \geq c'' > 0 \text{ holds for all } i \in \mathcal{P}''. \tag{D.3}$$

Then via Lemma 7 in Buchholz (2022) as well as Lemma D.7 we can choose a large subset $\mathcal{P}''' \subseteq [n]$ of the training point indices with $|\mathcal{P}'''| \geq n - |\mathcal{P}''|/2$, such that the $\boldsymbol{x}_i$ for $i \in \mathcal{P}'''$ are well-separated in the sense that $\min_{\{i,j \in \mathcal{P}''', i \neq j\}} \|\boldsymbol{x}_i - \boldsymbol{x}_j\| \geq d_{min}$ with $d_{min} := c''' n^{-1/d}$, where c''' depends on c_{fit}, d, the upper bound on the Lebesgue density C_u and on the smoothness of the RKHS s. Then the intersection $\mathcal{P}'' \cap \mathcal{P}'''$ contains at least $\frac{|\mathcal{P}''|}{2}$ points. Now we can replace $\mathcal{P}'$ in the proof of Proposition 4 for $s \in \mathbb{N}$ in Buchholz (2022) by the intersection $\mathcal{P}'' \cap \mathcal{P}'''$. The rest of the proof applies without modification, where (42) holds by our assumption $\|f\|_{\mathcal{H}} \leq C_{\mathcal{H}}\|g_D\|_{\mathcal{H}}$. Our modifications do not affect Buchholz' arguments for the extension to $s \notin \mathbb{N}$.

Step 2: The existence of $\mathcal{P}''$.

Given a choice of $\kappa'', c'' > 0$, consider the event (over the draw of D)

$$\begin{aligned} E &:= \{\nexists\, \mathcal{P}'' \subseteq [n] \text{ with } |\mathcal{P}''| \geq \lfloor \kappa'' \cdot n \rfloor \text{ that fulfills (D.3)}\} \\ &= \{\exists\, \tilde{\mathcal{P}} \subseteq [n] \text{ with } |\tilde{\mathcal{P}}| \geq \lceil (1-\kappa'')n \rceil \text{ such that } |f^*(\boldsymbol{x}_i) - f(\boldsymbol{x}_i)| < c'' \quad \forall i \in \tilde{\mathcal{P}}\}. \end{aligned}$$

With the proper choices of c'' and κ'' independent of n and f, we will show $P(E) \leq 3n^{-1}$. We will find a small $c'' > 0$ such that if f^* and f are closer than c'' on too many training points $\tilde{\mathcal{P}}$

and f overfits by at least the fraction c_{fit}, the noise variables ε_i on the complement $\tilde{\mathcal{P}}^c$ would have to be unreasonably large, contradicting the event E_{6i} defined below, and implying (D.3) with high probability. We will use the notation $\|\boldsymbol{f}\|_{\mathcal{P}}^2 \coloneqq \sum_{i\in\mathcal{P}} f(\boldsymbol{x}_i)^2$ and $\|\boldsymbol{y}\|_{\mathcal{P}}^2 \coloneqq \sum_{i\in\mathcal{P}} y_i^2$.

Step 2b: Noise bounds.

Lemma D.6 (i) states that there exists a $\kappa'' > 0$ small enough such that the event (over the draw of D)

$$E_{6i} \coloneqq \{\forall \mathcal{P}_1 \subseteq [n] \text{ with } |\mathcal{P}_1| \leq \lfloor \kappa'' \cdot n \rfloor \text{ it holds that } \frac{1}{n}\|\boldsymbol{f}^* - \boldsymbol{y}\|_{\mathcal{P}_1}^2 = \frac{1}{n}\sum_{i\in\mathcal{P}_1} \varepsilon_i^2 < \frac{c_{\text{fit}}}{4}\sigma^2\},$$

fulfills, for n large enough, $P(E_{6i}) \geq 1 - n^{-1}$.

Lemma D.6 (ii) implies that there exists a $c_{\text{lower}} > 0$ such that the event (over the draw of D)

$$E_{6ii} \coloneqq \{\forall \mathcal{P}_2 \text{ with } |\mathcal{P}_2| \geq \lfloor (1-\kappa'')n \rfloor \text{ it holds that } \frac{1}{n}\|\boldsymbol{f}^* - \boldsymbol{y}\|_{\mathcal{P}_2}^2 \geq c_{\text{lower}} \cdot \sigma^2\},$$

fulfills, for n large enough, $P(E_{6ii}) \geq 1 - n^{-1}$.

Lemma D.5 states that the total amount of noise $\|\boldsymbol{\varepsilon}\|_{[n]}^2$ concentrates around its mean $n\sigma^2$. More precisely, we will use that for any $c_\varepsilon \in (0,1)$ the event (over the draw of D)

$$E_5 \coloneqq \left\{\frac{1}{n}\|\boldsymbol{f}^* - \boldsymbol{y}\|_{[n]}^2 \geq c_\varepsilon \cdot \sigma^2\right\},$$

fulfills $P(E_5) \geq 1 - \exp\left(-n \cdot \left(\frac{1-c_\varepsilon}{2}\right)^2\right)$.

Step 2c: Lower bounding $\|\boldsymbol{\varepsilon}\|_{\tilde{\mathcal{P}}^c}^2$.

Given some function $f \in \mathcal{H}$, assume in steps 2c and 2d that event E holds and that $\tilde{\mathcal{P}} \subseteq [n]$ denotes a subset of the training set that fulfills $|\tilde{\mathcal{P}}| \geq \lceil (1-\kappa'')n \rceil$ and $|f^*(\boldsymbol{x}_i) - f(\boldsymbol{x}_i)| < c'' \quad \forall i \in \tilde{\mathcal{P}}$.

In step 2c, assume we choose $\tilde{c}_{\text{fit}} > 0$ such that $\tilde{c}_{\text{fit}}\|\boldsymbol{f}^* - \boldsymbol{y}\|_{\tilde{\mathcal{P}}}^2 \leq \|\boldsymbol{f} - \boldsymbol{y}\|_{\tilde{\mathcal{P}}}^2$. Then by the overfitting Assumption (O) it holds that

$$\frac{1}{n}\left(\tilde{c}_{\text{fit}}\|\boldsymbol{f}^* - \boldsymbol{y}\|_{\tilde{\mathcal{P}}}^2 + \|\boldsymbol{f} - \boldsymbol{y}\|_{\tilde{\mathcal{P}}^c}^2\right) \leq \frac{1}{n}\left(\|\boldsymbol{f} - \boldsymbol{y}\|_{\tilde{\mathcal{P}}}^2 + \|\boldsymbol{f} - \boldsymbol{y}\|_{\tilde{\mathcal{P}}^c}^2\right) \leq (1 - c_{\text{fit}})\sigma^2. \tag{D.4}$$

If we restrict ourselves to event E_5, dropping the term $\|\boldsymbol{f} - \boldsymbol{y}\|_{\tilde{\mathcal{P}}^c}^2$ in (D.4), then dividing by $\tilde{c}_{\text{fit}}$ and subtracting the result from the inequality in the definition of event E_5 yields

$$\frac{1}{n}\|\boldsymbol{\varepsilon}\|_{\tilde{\mathcal{P}}^c}^2 = \frac{1}{n}\|\boldsymbol{f}^* - \boldsymbol{y}\|_{\tilde{\mathcal{P}}^c}^2 \geq c_\varepsilon\sigma^2 - \frac{1 - c_{\text{fit}}}{\tilde{c}_{\text{fit}}}\sigma^2. \tag{D.5}$$

Step 2d: Choosing the constants.

If we choose $c_\varepsilon \coloneqq 1 - \frac{c_{\text{fit}}}{4}$ and $\tilde{c}_{\text{fit}} \coloneqq \frac{2-2c_{\text{fit}}}{2-c_{\text{fit}}} \in (0,1)$, then (D.5) becomes

$$\frac{1}{n}\|\boldsymbol{\varepsilon}\|_{\tilde{\mathcal{P}}^c}^2 \geq \frac{c_{\text{fit}}}{4}\sigma^2.$$

Now it is left to show that the condition $\tilde{c}_{\text{fit}}\|\boldsymbol{f}^* - \boldsymbol{y}\|_{\tilde{\mathcal{P}}}^2 \leq \|\boldsymbol{f} - \boldsymbol{y}\|_{\tilde{\mathcal{P}}}^2$, that is required for Step 2c, holds with high probability with our choice of $\tilde{c}_{\text{fit}}$.

With some arbitrary but fixed $\varepsilon_{\text{lower}} \in (0, \sqrt{c_{\text{lower}}})$, choose $c'' \coloneqq (1 - \sqrt{\tilde{c}_{\text{fit}}})(\sqrt{\frac{c_{\text{lower}}}{1-\kappa''}} - \frac{\varepsilon_{\text{lower}}}{\sqrt{1-\kappa''}})\sigma$. Then on event E_{6ii}, for n large enough, it holds that

$$(1 - \sqrt{\tilde{c}_{\text{fit}}})\frac{1}{\sqrt{n}}\|\boldsymbol{f}^* - \boldsymbol{y}\|_{\tilde{\mathcal{P}}} \geq (1 - \sqrt{\tilde{c}_{\text{fit}}})\sqrt{c_{\text{lower}}}\sigma \geq \sqrt{1-\kappa''} \cdot c'' + \frac{c''}{\sqrt{n}}. \tag{D.6}$$

By definition of $\tilde{\mathcal{P}}$, it holds that

$$\|\boldsymbol{f} - \boldsymbol{f}^*\|_{\tilde{\mathcal{P}}}^2 = \sum_{i\in\tilde{\mathcal{P}}}(f(\boldsymbol{x}_i) - f^*(\boldsymbol{x}_i))^2 < \lceil (1-\kappa'')n \rceil (c'')^2,$$

so that

$$\frac{1}{\sqrt{n}}\|\boldsymbol{f}-\boldsymbol{f}^*\|_{\tilde{\mathcal{P}}} < \sqrt{1-\kappa''}\cdot c'' + \frac{c''}{\sqrt{n}}. \tag{D.7}$$

Now, using the triangle inequality, (D.7) and (D.6) yields the condition required for Step 2c,

$$\begin{aligned}
&\frac{1}{\sqrt{n}}\|\boldsymbol{f}-\boldsymbol{y}\|_{\tilde{\mathcal{P}}}\\
\geq&\frac{1}{\sqrt{n}}\|\boldsymbol{f}^*-\boldsymbol{y}\|_{\tilde{\mathcal{P}}} - \frac{1}{\sqrt{n}}\|\boldsymbol{f}-\boldsymbol{f}^*\|_{\tilde{\mathcal{P}}}\\
\geq&\frac{1}{\sqrt{n}}\|\boldsymbol{f}^*-\boldsymbol{y}\|_{\tilde{\mathcal{P}}} - \sqrt{1-\kappa''}\cdot c'' - \frac{c''}{\sqrt{n}}\\
\geq&\sqrt{\tilde{c}_{\text{fit}}}\frac{1}{\sqrt{n}}\|\boldsymbol{f}^*-\boldsymbol{y}\|_{\tilde{\mathcal{P}}}.
\end{aligned}$$

Step 2e: Upper bounding the probability of E.

To conclude, we have seen in steps 2c and 2d that on $E\cap E_{6ii}\cap E_5$, it holds that

$$\frac{1}{n}\|\boldsymbol{\varepsilon}\|^2_{\tilde{\mathcal{P}}^c} \geq \frac{c_{\text{fit}}}{4}\sigma^2.$$

On E_{6i}, it holds that

$$\frac{1}{n}\|\boldsymbol{\varepsilon}\|^2_{\tilde{\mathcal{P}}^c} < \frac{c_{\text{fit}}}{4}\sigma^2.$$

Hence $E_{6i}\cap E\cap E_{6ii}\cap E_5=\emptyset$. This implies $E\subseteq(E_5\cap E_{6i}\cap E_{6ii})^c$, where the right hand side is independent of $f\in\mathcal{H}$ and just depends on the training data D. Since $P(E_{6i})\geq 1-n^{-1}$ and $P(E_{6ii}\cap E_5)\geq 1-n^{-1}-\exp\left(-n\cdot\left(\frac{1-c_\varepsilon}{2}\right)^2\right)$, it must hold that, for n large enough,

$$P(E)\leq P((E_5\cap E_{6i}\cap E_{6ii})^c)\leq 2n^{-1}+\exp\left(-n\cdot\left(\frac{1-c_\varepsilon}{2}\right)^2\right)\leq 3n^{-1}. \qquad \square$$

The following proposition generalizes the inconsistency result for small bandwidths, Proposition 5 in Buchholz (2022), beyond interpolating estimators to estimators whose RKHS norm is at most a constant factor larger than the RKHS norm of the minimum-norm interpolant. The intuition is that if the bandwidth is too small, then the minimum-norm interpolant $g_{D,\gamma}$ returns to 0 between the training points. Then $\|g_{D,\gamma}\|_{L_2(\rho)}$ is smaller and bounded away from $\|f^*\|_{L_2(\rho)}$. We can replace $g_{D,\gamma}$ by any other function $f\in\mathcal{H}$ that fulfills Assumption (N).

Proposition D.4 (Inconsistency for small bandwidths). *Under the assumptions of Proposition D.2, there exist constants $B,c>0$ such that, with probability $1-O(n^{-1})$ over the draw of D: For any function $f\in\mathcal{H}$ that fulfills Assumption (N) but not necessarily Assumption (O), the excess risk satisfies*

$$\mathbb{E}_{\boldsymbol{x}}(f(\boldsymbol{x})-f^*(\boldsymbol{x}))^2\geq c>0,$$

where c depends neither on n nor on $\gamma<Bn^{-1/d}$.

Proof. Denote the upper bound on the Lebesgue density of P_X by C_u. The triangle inequality implies

$$\begin{aligned}
\|f^*-f\|_{L_2(P_X)} &\geq \|f^*\|_{L_2(P_X)} - \|f\|_{L_2(P_X)} \geq \|f^*\|_{L_2(P_X)} - \sqrt{C_u}\|f\|_2\\
&\geq \|f^*\|_{L_2(P_X)} - \sqrt{C_u}\|f\|_{\mathcal{H}} \geq \|f^*\|_{L_2(P_X)} - C_{\mathcal{H}}\sqrt{C_u}\|g_{D,\gamma}\|_{\mathcal{H}},
\end{aligned}$$

where $\|f\|_2\leq\|f\|_{\mathcal{H}}$ follows from the fact that the Fourier transform $\hat{k}$ of the kernel satisfies $\hat{k}(\xi)\leq 1$. Now in the proof of Lemma 17 in Buchholz (2022) $a>0$ can be chosen smaller to generalize the statement to

$$\|g_{D,\gamma}\|^2_{\mathcal{H}}\leq\frac{1}{6C^2_{\mathcal{H}}C_u}\|f^*\|^2_{L_2(P_X)}+c_9(\gamma^2n^{2/d}+\gamma^{2s}n^{2s/d}),$$

where c_9 depends on c_u, f^*, d, s and C_{norm}. Finally we can choose B small enough such that Eq. (32) in Buchholz (2022) can be replaced by $C_{\mathcal{H}}\sqrt{C_u}\|g_{D,\gamma}\|_{\mathcal{H}}\leq\frac{2}{3}\|f^*\|_{L_2(P_X)}$ so that we get

$$\|f^*-f\|_{L_2(P_X)}\geq\frac{1}{3}\|f^*\|_{L_2(P_X)}>0. \qquad \square$$

D.1 Auxiliary results for the proof of Theorem 1

Lemma D.5 (Concentration of χ^2_n variables)**.** *Let U be a chi-squared distributed random variable with n degrees of freedom. Then, for any $c \in (0,1)$ it holds that*

$$P\left(\frac{U}{n} \leq c\right) \leq \exp\left(-n \cdot \left(\frac{1-c}{2}\right)^2\right).$$

Proof. Lemma 1 in Laurent and Massart (2000) implies for any $x > 0$,

$$P\left(\frac{U}{n} \leq 1 - 2\sqrt{\frac{x}{n}}\right) \leq \exp\left(-x\right).$$

Solving $c = 1 - 2\sqrt{\frac{x}{n}}$ for x yields $x = n \cdot \left(\frac{1-c}{2}\right)^2$. □

Lemma D.6. *Let $\varepsilon_1, \ldots, \varepsilon_n$ be i.i.d. $\mathcal{N}(0, \sigma^2)$ random variables, $\sigma^2 > 0$. Let $(\varepsilon^2)^{(i)}$ denote the i-th largest of $\varepsilon_1^2, \ldots, \varepsilon_n^2$.*

(i) ***A constant fraction of noise cannot concentrate on less than $\Theta(n)$ points:*** *For all constants $\alpha, c > 0$ there exists a constant $C \in (0,1)$ such that with probability at least $1 - n^{-\alpha}$, for n large enough,*

$$\frac{1}{n} \sum_{i=1}^{\lfloor Cn \rfloor} (\varepsilon^2)^{(i)} < c\sigma^2 .$$

(ii) $\Theta(n)$ ***points amount to a constant fraction of noise:*** *For all constants $\alpha > 0$ and $\kappa \in (0,1)$ there exists a constant $c > 0$ such that with probability at least $1 - n^{-\alpha}$, for n large enough,*

$$\frac{1}{n} \sum_{i=1}^{\lfloor (1-\kappa)n \rfloor} (\varepsilon^2)^{(n-i+1)} \geq c\sigma^2 .$$

Proof. Without loss of generality, we can assume $\sigma^2 = 1$.

(i) For a constant $C \in (0,1)$ yet to be chosen, consider the sum

$$S_{C,n} \coloneqq \frac{1}{n} \sum_{i=1}^{\lfloor Cn \rfloor} (\varepsilon^2)^{(i)} .$$

For $T > 0$ yet to be chosen, we consider the random set $\mathcal{I}_T \coloneqq \{i \in [n] \mid \varepsilon_i^2 > T\}$ and denote its size by $K \coloneqq |\mathcal{I}_T|$. To bound K, we note that $K = \xi_1 + \ldots + \xi_n$, where $\xi_i = \mathbb{1}_{\varepsilon_i^2 > T}$. We first want to bound $p_T \coloneqq \mathbb{E}\xi_i = P(\varepsilon_i^2 > T)$.
The random variables ε_i^2 follow a χ^2_1-distribution, whose CDF we denote by $F(t)$ and whose PDF is

$$f(t) = \mathbb{1}_{(0,\infty)}(t) C_1 t^{-1/2} \exp(-t/2) \tag{D.8}$$

for some absolute constant C_1. Moreover, we use Φ and ϕ to denote the CDF and PDF of $\mathcal{N}(0,1)$, respectively.
Step 1: Tail bounds. Following Duembgen (2010), we have for $x > 0$:

$$1 - \Phi(x) > \frac{2\phi(x)}{\sqrt{4 + x^2} + x} \geq \frac{2\phi(x)}{2 + x + x} = \frac{\phi(x)}{1 + x}$$
$$1 - \Phi(x) < \frac{2\phi(x)}{\sqrt{2 + x^2} + x} \leq \frac{2\phi(x)}{1 + x} .$$

Hence, for $t > 0$, we have

$$1 - F(t) = 2(1 - \Phi(\sqrt{t})) > \frac{2\phi(\sqrt{t})}{1 + \sqrt{t}} = \sqrt{\frac{2}{\pi}} \frac{\exp(-t/2)}{1 + \sqrt{t}}$$
$$1 - F(t) = 2(1 - \Phi(\sqrt{t})) < \frac{4\phi(\sqrt{t})}{1 + \sqrt{t}} = \sqrt{\frac{8}{\pi}} \frac{\exp(-t/2)}{1 + \sqrt{t}} .$$

By choosing $T := -2\log(C\sqrt{\pi/32}) > 0$, we obtain

$$p_T = 1 - F(T) < \sqrt{\frac{8}{\pi}}\exp(-T/2) = C/2\,.$$

Step 2: Bounding K. The random variables ξ_i from above satisfy $\xi_i \in [0,1]$. By Hoeffding's inequality (Steinwart and Christmann, 2008, Theorem 6.10), we have for $\tau > 0$

$$P\left(\frac{1}{n}\sum_{i=1}^{n}(\xi_i - \mathbb{E}\xi_i) \geq (1-0)\sqrt{\frac{\tau}{2n}}\right) \leq \exp(-\tau)\,.$$

We choose $\tau := C^2 n/2$, such that with probability $\geq 1 - \exp(-C^2 n/2)$, we have

$$K/n - p_T = \frac{1}{n}\sum_{i=1}^{n}(\xi_i - \mathbb{E}\xi_i) \leq \sqrt{\frac{C^2 n/2}{2n}} = C/2\,.$$

Suppose that this holds. Then, $K \leq np_T + Cn/2 < Cn$ and, since K is an integer, $K \leq \lfloor Cn \rfloor$. This implies

$$S_{C,n} \leq \frac{1}{n}\left(\sum_{i=1}^{K}(\varepsilon^2)^{(i)} + (\lfloor Cn \rfloor - K)T\right) \leq CT + \frac{1}{n}\sum_{i=1}^{K}(\varepsilon^2)^{(i)}\,. \tag{D.9}$$

We now want to bound $\sum_{i=1}^{K}(\varepsilon^2)^{(i)}$. To this end, we note that conditioned on $K = k$ for some $k \in [n]$, the k random variables $(\varepsilon_i)_{i\in\mathcal{I}_T}$ are i.i.d. drawn from the distribution of ε^2 given $\varepsilon^2 > T$, for $\varepsilon \sim \mathcal{N}(0,1)$. By $X, X_1, X_2, \ldots$, we denote i.i.d. random variables drawn from the distribution of $\varepsilon^2 - T \mid \varepsilon^2 > T$. This means that conditioned on $K = k$,

$$\sum_{i=1}^{k}(\varepsilon^2)^{(i)} = \sum_{i\in\mathcal{I}_T}\varepsilon_i^2 \quad \text{is distributed as} \quad kT + \sum_{i=1}^{k}X_i\,. \tag{D.10}$$

Step 3: Conditional expectation. The density of X is given by

$$\begin{aligned} p_X(t) &= \mathbb{1}_{t>0}\frac{f(T+t)}{1-F(T)} \overset{(D.8)}{\leq} \mathbb{1}_{t>0}\frac{C_1(T+t)^{-1/2}\exp(-(t+T)/2)}{\sqrt{2/\pi}\exp(-T/2)/(1+\sqrt{T})} \\ &\leq \mathbb{1}_{t>0}C_2\exp(-t/2)\,, \end{aligned}$$

where we have used that for $t > 0$,

$$\frac{1+\sqrt{T}}{\sqrt{T+t}} \leq \frac{1+\sqrt{T}}{\sqrt{T}} = 1 + \frac{1}{\sqrt{T}} \leq 2$$

since $T = -2\log(C\sqrt{\pi/32}) \geq -2\log(\sqrt{\pi/32}) \approx 1.008$. We can now bound

$$\begin{aligned} \mathbb{E}[X] &= \int_0^\infty t p_X(t)\,\mathrm{d}t \\ &\leq \int_0^\infty C_2 t\exp(-t/2)\,\mathrm{d}t = 4C_2\,. \end{aligned} \tag{D.11}$$

Step 4: Conditional subgaussian norm. For $t \geq 0$,

$$\begin{aligned} P(|X| > t) = P(X > t) &= \frac{1-F(T+t)}{1-F(T)} \leq 2\frac{1+\sqrt{T}}{1+\sqrt{T+t}}\frac{\exp(-(T+t)/2)}{\exp(-T/2)} \\ &\leq 2\exp(-t/2)\,. \end{aligned}$$

Since the denominator 2 in $2\exp(-t/2)$ is constant, by Proposition 2.7.1 and Definition 2.7.5 in Vershynin (2018), the subexponential norm $\|X\|_{\psi_1}$ is therefore bounded by an absolute constant C_3. Moreover, by Excercise 2.7.10 in Vershynin (2018), we have $\|X - \mathbb{E}X\|_{\psi_1} \leq C_4\|X\|_{\psi_1} \leq C_5$ for absolute constants C_4, C_5.

Step 5: Conditional Concentration. Now, Bernstein's inequality for subexponential random variables (Vershynin, 2018, Corollary 2.8.1) yields for $t \geq 0$ and some absolute constant $C_6 > 0$:

$$P\left(\left|\sum_{i=1}^{k}X_i - \mathbb{E}X_i\right| \geq t\right) \leq 2\exp\left(-C_6\min\left(\frac{t^2}{kC_5^2}, \frac{t}{C_5}\right)\right)\,. \tag{D.12}$$

We choose $t = C_5 Cn$ and obtain for $k \leq Cn$

$$\begin{aligned} & P\left(\sum_{i=1}^{k} (\varepsilon^2)^{(i)} \geq kT + 4C_2 k + C_5 Cn \;\middle|\; K = k\right) \\ \overset{(D.10)}{=}\; & P\left(\sum_{i=1}^{k} X_i \geq 4C_2 k + C_5 Cn \;\middle|\; K = k\right) \\ \overset{(D.11)}{\leq}\; & P\left(\left|\sum_{i=1}^{k} X_i - \mathbb{E}X_i\right| \geq t\right) \\ \overset{(D.12)}{\leq}\; & 2\exp\left(-C_6 Cn\right) . \end{aligned}$$

Step 6: Final bound. From Step 2, we know that $K \leq \lfloor Cn \rfloor$ with probability $\geq 1 - \exp(-C^2 n/2)$. Moreover, in this case, Step 5 yields

$$\sum_{i=1}^{K} (\varepsilon^2)^{(i)} < KT + 4C_2 K + C_5 Cn \leq Cn(T + 4C_2 + C_5)$$

with probability $\geq 1 - \exp(-C_6 Cn)$. By Eq. (D.9), we therefore have

$$S_{C,n} < CT + C(T + 4C_2 + C_5) = -4C\log(C\sqrt{\pi/32}) + C_7 C .$$

Since $\lim_{C \searrow 0} -C\log(C) = 0$, we can choose $C \in (0,1)$ such that $-4C\log(C\sqrt{\pi/32}) + C_7 C < c$ for the given constant $c > 0$ from the theorem statement, and obtain the desired bound with high probability in n.

(ii) Since the ε_i^2 are non-negative and their distribution has a density, there must exist $T > 0$ with $P(\varepsilon_i^2 < T) \leq (1-\kappa)/4$. Similar to the proof of (i), we then want to bound $K \coloneqq |\{i \in [n] \mid \varepsilon_i^2 < T\}| = \xi_1 + \ldots + \xi_i$ with $\xi_i = \mathbb{1}_{\varepsilon_i^2 < T}$. The $\xi_i \in [0,1]$ are independent with $\mathbb{E}\xi_i = P(\varepsilon_i^2 < T) \leq (1-\kappa)/4$. As in Step 2 of (i), Hoeffding's inequality then yields for $\tau > 0$:

$$P\left(\frac{1}{n}\sum_{i=1}^{n} (\xi_i - \mathbb{E}\xi_i) \geq (1-0)\sqrt{\frac{\tau}{2n}}\right) \leq \exp(-\tau) .$$

We set $\tau \coloneqq (1-\kappa)^2 n/2$, such that with probability $\geq 1 - \exp((1-\kappa)^2 n/2)$, we have

$$\begin{aligned} K/n - (1-\kappa)/4 \leq K/n - P(\varepsilon_i^2 < T) = \frac{1}{n}\sum_{i=1}^{n} (\xi_i - \mathbb{E}\xi_i) < \sqrt{\frac{(1-\kappa)^2 n/2}{2n}} \\ = \frac{1-\kappa}{2} . \end{aligned}$$

In this case, we have

$$\begin{aligned} \frac{1}{n}\sum_{i=1}^{\lfloor (1-\kappa)n \rfloor} (\varepsilon^2)^{(n-i+1)} \geq \frac{1}{n}(\lfloor (1-\kappa)n \rfloor - K)T \geq \frac{1}{n}(((1-\kappa)n - 1) - K)T \\ \geq \left(\frac{1-\kappa}{4} - \frac{1}{n}\right) T , \end{aligned}$$

where the right-hand side is lower bounded by $c \coloneqq (1-\kappa)T/8$ for n large enough. □

The next lemma is a generalization of Lemma 9 in Buchholz (2022) to arbitrary fractions κ of the training points. Therefore, for any $\kappa \in (0,1)$ define

$$\delta_{\min}(\kappa) = n^{-1/d} \left(\frac{\kappa}{C_\rho \omega_d}\right)^{1/d} ,$$

Lemma D.7 (Generalization of Lemma 9 in Buchholz (2022))**.** *Let $\kappa, \nu \in (0,1)$, and let $c_\Omega > 0$ be a constant that satisfies $P_X(\mathrm{dist}(\boldsymbol{x}, \partial\Omega) < c_\Omega) \leq \kappa$. Let $\mathcal{P} = \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n\}$ be i.i.d. points distributed according to the measure P_X, which has lower and upper bounded density on its entire*

bounded open Lipschitz domain $\Omega \subseteq \mathbb{R}^d$, $C_l \leq p_X(\boldsymbol{x}) \leq C_u$. Then there exists a constant $\Theta > 0$ depending on d, C_u, ν such that with probability at least $1 - \exp\left(-\frac{3\kappa n}{7}\right)$ there exists a good subset $\mathcal{P}' \subseteq \mathcal{P}$, $|\mathcal{P}'| \geq (1 - 7\kappa)n$, with the following properties: For $\boldsymbol{x} \in \mathcal{P}'$ we have $\operatorname{dist}(\boldsymbol{x}, \partial\Omega) \geq c_\Omega$, $|\boldsymbol{x} - \boldsymbol{y}| > \delta_{\min}(\kappa)$ for $\boldsymbol{x} \neq \boldsymbol{y} \in \mathcal{P}'$, and for all $\boldsymbol{x} \in \mathcal{P}'$ we have

$$\sum_{\boldsymbol{y} \in \mathcal{P}' \setminus \{\boldsymbol{x}\}} |\boldsymbol{x} - \boldsymbol{y}|^{-d-2\nu} \leq \frac{2\Theta\delta_{\min}(\kappa)^{-2\nu} n}{\kappa^2}.$$

Proof. First by the definition of $\delta_{\min}$, it holds that

$$P\left(\boldsymbol{x}_j \in \bigcup_{i<j} B\left(\boldsymbol{x}_i, \delta_{\min}\right)\right) \leq C_u \omega_d \delta_{\min}^d n \leq \kappa$$

Also for all $\boldsymbol{y} \in \Omega$

$$\begin{aligned}\mathbb{E}_{\boldsymbol{x}}\left((\boldsymbol{x} - \boldsymbol{y})^{-d-2s} \mathbf{1}\left(|\boldsymbol{x} - \boldsymbol{y}| \geq \delta_{\min}\right)\right) &= \int_{B(\boldsymbol{y}, \delta_{\min})^c} |\boldsymbol{x} - \boldsymbol{y}|^{-d-2\nu} p_X(\boldsymbol{x}) \mathrm{d}\boldsymbol{x} \\ &\leq C_u \int_{B(\boldsymbol{x}, \delta_{\min})^c} |\boldsymbol{x} - \boldsymbol{y}|^{-d-2\nu} \mathrm{d}\boldsymbol{y} \leq \Theta\delta_{\min}^{-2\nu}\end{aligned}$$

for some $\Theta > 0$ depending only on C_u, d and ν. We conclude that for each j

$$P\left(\sum_{i<j} |\boldsymbol{x}_i - \boldsymbol{x}_j|^{-d-2\nu} \mathbf{1}\left(|\boldsymbol{x}_i - \boldsymbol{x}_j| > \delta_{\min}\right) > \frac{\Theta\delta_{\min}^{-2\nu} n}{\kappa}\right) \leq \kappa.$$

Also $P\left(\operatorname{dist}\left(\boldsymbol{x}_j, \partial\Omega\right) < c_\Omega\right) < \kappa$. The union bound implies that

$$\begin{aligned}&P\left(\boldsymbol{x}_j \notin \bigcup_{i<j} B\left(\boldsymbol{x}_i, \delta_{\min}\right), \ \sum_{i<j} |\boldsymbol{x}_i - \boldsymbol{x}_j|^{-d-2\nu} \mathbf{1}_{|\boldsymbol{x}_i - \boldsymbol{x}_j| > \delta_{\min}} < \frac{\Theta\delta_{\min}^{-2\nu} n}{\kappa}, \ \operatorname{dist}\left(\boldsymbol{x}_j, \partial\Omega\right) > c_\Omega\right) \\ =&P\left(\boldsymbol{x}_j \notin \bigcup_{i<j} B\left(\boldsymbol{x}_i, \delta_{\min}\right), \ \sum_{i<j} |\boldsymbol{x}_i - \boldsymbol{x}_j|^{-d-2\nu} < \frac{\Theta\delta_{\min}^{-2\nu} n}{\kappa}, \ \operatorname{dist}\left(\boldsymbol{x}_j, \partial\Omega\right) > c_\Omega\right) \geq 1 - 3\kappa.\end{aligned}$$

We use a martingale construction similar to the one in Lemma 7 of Buchholz (2022) by defining

$$E_j := \left\{\boldsymbol{x}_j \in \bigcup_{i<j} B\left(\boldsymbol{x}_i, \delta_{\min}\right), \quad \text{or} \quad \sum_{i<j} |\boldsymbol{x}_i - \boldsymbol{x}_j|^{-d-2\nu} \geq \frac{\Theta\delta_{\min}^{-2\nu} n}{\kappa}, \quad \text{or} \quad \operatorname{dist}(\boldsymbol{x}_j, \partial\Omega) \leq c_\Omega\right\}.$$

Now define $S_n \coloneqq \sum_{i=1}^n \mathbf{1}_{E_i}$. Using the filtration $\mathcal{F}_i = \sigma(\boldsymbol{x}_1, \ldots, \boldsymbol{x}_i)$, S_n can be decomposed into $S_n = M_n + A_n$, where M_n is a martingale and A_n is predictable with respect to $\mathcal{F}_n$. We then get $A_n \leq \sum_{i=1}^n P(E_i|\mathcal{F}_{i-1}) \leq 3\kappa n$ as well as $\operatorname{Var}(M_i|\mathcal{F}_{i-1}) \leq 3\kappa$. Hence Freedman's inequality Theorem D.8 yields

$$P(S_n \geq 6\kappa n) \leq P(A_n \geq 3\kappa n) + P(M_n \geq 3\kappa n) \leq \exp\left(-\frac{3\kappa n}{7}\right).$$

This implies that with probability at least $1 - \exp\left(-\frac{3\kappa n}{7}\right)$ we can find a subset $\mathcal{P}_s = \{\boldsymbol{z}_1, \ldots, \boldsymbol{z}_m\}$ with $|\mathcal{P}_s| \geq (1 - 6\kappa)n$ on which it holds that $\min_{i \neq j} |\boldsymbol{z}_i - \boldsymbol{z}_j| \geq \delta_{\min}, \operatorname{dist}\left(\boldsymbol{z}_j, \partial\Omega\right) \geq c_\Omega$ and

$$\sum_{i \neq j} |\boldsymbol{z}_i - \boldsymbol{z}_j|^{-d-2\nu} \leq \frac{2\Theta\delta_{\min}^{-2\nu} n^2}{\kappa}.$$

Using Markov's inequality we see that there are at most κn points in $\mathcal{P}_s$ such that

$$\sum_{\boldsymbol{z}' \in \mathcal{P}_s, \boldsymbol{z} \neq \boldsymbol{z}'} |\boldsymbol{z} - \boldsymbol{z}'|^{-d-2\nu} \geq \frac{2\Theta\delta_{\min}^{-2\nu} n}{\kappa^2}.$$

Removing those points we find a subset $\mathcal{P}' \subset \mathcal{P}_s$ such that $|\mathcal{P}'| \geq (1-7\kappa)n$ and for each $\boldsymbol{z} \in \mathcal{P}'$

$$\sum_{\boldsymbol{z}' \in \mathcal{P}_s, \boldsymbol{z} \neq \boldsymbol{z}'} |\boldsymbol{z} - \boldsymbol{z}'|^{-d-2\nu} \leq \frac{2\Theta \delta_{\min}^{-2\nu} n}{\kappa^2}. \qquad \square$$

Theorem D.8 (Freedman's inequality, Theorem 6.1 in Chung and Lu (2006))**.** *Let M_i be a discrete martingale adapted to the filtration $\mathcal{F}_i$ with $M_0 = 0$ that satisfies for all $i \geq 0$*

$$|M_{i+1} - M_i| \leq K$$
$$\operatorname{Var}(M_i \mid \mathcal{F}_{i-1}) \leq \sigma_i^2.$$

Then

$$P(M_n - \mathbb{E}(M_n) \geq \lambda) \leq e^{-\frac{\lambda^2}{2\sum_{i=1}^n \sigma_i^2 + K\lambda/3}}.$$

E Translating between $\mathbb{R}^d$ and $\mathbb{S}^d$

Since the RKHS of the ReLU NTK and NNGP kernels mentioned in Theorem 4 are equivalent to the Sobolev spaces $H^{(d+1)/2}(\mathbb{S}^d)$ and $H^{(d+3)/2}(\mathbb{S}^d)$, respectively (Chen and Xu, 2021, Bietti and Bach, 2021) (detailed summary in Appendix B.4). Inconsistency of functions in these RKHS that fulfill Assumptions (O) and (N), as in Theorem 1, follows immediately by adapting Theorem 1 via Lemma E.1. In particular, inconsistency holds for the gradient flow and gradient descent estimators $f_{t,\rho}$ and $f_{t,\rho}^{\mathrm{GD}}$ as soon as they overfit with lower bounded probability.

For arbitrary open sphere caps $T \coloneqq \{\boldsymbol{x} \in \mathbb{S}^d \mid x_{d+1} < v\}$, $v \in (-1,1)$, and the open unit ball $B_1(0) \coloneqq \{\boldsymbol{y} \in \mathbb{R}^d \mid \|\boldsymbol{y}\|_2 < 1\}$, define the scaled stereographic projection $\phi : T \to B_1(0) \subseteq \mathbb{R}^d$ as

$$\phi(x_1, \ldots, x_{d+1}) = \left(\frac{c_v x_1}{1 - x_{d+1}}, \ldots, \frac{c_v x_d}{1 - x_{d+1}} \right),$$

where the normalization constant $c_v = \sqrt{\frac{1-v}{1+v}}$ ensures surjectivity.

Straightforward calculations show that ϕ defines a diffeomorphism. Its inverse $\phi^{-1} : B_1(0) \to T$ is given by

$$\phi^{-1}(y_1, \ldots, y_d) = \left(\frac{2c_v^{-1} y_1}{c_v^{-2}\|\boldsymbol{y}\|_2^2 + 1}, \ldots, \frac{2c_v^{-1} y_d}{c_v^{-2}\|\boldsymbol{y}\|_2^2 + 1}, \frac{c_v^{-2}\|\boldsymbol{y}\|_2^2 - 1}{c_v^{-2}\|\boldsymbol{y}\|_2^2 + 1} \right).$$

We can translate kernel learning with the kernel k on $\mathbb{S}^d$ and the probability distribution P, where P_X is supported on T, to kernel learning with a transformed kernel $\tilde{k}$ and $\tilde{P}$ using a sufficiently smooth diffeomorphism like $\phi : T \to B_1(0) \subseteq \mathbb{R}^d$. If the RKHS of k is equivalent to $H^s(\mathbb{S}^d)$ then the RKHS of $\tilde{k}$ is equivalent to $H^s(B_1(0))$. We formalize this argument in the following lemma. As a consequence it suffices to prove all inconsistency results for Sobolev kernels on $B_1(0)$.

Lemma E.1 (**Transfer to sphere caps**)**.** *Let k be a kernel on $\mathbb{S}^d$ whose RKHS is equivalent to a Sobolev space $H^s(\mathbb{S}^d)$. For fixed $v \in (-1,1)$, consider an "open sphere cap" $T \coloneqq \{\boldsymbol{x} \in \mathbb{S}^d \mid x_{d+1} < v\}$. Furthermore, consider a distribution P such that P_X is supported on T and has lower and upper bounded density p_X on T, i.e. $0 < C_l \leq p_X(\boldsymbol{x}) \leq C_u < \infty$ for all $\boldsymbol{x} \in T$. Then*

- *$\tilde{k}(\boldsymbol{x}, \boldsymbol{x}') \coloneqq k(\phi^{-1}(\boldsymbol{x}), \phi^{-1}(\boldsymbol{x}'))$ defines a positive definite kernel on $B_1(0) \subseteq \mathbb{R}^d$ whose RKHS is equivalent to the Sobolev space $H^s(B_1(0))$,*
- *$\tilde{P} \coloneqq P \circ \psi^{-1}$ with $\psi(\boldsymbol{x}, y) \coloneqq (\phi(\boldsymbol{x}), y)$ defines a probability distribution such that $\tilde{P}_{\tilde{X}}$ has lower and upper bounded density on $B_1(0) \subseteq \mathbb{R}^d$,*

and kernel learning with (k, P) or with $(\tilde{k}, \tilde{P})$ is equivalent in the following sense:

For every function $f \in \mathcal{H}(k|_T)$ the transformed function $\tilde{f} = f \circ \phi^{-1} \in \mathcal{H}(\tilde{k})$ has the same RKHS norm, i.e. $\|f\|_{\mathcal{H}(k|_T)} = \|\tilde{f}\|_{\mathcal{H}(\tilde{k})}$. Furthermore, the excess risks of f over P and of $\tilde{f}$ over $\tilde{P}$ coincide, i.e.

$$\mathbb{E}_{\boldsymbol{x} \sim P_X}(f(\boldsymbol{x}) - f_P^*(\boldsymbol{x}))^2 = \mathbb{E}_{\tilde{\boldsymbol{x}} \sim \tilde{P}_X}(\tilde{f}(\tilde{\boldsymbol{x}}) - \tilde{f}_{\tilde{P}}^*(\tilde{\boldsymbol{x}}))^2,$$

where $\tilde{f}_{\tilde{P}}^*(\tilde{\boldsymbol{x}}) = \mathbb{E}_{(\tilde{X}, \tilde{Y}) \sim \tilde{P}}(\tilde{Y} | \tilde{X} = \tilde{\boldsymbol{x}})$ denotes the Bayes optimal predictor under $\tilde{P}$.

Remark E.2. Many kernel regression estimators can be explicitly written as $f_D^k(\boldsymbol{x}) = \hat{f}_n(k(\boldsymbol{x}, \boldsymbol{X}), k(\boldsymbol{X}, \boldsymbol{X}), \boldsymbol{y})$ where $\hat{f}_n : \mathbb{R}^n \times \mathbb{R}^{n\times n} \times \mathbb{R}^n \to \mathbb{R}$ denotes a measurable function for all $n \in \mathbb{N}$. Then the explicit form is preserved under the transformation, i.e. $f \circ \phi^{-1} = f_{\tilde{D}}^{\tilde{k}}$ with the transformed data set $\tilde{D} = \{(\phi(\boldsymbol{x}_i), y_i)\}_{i\in[n]}$. ◀

Proof of Lemma E.1. **Step 1: Bounded density.** For $i \in [d], j \in [d+1]$, the partial derivatives of ϕ are given by

$$\partial_{x_j}\phi_i(\boldsymbol{x}) = \begin{cases} \frac{c_v}{1-x_{d+1}}, & \text{for } i = j, \\ \frac{c_v x_i}{(1-x_{d+1})^2}, & \text{for } i \in [d],\ j = d+1, \\ 0, & \text{otherwise.} \end{cases}$$

Given an arbitrary multi-index α, the partial derivatives $\partial_\alpha\phi_i \in L^2(T)$, $\partial_\alpha\phi_j^{-1} \in L^2(B_1(0))$ are bounded for all $i \in [d], j \in [d+1]$, using $x_{d+1} \le v < 1$ and the inverse function theorem.

Now define $\tilde{k}(\boldsymbol{x}, \boldsymbol{x}') \coloneqq k(\phi^{-1}(\boldsymbol{x}), \phi^{-1}(\boldsymbol{x}'))$, $\psi(\boldsymbol{x}, y) \coloneqq (\phi(\boldsymbol{x}), y)$ and $\tilde{P} \coloneqq P \circ \psi^{-1}$. Then using integration by substitution (Stroock et al., 2011, Theorem 5.2.16), the Lebesgue density of $\tilde{P}_X$ is given by

$$p_{\tilde{X}}(\tilde{\boldsymbol{x}}) = p_X(\phi^{-1}(\tilde{\boldsymbol{x}}))J\phi^{-1}(\tilde{\boldsymbol{x}}),$$

where

$$J\phi^{-1}(\tilde{\boldsymbol{x}}) \coloneqq \left[\det\left(\left(\langle\partial_i\phi^{-1}(\tilde{\boldsymbol{x}}), \partial_j\phi^{-1}(\tilde{\boldsymbol{x}})\rangle_{\mathbb{R}^{d+1}}\right)_{i,j\in\{1,\dots,d\}}\right)\right]^{1/2}.$$

$J\phi$ and $J\phi^{-1}$ can be continuously extended to $\bar{T}$ and $\bar{B}_1(0)$, respectively. Then, since $J\phi^{-1}$ is continuous on a compact set and because ϕ with the extended domain remains a diffeomorphism so that $J\phi^{-1}$ cannot attain the value 0, there exists a constant $C_\phi > 0$ such that $\frac{1}{C_\phi} \le J\phi^{-1}(\tilde{\boldsymbol{x}}) \le C_\phi$ for all $\tilde{\boldsymbol{x}} \in B_1(0)$. Hence, $p_{\tilde{X}}$ is lower and upper bounded.

Step 2: Excess risks coincide. If $(\tilde{X}, \tilde{Y}) \sim \tilde{P}$, the Bayes predictor of $\tilde{Y}$ given $\tilde{X}$ is given by $\tilde{f}^*(\tilde{\boldsymbol{x}}) = \mathbb{E}(\tilde{Y}|\tilde{X} = \tilde{\boldsymbol{x}}) = f^*(\phi^{-1}(\tilde{\boldsymbol{x}}))$.

Let $\pi_1(\boldsymbol{x}, y) = \boldsymbol{x}$ be the projection onto the first component. Then, $\phi(\pi_1(\boldsymbol{x}, y)) = \phi(\boldsymbol{x}) = \pi_1(\phi(\boldsymbol{x}), y) = \pi_1(\psi(\boldsymbol{x}, y))$ and hence

$$\begin{aligned}
\mathbb{E}_{\boldsymbol{x}\sim P_X}(f(\boldsymbol{x}) - f^*(\boldsymbol{x}))^2 &= \mathbb{E}_{(\boldsymbol{x},y)\sim P}(f(\pi_1(\boldsymbol{x}, y)) - f^*(\pi_1(\boldsymbol{x}, y)))^2 \\
&= \mathbb{E}_{(\boldsymbol{x},y)\sim P}(f(\phi^{-1}(\phi(\pi_1(\boldsymbol{x}, y)))) - f^*(\phi^{-1}(\phi(\pi_1(\boldsymbol{x}, y))))^2 \\
&= \mathbb{E}_{(\boldsymbol{x},y)\sim P}(\tilde{f}(\pi_1(\psi(\boldsymbol{x}, y))) - \tilde{f}^*(\pi_1(\psi(\boldsymbol{x}, y))))^2 \\
&= \mathbb{E}_{(\boldsymbol{x},y)\sim \tilde{P}}(\tilde{f}(\pi_1(\boldsymbol{x}, y)) - \tilde{f}^*(\pi_1(\boldsymbol{x}, y)))^2 \\
&= \mathbb{E}_{\boldsymbol{x}\sim \tilde{P}_{\tilde{X}}}(\tilde{f}(\boldsymbol{x}) - \tilde{f}^*(\boldsymbol{x}))^2 .
\end{aligned}$$

Step 3: Transformed RKHS. We want to show that $\mathcal{H}(k|_T) \to \mathcal{H}(\tilde{k}), f \mapsto f \circ \phi^{-1}$ defines an isometric isomorphism, which especially shows the statement $\|f\|_{\mathcal{H}(k|_T)} = \|\tilde{f}\|_{\mathcal{H}(\tilde{k})}$ from the proposition. For this, we use the following theorem characterizing RKHSs:

Theorem E.3 (Theorem 4.21 in Steinwart and Christmann (2008))**.** *Let $k : X \times X \to \mathbb{R}$ be a positive definite kernel function with feature space H_0 and feature map $\Phi_0 : X \to H_0$. Then*

$$\begin{aligned}
H = \{f : X \to \mathbb{R} \mid \exists w \in H_0 :\ f = \langle w, \Phi_0(\cdot)\rangle_{H_0}\} \text{ with} \\
\|f\|_H \coloneqq \inf\{\|w\|_{H_0} :\ f = \langle w, \Phi_0(\cdot)\rangle_{H_0}\},
\end{aligned}$$

is the only RKHS for which k is a reproducing kernel.

A feature map for $k|_T$ is given by $\Phi : T \to \mathcal{H}(k|_T)$, $\Phi(\boldsymbol{x}) = k(\boldsymbol{x}, \cdot)$. Hence a feature map for $\tilde{k}$ is given by $\Phi \circ \phi^{-1} : B_1(0) \to \mathcal{H}(k|_T)$. Theorem E.3 states that

$$\begin{aligned}
\mathcal{H}(k|_T) = \left\{f : T \to \mathbb{R} \mid \exists w \in \mathcal{H}(k|_T) :\ f = \langle w, \Phi(\cdot)\rangle_{\mathcal{H}(k|_T)}\right\} \text{ with} \\
\|f\|_{\mathcal{H}(k|_T)} \coloneqq \inf\{\|w\|_{\mathcal{H}(k|_T)} :\ f = \langle w, \Phi(\cdot)\rangle_{\mathcal{H}(k|_T)}\},
\end{aligned} \tag{E.1}$$

as well as

$$\mathcal{H}(\tilde{k}) = \left\{ \tilde{f} : B_1(0) \to \mathbb{R} \mid \exists w \in \mathcal{H}(k|_T) : \ \tilde{f} = \langle w, \Phi \circ \phi^{-1}(\cdot) \rangle_{\mathcal{H}(k|_T)} \right\} \text{ with} \qquad \text{(E.2)}$$
$$\|\tilde{f}\|_{\mathcal{H}(\tilde{k})} \coloneqq \inf\{ \|w\|_{\mathcal{H}(k|_T)} : \ \tilde{f} = \langle w, \Phi \circ \phi^{-1}(\cdot) \rangle_{\mathcal{H}(k|_T)} \}.$$

As ϕ^{-1} is bijective, this characterization induces an isometric isomorphism between $\mathcal{H}(k|_T)$ and $\mathcal{H}(\tilde{k})$ by mapping $f = \langle w, \Phi(\cdot) \rangle_{\mathcal{H}(k|_T)} \in \mathcal{H}(k|_T)$ to $\tilde{f} = f \circ \phi^{-1} = \langle w, \Phi \circ \phi^{-1}(\cdot) \rangle_{\mathcal{H}(k|_T)} \in \mathcal{H}(\tilde{k})$. This shows $\|f\|_{\mathcal{H}(k|_T)} = \|\tilde{f}\|_{\mathcal{H}(\tilde{k})}$.

Step 4: RKHS of $\tilde{k}$. We now show that the RKHS of $\tilde{k}$, denoted as $\mathcal{H}(\tilde{k})$, is equivalent to $H^s(B_1(0))$. To this end, denoting $\mathcal{A} \circ \phi \coloneqq \{f \circ \phi \mid f \in \mathcal{A}\}$ and $\mathcal{A}|_T \coloneqq \{f|_T \mid f \in \mathcal{A}\}$, we show the following equality of sets (ignoring the norms):

$$\mathcal{H}(\tilde{k}) \circ \phi \overset{\text{(I)}}{=} \mathcal{H}(k|_T) \overset{\text{(II)}}{=} \mathcal{H}(k)|_T \overset{\text{(III)}}{=} H^s(\mathbb{S}^d)|_T \overset{\text{(IV)}}{=} H^s(B_1(0)) \circ \phi \, .$$

Since ϕ is bijective, this implies $\mathcal{H}(\tilde{k}) = H^s(B_1(0))$ as sets, and the norm equivalence then follows from Lemma F.7.

Equality (I) follows from Step 3. Equality (II) follows from Theorem E.3 by observing that if Φ is a feature map for k, then $\Phi|_T$ is a feature map for $k|_T$. Equality (III) holds by assumption. To show (IV), we need a characterization of $H^s(\mathbb{S}^d)$ that allows to work with charts like ϕ.

Step 4.1: Chart-based characterization of $H^s(\mathbb{S}^d)$. A trivialization of a Riemannian manifold (M, g) with bounded geometry of dimension d consists of a locally finite open covering $\{U_\alpha\}_{\alpha \in I}$ of M, smooth diffeomorphisms $\kappa_\alpha : V_\alpha \subset \mathbb{R}^d \to U_\alpha$, also called charts, and a partition of unity $\{h_\alpha\}_{\alpha \in I}$ of M that fulfills $\operatorname{supp}(h_\alpha) \subseteq U_\alpha$, $0 \leq h_\alpha \leq 1$ and $\sum_{\alpha \in I} h_\alpha = 1$. An admissible trivialization of (M, g) is a uniformly locally finite trivialization of M that is compatible with geodesic coordinates, for details see (Schneider and Große, 2013, Definition 12).

In our case, define an open neighborhood of T by $U_1 \coloneqq \{\boldsymbol{x} \in \mathbb{S}^d \mid x_{d+1} < v + \varepsilon\}$ with some $\varepsilon \in (0, 1 - v)$ arbitrary but fixed, and $U_2 \coloneqq \{\boldsymbol{x} \in \mathbb{S}^d \mid x_{d+1} > v + \varepsilon/2\}$. It holds that $U_1 \cup U_2 = \mathbb{S}^d$. Moreover, there exists an appropriate partition of unity consisting of C^∞ functions $h_1, h_2 : \mathbb{S}^d \to [0, 1]$. Especially, we have $h_1(T) \subseteq h_1(U_2^c) = \{1\}$. Let $\phi_1 : U_1 \to B_{r_1}(0)$ denote the stereographic projection with respect to $\boldsymbol{x}_0 = (0, \ldots, 0, 1)$ as above, scaled such that $\phi_1|_T = \phi$ and hence $\phi_1(T) = B_1(0)$. Similarly, let $\phi_2 : U_2 \to B_{r_2}(0)$ denote an arbitrarily scaled stereographic projection with respect to $\boldsymbol{x}_0 = (0, \ldots, 0, -1)$. Then $(\{U_1, U_2\}, \{\phi_1^{-1}, \phi_1^{-1}\}, \{h_1, h_2\})$ yields an admissible trivialization of $\mathbb{S}^d$ consisting of only two charts. A detailed derivation can be found in (Hubbert et al., 2015, Section 1.7). Therefore (Schneider and Große, 2013, Theorem 14) lets us define the Sobolev norm on $\mathbb{S}^d$ (up to equivalence) as[2]

$$\begin{aligned} \|g\|_{H^s(\mathbb{S}^d)} &\coloneqq \left(\sum_{\alpha \in I} \|(h_\alpha g) \circ \kappa_\alpha\|^2_{H^s(\mathbb{R}^d)} \right)^{1/2} \\ &= \left(\|(h_1 g) \circ \phi_1^{-1}\|^2_{H^s(\mathbb{R}^d)} + \|(h_2 g) \circ \phi_2^{-1}\|^2_{H^s(\mathbb{R}^d)} \right)^{1/2}, \end{aligned}$$

for any distribution $g \in \mathcal{D}'(\mathbb{S}^d)$ (i.e. any continuous linear functional on $C_c^\infty(\mathbb{S}^d)$). Then $g \in H^s(\mathbb{S}^d)$ if and only if $\|g\|_{H^s(\mathbb{S}^d)} < \infty$.

Step 4.2: Showing (IV). First, let $g \in H^s(\mathbb{S}^d)$. Then, as we saw in Step 4.1, we must have $\|(h_1 g) \circ \phi_1^{-1}\|_{H^s(\mathbb{R}^d)} < \infty$ and thus $(h_1 g) \circ \phi_1^{-1} \in H^s(\mathbb{R}^d)$. By our discussion in Appendix B.1, we then have

$$(g|_T) \circ \phi^{-1} = ((h_1 g) \circ \phi_1^{-1})|_{B_1(0)} \in H^s(B_1(0)) \, ,$$

which shows $g|_T \in H^s(B_1(0)) \circ \phi$.

Now, let $f \in H^s(B_1(0))$. Then, again following our discussion in Appendix B.1, there exists an extension $\bar{f} \in H^s(\mathbb{R}^d)$ with $\bar{f}|_{B_1(0)} = f$. The set $\mathcal{B} \coloneqq \phi_1(U_1 \setminus U_2)$ is a closed ball $\overline{B_r(0)}$ of radius

[2] Here, the norms are taken on $H^s(\mathbb{R}^d)$ since the respective functions can be extended to $\mathbb{R}^d$ by zero outside of their domain of definition, thanks to the properties of the partition of unity.

$1 < r < r_1$. Hence, we can find $\varphi \in C^\infty(\mathbb{R}^d)$ with $\varphi(B_1(0)) = \{1\}$ and $\varphi((\overline{B_r(0)})^c) = \{0\}$. Since φ is smooth with compact support, we have $\varphi \cdot \bar{f} \in H^s(\mathbb{R}^d)$. Define

$$f_{\mathbb{S}^d} : \mathbb{S}^d \to \mathbb{R}, \boldsymbol{x} \mapsto \begin{cases} (\varphi \cdot \bar{f})(\phi(\boldsymbol{x})) & , \boldsymbol{x} \in U_1 \\ 0 & , \boldsymbol{x} \notin U_1 \end{cases}.$$

By construction, we have $f_{\mathbb{S}^d}(\boldsymbol{x}) = 0$ for all $\boldsymbol{x} \in U_2$. Hence, the equivalent Sobolev norm from Step 4.1 is

$$\begin{aligned} \|f_{\mathbb{S}^d}\|_{H^s(\mathbb{S}^d)} &= \left(\|(h_1 f_{\mathbb{S}^d}) \circ \phi_1^{-1}\|^2_{H^s(\mathbb{R}^d)} + \|(h_2 f_{\mathbb{S}^d}) \circ \phi_2^{-1}\|^2_{H^s(\mathbb{R}^d)} \right)^{1/2} \\ &= \|(h_1 \circ \phi_1^{-1}) \cdot \varphi \cdot \bar{f}\|_{H^s(\mathbb{R}^d)} < \infty , \end{aligned}$$

which shows $f_{\mathbb{S}^d} \in H^s(\mathbb{S}^d)$. But then, $f \circ \phi = f_{\mathbb{S}^d}|_T \in H^s(\mathbb{S}^d)|_T$.

In total, we obtain $H^s(\mathbb{S}^d)|_T = H^s(B_1(0)) \circ \phi$, which shows (IV). □

F Spectral lower bound

F.1 General lower bounds

A common first step to analyze the expected excess risk caused by label noise is to perform a bias-variance decomposition and integrate over $\boldsymbol{y}$ first (see e.g. Liang and Rakhlin, 2020, Hastie et al., 2022, Holzmüller, 2021), which is also used in the following lemma.

Lemma F.1. *Consider an estimator of the form $f_{\boldsymbol{X},\boldsymbol{y}}(\boldsymbol{x}) = (\boldsymbol{v}_{\boldsymbol{X},\boldsymbol{x}})^\top \boldsymbol{y}$. If $\mathrm{Var}_P(y|\boldsymbol{x}) \geq \sigma^2$ for P_X-almost all $\boldsymbol{x}$, then the expected excess risk satisfies*

$$\mathbb{E}_D R_P(f_{\boldsymbol{X},\boldsymbol{y}}) - R_P^* \geq \sigma^2 \mathbb{E}_{\boldsymbol{X},\boldsymbol{x}} \operatorname{tr}(\boldsymbol{v}_{\boldsymbol{X},\boldsymbol{x}} (\boldsymbol{v}_{\boldsymbol{X},\boldsymbol{x}})^\top) .$$

Proof. A standard bias-variance decomposition lets us lower-bound the expected excess risk by the estimator variance due to the label noise, which can then be further simplified:

$$\begin{aligned} \mathbb{E}_D R_P(f_{\boldsymbol{X},\boldsymbol{y}}) - R_P^* &\geq \mathbb{E}_{\boldsymbol{X},\boldsymbol{x}} \left(\mathbb{E}_{\boldsymbol{y}|\boldsymbol{X}} \left[f_{\boldsymbol{X},\boldsymbol{y}}(\boldsymbol{x})^2 \right] - \left(\mathbb{E}_{\boldsymbol{y}|\boldsymbol{X}} [f_{\boldsymbol{X},\boldsymbol{y}}(\boldsymbol{x})] \right)^2 \right) . \\ &= \mathbb{E}_{\boldsymbol{X},\boldsymbol{x}} \mathbb{E}_{\boldsymbol{y}|\boldsymbol{X}} \left(f_{\boldsymbol{X},\boldsymbol{y}}(\boldsymbol{x}) - \mathbb{E}_{\boldsymbol{y}|\boldsymbol{X}} f_{\boldsymbol{X},\boldsymbol{y}}(\boldsymbol{x}) \right)^2 \\ &= \mathbb{E}_{\boldsymbol{X},\boldsymbol{x}} \mathbb{E}_{\boldsymbol{y}|\boldsymbol{X}} (\boldsymbol{v}_{\boldsymbol{X},\boldsymbol{x}})^\top (\boldsymbol{y} - \mathbb{E}_{\boldsymbol{y}|\boldsymbol{X}} \boldsymbol{y})(\boldsymbol{y} - \mathbb{E}_{\boldsymbol{y}|\boldsymbol{X}} \boldsymbol{y})^\top \boldsymbol{v}_{\boldsymbol{X},\boldsymbol{x}} \\ &= \mathbb{E}_{\boldsymbol{X},\boldsymbol{x}} (\boldsymbol{v}_{\boldsymbol{X},\boldsymbol{x}})^\top \left[\mathbb{E}_{\boldsymbol{y}|\boldsymbol{X}} (\boldsymbol{y} - \mathbb{E}_{\boldsymbol{y}|\boldsymbol{X}} \boldsymbol{y})(\boldsymbol{y} - \mathbb{E}_{\boldsymbol{y}|\boldsymbol{X}} \boldsymbol{y})^\top \right] \boldsymbol{v}_{\boldsymbol{X},\boldsymbol{x}} \\ &= \mathbb{E}_{\boldsymbol{X},\boldsymbol{x}} (\boldsymbol{v}_{\boldsymbol{X},\boldsymbol{x}})^\top \operatorname{Cov}(\boldsymbol{y}|\boldsymbol{X}) \boldsymbol{v}_{\boldsymbol{X},\boldsymbol{x}} . \end{aligned}$$

Here, the conditional covariance matrix can be lower bounded in terms of the Loewner order (which is defined as $A \succeq B \Leftrightarrow B - A$ positive semi-definite):

$$\operatorname{Cov}(\boldsymbol{y}|\boldsymbol{X}) = \begin{pmatrix} \mathrm{Var}(y_1|\boldsymbol{x}_1) & & \\ & \ddots & \\ & & \mathrm{Var}(y_n|\boldsymbol{x}_n) \end{pmatrix} \succeq \sigma^2 \boldsymbol{I}_n$$

since the labels y_i are conditionally independent given $\boldsymbol{X}$. We therefore obtain

$$\begin{aligned} \mathbb{E}_D R_P(f_{\boldsymbol{X},\boldsymbol{y}}) - R_P^* &\geq \mathbb{E}_{\boldsymbol{X},\boldsymbol{x}} (\boldsymbol{v}_{\boldsymbol{X},\boldsymbol{x}})^\top \operatorname{Cov}(\boldsymbol{y}|\boldsymbol{X}) \boldsymbol{v}_{\boldsymbol{X},\boldsymbol{x}} \\ &\geq \sigma^2 \mathbb{E}_{\boldsymbol{X},\boldsymbol{x}} \operatorname{tr}((\boldsymbol{v}_{\boldsymbol{X},\boldsymbol{x}})^\top \boldsymbol{v}_{\boldsymbol{X},\boldsymbol{x}}) \\ &= \sigma^2 \mathbb{E}_{\boldsymbol{X},\boldsymbol{x}} \operatorname{tr}(\boldsymbol{v}_{\boldsymbol{X},\boldsymbol{x}} (\boldsymbol{v}_{\boldsymbol{X},\boldsymbol{x}})^\top) . \end{aligned}$$

□

Proposition 5 (Spectral lower bound). *Assume that the kernel matrix $k(\boldsymbol{X}, \boldsymbol{X})$ is almost surely positive definite, and that $\mathrm{Var}(y|\boldsymbol{x}) \geq \sigma^2$ for P_X-almost all $\boldsymbol{x}$. Then, the expected excess risk satisfies*

$$\mathbb{E}_D R_P(f_{t,\rho}) - R_P^* \geq \frac{\sigma^2}{n} \sum_{i=1}^n \mathbb{E}_{\boldsymbol{X}} \frac{\lambda_i(k_*(\boldsymbol{X},\boldsymbol{X})/n) \left(1 - e^{-2t(\lambda_i(k(\boldsymbol{X},\boldsymbol{X})/n)+\rho)}\right)^2}{(\lambda_i(k(\boldsymbol{X},\boldsymbol{X})/n) + \rho)^2} . \quad (3)$$

Proof. Recall from Eq. (1) that

$$f_{t,\rho}(\boldsymbol{x}) = k(\boldsymbol{x},\boldsymbol{X})\boldsymbol{A}_{t,\rho}(\boldsymbol{X})\boldsymbol{y}\ ,$$
$$\boldsymbol{A}_{t,\rho}(\boldsymbol{X}) \coloneqq \left(\boldsymbol{I}_n - e^{-\frac{2}{n}t(k(\boldsymbol{X},\boldsymbol{X})+\rho n\boldsymbol{I}_n)}\right)\left(k(\boldsymbol{X},\boldsymbol{X})+\rho n\boldsymbol{I}_n\right)^{-1}\ .$$

By setting $(\boldsymbol{v}_{\boldsymbol{X},\boldsymbol{x}})^\top \coloneqq k(\boldsymbol{x},\boldsymbol{X})\boldsymbol{A}_{t,\rho}(\boldsymbol{X})$, we can write $f_{\boldsymbol{X},\boldsymbol{y},t,\rho}(\boldsymbol{x}) \coloneqq f_{t,\rho}(\boldsymbol{x}) = (\boldsymbol{v}_{\boldsymbol{X},\boldsymbol{x}})^\top\boldsymbol{y}$. Using Lemma F.1, we then obtain

$$\begin{aligned}\mathbb{E}_D R_P(f_{\boldsymbol{X},\boldsymbol{y},t,\rho}) - R_P^* &\geq \sigma^2\mathbb{E}_{\boldsymbol{X},\boldsymbol{x}}\operatorname{tr}(\boldsymbol{v}_{\boldsymbol{X},\boldsymbol{x}}(\boldsymbol{v}_{\boldsymbol{X},\boldsymbol{x}})^\top)\\ &= \sigma^2\mathbb{E}_{\boldsymbol{X},\boldsymbol{x}}\operatorname{tr}\left(\boldsymbol{A}_{t,\rho}(\boldsymbol{X})^\top k(\boldsymbol{X},\boldsymbol{x})k(\boldsymbol{x},\boldsymbol{X})\boldsymbol{A}_{t,\rho}(\boldsymbol{X})\right)\ .\end{aligned}$$

Since

$$\left(\mathbb{E}_{\boldsymbol{x}}k(\boldsymbol{X},\boldsymbol{x})k(\boldsymbol{x},\boldsymbol{X})\right)_{ij} = \mathbb{E}_{\boldsymbol{x}}k(\boldsymbol{x}_i,\boldsymbol{x})k(\boldsymbol{x},\boldsymbol{x}_j) = k_*(\boldsymbol{x}_i,\boldsymbol{x}_j) = k_*(\boldsymbol{X},\boldsymbol{X})_{ij}\ ,$$

we conclude

$$\begin{aligned}\mathbb{E}_D R_P(f_{\boldsymbol{X},\boldsymbol{y},t,\rho}) - R_P^* &\geq \sigma^2\mathbb{E}_{\boldsymbol{X}}\operatorname{tr}(\boldsymbol{A}_{t,\rho}^\top k_*(\boldsymbol{X},\boldsymbol{X})\boldsymbol{A}_{t,\rho})\\ &= \sigma^2\mathbb{E}_{\boldsymbol{X}}\operatorname{tr}(k_*(\boldsymbol{X},\boldsymbol{X})\boldsymbol{A}_{t,\rho}(\boldsymbol{X})\boldsymbol{A}_{t,\rho}(\boldsymbol{X})^\top)\ .\end{aligned}$$

Richter (1958) showed (see also Mirsky, 1959) that for two symmetric matrices $\boldsymbol{B},\boldsymbol{C}$, we have $\operatorname{tr}(\boldsymbol{B}\boldsymbol{C}) \geq \sum_{i=1}^n \lambda_i(\boldsymbol{B})\lambda_{n+1-i}(\boldsymbol{C})$. We can therefore conclude

$$\mathbb{E}_D R_P(f_{\boldsymbol{X},\boldsymbol{y},t,\rho}) - R_P^* \geq \sigma^2\mathbb{E}_{\boldsymbol{X}}\sum_{i=1}^n \lambda_i(k_*(\boldsymbol{X},\boldsymbol{X}))\lambda_{n+1-i}(\boldsymbol{A}_{t,\rho}(\boldsymbol{X})\boldsymbol{A}_{t,\rho}(\boldsymbol{X})^\top)\ .$$

As $\boldsymbol{A}_{t,\rho}(\boldsymbol{X})\boldsymbol{A}_{t,\rho}(\boldsymbol{X})^\top$ is built only out of the matrices $k(\boldsymbol{X},\boldsymbol{X})$ and $\boldsymbol{I}_n$, it is not hard to see that $\boldsymbol{A}_{t,\rho}(\boldsymbol{X})\boldsymbol{A}_{t,\rho}(\boldsymbol{X})^\top$ has the same eigenbasis as $k(\boldsymbol{X},\boldsymbol{X})$ with eigenvalues

$$\tilde{\lambda}_i \coloneqq \left(\frac{1-e^{-\frac{2}{n}t(\lambda_i(k(\boldsymbol{X},\boldsymbol{X}))+\rho n)}}{\lambda_i(k(\boldsymbol{X},\boldsymbol{X}))+\rho n}\right)^2 = \frac{1}{n^2}\left(\frac{1-e^{-2t(\lambda_i(k(\boldsymbol{X},\boldsymbol{X})/n)+\rho)}}{\lambda_i(k(\boldsymbol{X},\boldsymbol{X})/n)+\rho}\right)^2\ .$$

It remains to order these eigenvalues correctly. To this end, we observe that for $\lambda > 0$, the function $g(\lambda) \coloneqq \frac{1-e^{-2t\lambda}}{\lambda}$ satisfies

$$g'(\lambda) = \frac{2t\lambda e^{-2t\lambda} - (1-e^{-2t\lambda})}{\lambda^2} = \frac{(2t\lambda+1)e^{-2t\lambda}-1}{\lambda^2} \leq \frac{e^{2t\lambda}e^{-2t\lambda}-1}{\lambda^2} = 0\ .$$

Therefore, g is nonincreasing, hence the sequence $(\tilde{\lambda}_i)$ is nondecreasing and thus

$$\lambda_{n+1-i}(\boldsymbol{A}_{t,\rho}\boldsymbol{A}_{t,\rho}^\top) = \tilde{\lambda}_i\ ,$$

from which the claim follows. □

Theorem F.2. *Let k be a kernel on a compact set Ω and let P_X be supported on Ω. Suppose that $k(\boldsymbol{X},\boldsymbol{X})$ is almost surely positive definite and that $\operatorname{Var}(y|\boldsymbol{x}) \geq \sigma^2$ for P_X-almost all $\boldsymbol{x}$. Fix constants $c > 0$ and $q, C \geq 1$. Suppose that $\lambda_i \coloneqq \lambda_i(T_{k,P_X}) \geq ci^{-q}$. Let $\mathcal{I}(n)$ be the set of all $i \in [n]$ for which*

$$\begin{aligned}&\lambda_i/C \leq \lambda_i(k(\boldsymbol{X},\boldsymbol{X})/n) \leq C\lambda_i\\ &\lambda_i^2/C \leq \lambda_i(k_*(\boldsymbol{X},\boldsymbol{X})/n)\end{aligned} \tag{F.1}$$

both hold at the same time with probability $\geq 1/2$. Moreover, let $I(n) \coloneqq \max\{m \in [n] \mid [m] \subseteq \mathcal{I}(n)\}$. Then, there exists a constant $c' > 0$ depending only on c, C such that for all $\rho \in [0,\infty)$ and $t \in (0,\infty]$, the following two bounds hold:

$$\mathbb{E}_D R_P(f_{\boldsymbol{X},\boldsymbol{y},t,\rho}) - R_P^* \geq c'\sigma^2\frac{1}{1+(\rho+t^{-1})n^q}\cdot\frac{|\mathcal{I}(n)|}{n}\ ,$$
$$\mathbb{E}_D R_P(f_{\boldsymbol{X},\boldsymbol{y},t,\rho}) - R_P^* \geq c'\sigma^2\min\left\{\frac{(\rho+t^{-1})^{-2}}{n}, \frac{(\rho+t^{-1})^{-1/q}}{n}, \frac{I(n)}{n}\right\}\ .$$

Remark F.3. Theorem F.2 provides two lower bounds, one for general "concentration sets" $\mathcal{I}(n)$ and one that applies if concentration holds for a sequence of "head eigenvalues" $\{1, \dots, I(n)\} \subseteq \mathcal{I}(n)$. If $I(n) \approx |\mathcal{I}(n)|$, the latter bound is stronger for larger regularization levels, and this bound would be particularly suitable for typical forms of relative concentration inequalities for kernel matrices. However, in this paper, we obtain concentration only for "middle eigenvalues" $\mathcal{I}(n) = \{i \mid \varepsilon n \leq i \leq (1-\varepsilon)n\}$, and therefore we only use the first bound in the proof of Theorem 6. ◀

Proof of Theorem F.2. **Step 1: Miscellaneous inequalities.** For $x > 0$,

$$1 - e^{-x} = 1 - \frac{1}{e^x} \geq 1 - \frac{1}{x+1} = \frac{x}{x+1} = \frac{1}{1+x^{-1}} \,. \tag{F.2}$$

Moreover, since $(1+a)^2 \leq (1+a)^2 + (1-a)^2 = 2 + 2a^2$, we have for $a \neq -1$:

$$\left(\frac{1}{1+a}\right)^2 \geq \frac{1}{2(1+a^2)} \,. \tag{F.3}$$

Step 2: Applying the eigenvalue bound. Define

$$S_i(\boldsymbol{X}) \coloneqq \frac{\lambda_i(k_*(\boldsymbol{X},\boldsymbol{X})/n)\left(1 - e^{-2t(\lambda_i(k(\boldsymbol{X},\boldsymbol{X})/n)+\rho)}\right)^2}{(\lambda_i(k(\boldsymbol{X},\boldsymbol{X})/n)+\rho)^2} \,.$$

By Proposition 5, we have

$$\mathbb{E}_D R_P(f_{\boldsymbol{X},\boldsymbol{y},t,\rho}) - R_P^* \geq \frac{\sigma^2}{n}\sum_{i=1}^{n} \mathbb{E}_{\boldsymbol{X}} S_i(\boldsymbol{X}) \geq \frac{\sigma^2}{n}\sum_{i\in\mathcal{I}(n)} \mathbb{E}_{\boldsymbol{X}} S_i(\boldsymbol{X}) \,. \tag{F.4}$$

Since $S_i(\boldsymbol{X})$ is almost surely positive, we can focus on the case where (F.1) and (F.2) hold, which is true with probability $\geq 1/2$ by assumption for $i \in \mathcal{I}(n)$. Hence,

$$\mathbb{E}_{\boldsymbol{X}} S_i(\boldsymbol{X}) \geq \frac{1}{2}\frac{\lambda_i^2/C \cdot (1 - e^{-2t(\lambda_i/C+\rho)})^2}{(C\lambda_i+\rho)^2} \overset{(F.2)}{\geq} \frac{1}{2}\frac{\lambda_i^2/C}{(C\lambda_i+\rho)^2(1+(2t(\lambda_i/C+\rho))^{-1})^2} \,.$$

We can upper-bound the denominator, using $C \geq 1$, as

$$\begin{aligned}(C\lambda_i+\rho)\left(1+\frac{1}{2t(\lambda_i/C+\rho)}\right) &\leq C\lambda_i + \rho + \frac{C^2\lambda_i + C\rho}{(\lambda_i + C\rho)t} \leq C\lambda_i + \rho + \frac{C^2\lambda_i + C^3\rho}{(\lambda_i+C\rho)t} \\ &\leq C^2\left(\lambda_i + \rho + t^{-1}\right) \,,\end{aligned}$$

which yields

$$\begin{aligned}\mathbb{E}_{\boldsymbol{X}} S_i(\boldsymbol{X}) &\geq \frac{1}{2C^5}\frac{\lambda_i^2}{(\lambda_i+\rho+t^{-1})^2} = \frac{1}{2C^5}\frac{1}{\left(1+\frac{\rho+t^{-1}}{\lambda_i}\right)^2} \overset{(F.3)}{\geq} \frac{1}{4C^5}\frac{1}{1+\left(\frac{\rho+t^{-1}}{\lambda_i}\right)^2} \\ &\geq \frac{1}{4C^5}\frac{1}{1+\left(\frac{\rho+t^{-1}}{c}i^q\right)^2} \,.\end{aligned} \tag{F.5}$$

Step 3: Analyzing the sum. We want to analyze the behavior of the sum

$$S(\beta) \coloneqq \sum_{i\in\mathcal{I}(n)} \frac{1}{1+(\beta i^q)^2}$$

for $\beta \coloneqq \frac{\rho+t^{-1}}{c} > 0$. We first obtain the trivial bound

$$S(\beta) \geq |\mathcal{I}(n)|\frac{1}{1+(\beta n^q)^2} \,.$$

Moreover, we can bound

$$S(\beta) \geq \sum_{i=1}^{I(n)} \frac{1}{1+(\beta i^q)^2}$$

and distinguish three cases:

(a) If $\beta \geq 1$, we bound

$$S(\beta) \geq \sum_{i=1}^{I(n)} \frac{1}{2(\beta i^q)^2} \geq \frac{1}{2\beta^2} .$$

(b) If $\beta \in (I(n)^{-q}, 1)$, we observe that

$$J(\beta) \coloneqq \lfloor \beta^{-1/q} \rfloor \geq \lceil \beta^{-1/q} \rceil - 1 \geq \frac{1}{2} \lceil \beta^{-1/q} \rceil \geq \frac{\beta^{-1/q}}{2}$$

and therefore

$$S(\beta) \geq \sum_{i=1}^{J(\beta)} \frac{1}{1 + (\beta i^q)^2} \geq \sum_{i=1}^{J(\beta)} \frac{1}{1+1} = \frac{J(\beta)}{2} \geq \frac{\beta^{-1/q}}{4} .$$

(c) If $\beta \in (0, I(n)^{-q}]$, we similarly find that

$$S(\beta) \geq \sum_{i=1}^{I(n)} \frac{1}{1+1} = \frac{I(n)}{2} .$$

Moreover, there is an absolute constant $c_1 > 0$ such that for any $\beta > 0$,

$$S(\beta) \geq c_1 \min\{\beta^{-2}, \beta^{-1/q}, I(n)\} , \tag{F.6}$$

because

(a) $\beta^{-2} = \min\{\beta^{-2}, \beta^{-1/q}, I(n)\}$ for $\beta \geq 1$,
(b) $\beta^{-1/q} = \min\{\beta^{-2}, \beta^{-1/q}, I(n)\}$ for $\beta \in (I(n)^{-q}, 1)$, and
(c) $I(n) = \min\{\beta^{-2}, \beta^{-1/q}, I(n)\}$ for $\beta \in (0, I(n)^{-q}]$.

Step 4: Putting it together. Combining the trivial bound in Step 3 with Eq. (F.4) and Eq. (F.5), we obtain

$$\begin{aligned} \mathbb{E}_D R_P(f_{\boldsymbol{X},\boldsymbol{y},t,\rho}) - R_P^* &\geq \frac{\sigma^2}{n} \sum_{i \in \mathcal{I}(n)} \mathbb{E}_{\boldsymbol{X}} S_i(\boldsymbol{X}) \geq \frac{\sigma^2}{n} \cdot \frac{1}{4C^5} S(\beta) \\ &\geq c' \sigma^2 \frac{1}{1 + (\rho + t^{-1}) n^q} \cdot \frac{|\mathcal{I}(n)|}{n} \end{aligned} \tag{F.7}$$

for a suitable constant $c' > 0$ depending only on c and C.

Moreover, from Eq. (F.6), we obtain

$$S(\beta) \geq \tilde{c}_1 \min\{\beta^{-2}, \beta^{-1/q}, I(n)\} \geq \tilde{c}'' \min\{(\rho + t^{-1})^{-2}, (\rho + t^{-1})^{-1/q}, I(n)\}$$

for a suitable constant $\tilde{c}'' > 0$ depending only on c. Again, (F.4) and (F.5) yield

$$\begin{aligned} \mathbb{E}_D R_P(f_{\boldsymbol{X},\boldsymbol{y},t,\rho}) - R_P^* &\geq \frac{\sigma^2}{n} \cdot \frac{1}{4C^5} S(\beta) \\ &\geq \frac{\tilde{c}''}{4C^5} \sigma^2 \min\left\{ \frac{(\rho + t^{-1})^{-2}}{n}, \frac{(\rho + t^{-1})^{-1/q}}{n}, \frac{I(n)}{n} \right\} . \end{aligned}$$

□

F.2 Equivalences of norms and eigenvalues

Later, we will use concentration inequalities for kernel matrix eigenvalues proved for specific kernels, which we then want to transfer to other kernels with equivalent RKHSs. In this subsection, we show that this is possible.

Definition F.4 (C-equivalence of matrices and norms)**.** Let $n \geq 1$ and let $\boldsymbol{K}, \tilde{\boldsymbol{K}} \in \mathbb{R}^{n \times n}$ be symmetric. For $C \geq 1$, we say that $\boldsymbol{K}$ and $\tilde{\boldsymbol{K}}$ are C-equivalent if their ordered eigenvalues satisfy

$$C^{-1} \lambda_i(\boldsymbol{K}) \leq \lambda_i(\tilde{\boldsymbol{K}}) \leq C \lambda_i(\boldsymbol{K})$$

for all $i \in [n]$. Moreover, we say that two norms $\|\cdot\|_A$, $\|\cdot\|_B$ on a vector space V are C-equivalent if

$$C^{-1} \|\boldsymbol{v}\|_A \leq \|\boldsymbol{v}\|_B \leq C \|\boldsymbol{v}\|_A$$

for all $\boldsymbol{v} \in V$. ◀

Lemma F.5. *Let $n \geq 1$ and let $\boldsymbol{K}, \tilde{\boldsymbol{K}} \in \mathbb{R}^{n \times n}$ be symmetric. Then, $\boldsymbol{K}$ and $\tilde{\boldsymbol{K}}$ are C-equivalent iff the Moore-Penrose pseudoinverses $\boldsymbol{K}^+$ and $\tilde{\boldsymbol{K}}^+$ are C-equivalent.*

Proof. This follows from the fact that if $\boldsymbol{K}$ has eigenvalues $\lambda_1, \dots, \lambda_n$, then $\boldsymbol{K}^+$ has eigenvalues $1/\lambda_1, \dots, 1/\lambda_n$, where we define $1/0 \coloneqq 0$. (A detailed proof would be a bit technical due to the sorting of eigenvalues.) □

Lemma F.6. *Let $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ be a kernel on a set $\mathcal{X}$. Then, for any $\boldsymbol{y} \in \mathbb{R}^n$,*

$$\boldsymbol{y}^\top k(\boldsymbol{X}, \boldsymbol{X})^+ \boldsymbol{y} = \|f^*_{k,\boldsymbol{y}}\|^2_{\mathcal{H}_k} ,$$

where $\mathcal{H}_k$ is the RKHS associated with k and $f^*_{k,\boldsymbol{y}}$ is the minimum-norm regression solution

$$f^*_{k,\boldsymbol{y}} \coloneqq \operatorname*{argmin}_{f \in B} \|f\|^2_{\mathcal{H}_k},$$
$$B \coloneqq \{f \in \mathcal{H}_k \mid \sum_{i=1}^n (f(\boldsymbol{x}_i) - y_i)^2 = \inf_{\tilde{f} \in \mathcal{H}_k} \sum_{i=1}^n (\tilde{f}(\boldsymbol{x}_i) - y_i)^2\} .$$

Proof. It is well-known that $f^*_{k,\boldsymbol{y}}(\boldsymbol{x}) = \sum_{i=1}^n \alpha_i k(\boldsymbol{x}, \boldsymbol{x}_i)$, where $\boldsymbol{\alpha} \coloneqq \boldsymbol{K}^+ \boldsymbol{y}$ (see e.g. Rangamani et al., 2023). We then have

$$\begin{aligned}\|f^*_{k,\boldsymbol{y}}\|^2_{\mathcal{H}_k} &= \left\langle \sum_{i=1}^n \alpha_i k(\boldsymbol{x}_i, \cdot), \sum_{j=1}^n \alpha_j k(\boldsymbol{x}_j, \cdot) \right\rangle_{\mathcal{H}_k} = \sum_{i=1}^n \sum_{j=1}^n \alpha_i \boldsymbol{K}_{ij} \alpha_j \\ &= \boldsymbol{y}^\top \boldsymbol{K}^+ \boldsymbol{K} \boldsymbol{K}^+ \boldsymbol{y} = \boldsymbol{y}^\top \boldsymbol{K}^+ \boldsymbol{y} ,\end{aligned}$$

where the last step follows from a standard identity for the Moore-Penrose pseudoinverse (see e.g. Section 1.1.1 in Wang et al., 2018). □

Lemma F.7. *Let $\mathcal{H}_1$ and $\mathcal{H}_2$ be two RKHSs with $\mathcal{H}_1 \subset \mathcal{H}_2$. Then there exists a constant $C > 0$ such that $\|f\|_{\mathcal{H}_2} \leq C \|f\|_{\mathcal{H}_1}$.*

Proof. Let $I : \mathcal{H}_1 \to \mathcal{H}_2$ be the inclusion map, i.e. $I_h \coloneqq h$ for all $h \in \mathcal{H}_1$. Obviously, I is linear and we need to show that I is bounded. To this end, let $(h_n)_{n \geq 1} \subset \mathcal{H}_1$ be a sequence such that there exist $h \in \mathcal{H}_1$ and $g \in \mathcal{H}_2$ with $h_n \to h$ in $\mathcal{H}_1$ and $I h_n \to g$ in $\mathcal{H}_2$. This implies $h_n \to h$ pointwise and $h_n = I h_n \to g$ pointwise, which in turn gives $h = g$. The closed graph theorem, see e.g. (Megginson, 1998, Theorem 1.6.11), then shows that I is bounded. □

Applying Lemma F.7 twice shows that RKHSs $\mathcal{H}_1$ and $\mathcal{H}_2$ with $\mathcal{H}_1 = \mathcal{H}_2$ automatically have C-equivalent norms for a suitable constant $C \geq 1$. The following result investigates the corresponding kernels.

Proposition F.8 (Equivalent kernels have equivalent kernel matrices)**.** *Let $k, \tilde{k} : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ be kernels such that their RKHSs are equal as sets and the corresponding RKHS-norms are C-equivalent as defined in Definition F.4. Then, for any $n \geq 1$ and any $\boldsymbol{x}_1, \dots, \boldsymbol{x}_n \in \mathcal{X}$, the corresponding kernel matrices $k(\boldsymbol{X}, \boldsymbol{X}), \tilde{k}(\boldsymbol{X}, \boldsymbol{X})$ are C^2-equivalent.*

Proof. Let $i \in [n]$. For $\boldsymbol{y} \in \mathbb{R}^n$ we have, using the notation of Lemma F.6:

$$\boldsymbol{y}^\top k(\boldsymbol{X}, \boldsymbol{X})^+ \boldsymbol{y} = \|f^*_{k,\boldsymbol{y}}\|^2_{\mathcal{H}_k} \geq C^{-2} \|f^*_{k,\boldsymbol{y}}\|^2_{\mathcal{H}_{\tilde{k}}} \geq C^{-2} \|f^*_{\tilde{k},\boldsymbol{y}}\|^2_{\mathcal{H}_{\tilde{k}}} = C^{-2} \boldsymbol{y}^\top \tilde{k}(\boldsymbol{X}, \boldsymbol{X})^+ \boldsymbol{y} .$$

Now, by the Courant-Fischer-Weyl theorem,

$$\begin{aligned}\lambda_i(k(\boldsymbol{X}, \boldsymbol{X})^+) &= \sup_{V : \dim V = i} \inf_{y \in V : \|y\|_2 = 1} y^\top k(\boldsymbol{X}, \boldsymbol{X})^+ y \\ &\geq C^{-2} \sup_{V : \dim V = i} \inf_{y \in V : \|y\|_2 = 1} y^\top \tilde{k}(\boldsymbol{X}, \boldsymbol{X})^+ y \\ &= C^{-2} \lambda_i(\tilde{k}(\boldsymbol{X}, \boldsymbol{X})^+) .\end{aligned}$$

By switching the roles of k and $\tilde{k}$, we obtain that $k(\boldsymbol{X}, \boldsymbol{X})^+$ and $\tilde{k}(\boldsymbol{X}, \boldsymbol{X})^+$ are C^2-equivalent. By Lemma F.5 $k(\boldsymbol{X}, \boldsymbol{X})$ and $\tilde{k}(\boldsymbol{X}, \boldsymbol{X})$ are then also C^2-equivalent. □

To prove Theorem 6 for arbitrary input distributions P_X with lower and upper bounded densities, we need the following theorem investigating the corresponding eigenvalues of the integral operator.

Lemma F.9 (Integral operators for equivalent densities have equivalent eigenvalues)**.** *Let $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ be a kernel and let μ, ν be finite measures on $\mathcal{X}$ whose support is $\mathcal{X}$ such that ν has an lower and upper bounded density w.r.t. μ. Then, $\lambda_i(T_{k,\nu}) = \Theta(\lambda_i(T_{k,\mu}))$.*

Proof. Let p be such an upper bounded density, that is, $\mathrm{d}\nu = p\,\mathrm{d}\mu$ and there exist $c, C > 0$ such that $c \leq p(\boldsymbol{x}) \leq C$ for all $\boldsymbol{x} \in \mathcal{X}$. For $f \in L_2(\nu)$, we have

$$\|p \cdot f\|^2_{L_2(\mu)} = \int f^2 p^2 \,\mathrm{d}\mu \leq C \int f^2 p \,\mathrm{d}\mu = C \int f^2 \,\mathrm{d}\nu = C\|f\|^2_{L_2(\nu)} \,.$$

Hence, the linear operator

$$A : L_2(\nu) \to L_2(\mu), f \mapsto p \cdot f$$

is well-defined and continuous. It is also easily verified that A is bijective. Moreover, we have

$$\langle Af, Af\rangle_{L_2(\mu)} = \int f^2 p^2 \,\mathrm{d}\mu \geq c \int f^2 p \,\mathrm{d}\mu = c \int f^2 \,\mathrm{d}\nu = c\langle f, f\rangle_{L_2(\nu)} \,.$$

and

$$\langle f, T_{k,\nu} f\rangle_{L_2(\nu)} = \int\int p(\boldsymbol{x}) f(\boldsymbol{x}) k(\boldsymbol{x}, \boldsymbol{x}') f(\boldsymbol{x}') p(\boldsymbol{x}') \,\mathrm{d}\mu(\boldsymbol{x}) \,\mathrm{d}\mu(\boldsymbol{x}') = \langle Af, T_{k,\mu} Af\rangle_{L_2(\mu)} \,.$$

Since $T_{k,\mu}$ and $T_{k,\nu}$ are compact, self-adjoint, and positive, we can use the Courant-Fischer minmax principle for operators (see e.g. Bell, 2014) to obtain

$$\begin{aligned}
\lambda_i(T_{k,\nu}) &= \max_{\substack{V \subseteq L_2(\nu) \text{ subspace} \\ \dim V = i}} \min_{f \in V \setminus \{0\}} \frac{\langle f, T_{k,\nu} f\rangle_{L_2(\nu)}}{\langle f, f\rangle_{L_2(\nu)}} \\
&\geq c \max_{\substack{V \subseteq L_2(\nu) \text{ subspace} \\ \dim V = i}} \min_{f \in V \setminus \{0\}} \frac{\langle Af, T_{k,\mu} Af\rangle_{L_2(\mu)}}{\langle Af, Af\rangle_{L_2(\mu)}} \\
&= c \max_{\substack{\tilde{V} \subseteq L_2(\nu) \text{ subspace} \\ \dim \tilde{V} = i}} \min_{g \in \tilde{V} \setminus \{0\}} \frac{\langle g, T_{k,\mu} g\rangle_{L_2(\mu)}}{\langle g, g\rangle_{L_2(\mu)}} \\
&= c\lambda_i(T_{k,\mu}) \,.
\end{aligned}$$

Here, we have used that since A is bijective, the subspaces AV for $\dim(V) = i$ are exactly the i-dimensional subspaces of $L_2(\mu)$. Our calculation above shows that $\lambda_i(T_{k,\mu}) \leq O(\lambda_i(T_{k,\nu}))$. Since $\mathrm{d}\mu = \frac{1}{p}\,\mathrm{d}\nu$ with the lower and upper bounded density $1/p$, we can reverse the roles of ν and μ to also obtain $\lambda_i(T_{k,\nu}) \leq O(\lambda_i(T_{k,\mu}))$, which proves the claim. □

Lemma F.10 (Integral operators of equivalent kernels have equivalent eigenvalues)**.** *Let $k, \tilde{k} : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ be bounded kernels with RKHSs $\mathcal{H}$ and $\tilde{\mathcal{H}}$ satisfying $\mathcal{H} = \tilde{\mathcal{H}}$ as sets. Moreover, let $C \geq 1$ be a constant such that the corresponding RKHS-norms are C-equivalent and let ν be a finite measure on $\mathcal{X}$. If there exist constants $q > 0$ and $c > 0$ with*

$$\lambda_i(T_{k,\nu}) \leq ci^{-q} \,, \qquad i \geq 1,$$

then we also have

$$\lambda_i(T_{\tilde{k},\nu}) \leq c \cdot C^2 \cdot K_q \cdot i^{-q} \,, \qquad i \geq 1,$$

where $K_q > 0$ is a constant only depending on q.

Proof. We follow the ideas outlined in (Steinwart, 2017, Section 3). To this end, let $I_{k,\nu} : \mathcal{H} \to L_2(\nu)$ be the embedding $h \mapsto [h]_\sim$, which is defined and compact since k is bounded and ν is finite, see e.g. (Steinwart and Scovel, 2012, Lemma 2.3). We write $S_{k,\nu} := I^*_{k,\nu}$ for its adjoint, which in turn gives $I_{k,\nu} = S^*_{k,\nu}$. Then (Steinwart and Scovel, 2012, Lemma 2.2) shows $T_{k,\nu} = S^*_{k,\nu} \circ S_{k,\nu}$. We denote the i-th (dyadic) entropy number of $I_{k,\nu}$ by $\varepsilon_i(I_{k,\nu})$, see e.g. Carl and Stephani (1990), (Edmunds and Triebel, 1996, Chapter 1.3.1), or (Steinwart and Christmann, 2008, Chapter 6) for

a definition[3]. Moreover, we denote the i-approximation and singular numbers of $I_{k,\nu}$ by $a_i(I_{k,\nu})$, respectively $s_i(I_{k,\nu})$. Since $I_{k,\nu}$ compactly acts between Hilbert spaces, we then have $a_i(I_{k,\nu}) = s_i(I_{k,\nu})$, see e.g. (Pietsch, 1987, Chapter 2.11). This implies

$$\lambda_i(T_{k,\nu}) = a_i^2(I_{k,\nu}) \tag{F.8}$$

for all $i \geq 1$ by the very definition of singular numbers. Finally, analogous definitions and considerations are made for the kernel $\tilde{k}$. From $C^{-1}\|\cdot\|_{\mathcal{H}} \leq \|\cdot\|_{\tilde{\mathcal{H}}} \leq C\|\cdot\|_{\mathcal{H}}$ we then conclude that

$$C^{-1}\varepsilon_i(I_{k,\nu}) \leq \varepsilon_i(I_{\tilde{k},\nu}) \leq C\varepsilon_i(I_{k,\nu})\,, \qquad i \geq 1, \tag{F.9}$$

by the multiplicativity of entropy numbers, see e.g. (Edmunds and Triebel, 1996, Chapter 1.3.1).

Now, (F.8) and our eigenvalue assumption yield

$$a_i(I_{k,\nu}) \leq \sqrt{c}\cdot i^{-q/2}\,, \qquad i \geq 1\,,$$

and Carl's inequality, see e.g. (Carl and Stephani, 1990, Theorem 3.1.1) then gives

$$\varepsilon_i(I_{k,\nu}) \leq \sqrt{c}\cdot \tilde{K}_q \cdot i^{-q/2}\,, \qquad i \geq 1\,,$$

where $\tilde{K}_q$ is a constant only depending on q. By (Carl and Stephani, 1990, Inequality (3.0.9)) and (F.9) we then obtain

$$a_i(I_{\tilde{k},\nu}) \leq 2\varepsilon_i(I_{\tilde{k},\nu}) \leq 2C\varepsilon_i(I_{k,\nu}) \leq 2C\sqrt{c}\cdot \tilde{K}_q \cdot i^{-q/2}\,, \qquad i \geq 1\,.$$

Another application of (F.8) then yields the assertion for $K_q := 4\tilde{K}_q^2$. □

F.3 Kernel matrix eigenvalue bounds

For upper bounds on the eigenvalues of kernel matrices, we use the following result:

Proposition F.11 (Kernel matrix eigenvalue upper bound in expectation)**.** *For $m \geq 1$, we have*

$$\mathbb{E}_{\boldsymbol{X}} \sum_{i=m}^{n} \lambda_i(k(\boldsymbol{X},\boldsymbol{X})/n) \leq \sum_{i=m}^{\infty} \lambda_i(T_k)\,. \tag{F.10}$$

Proof. Theorem 7.29 in Steinwart and Christmann (2008) shows that

$$\mathbb{E}_{D\sim\mu^n} \sum_{i=m}^{\infty} \lambda_i(T_{k,D}) \leq \sum_{i=m}^{\infty} \lambda_i(T_{k,\mu})\,, \tag{F.11}$$

where $T_{k,\mu} : L_2(\mu) \to L_2(\mu), f \mapsto \int k(x,\cdot)f(x)\,\mathrm{d}\mu(x)$ is the integral operator corresponding to the measure μ and $T_{k,D}$ is the corresponding discrete version thereof. We set $\mu := P_X$ and need to show that $k(\boldsymbol{X},\boldsymbol{X})/n$ has the same eigenvalues as $T_{k,D}$ if D and $\mathcal{X}$ contain the same data points $\boldsymbol{x}_1,\ldots,\boldsymbol{x}_n$. Consider a fixed D. Then, we can write $T_{k,D}(f) = n^{-1}ABf$, where

$$A : \mathbb{R}^n \to L_2(D), \boldsymbol{v} \mapsto \sum_{i=1}^{n} v_i k(\boldsymbol{x}_i,\cdot)$$
$$B : L_2(D) \to \mathbb{R}^n, f \mapsto (f(\boldsymbol{x}_1),\ldots,f(\boldsymbol{x}_n))^\top\,.$$

Then, $k(\boldsymbol{X},\boldsymbol{X})/n$ is the matrix representation of $n^{-1}BA$ with respect to the standard basis of $\mathbb{R}^n$. But AB and BA have the same non-zero eigenvalues, which means that

$$\sum_{i=m}^{n} \lambda_i(k(\boldsymbol{X},\boldsymbol{X})/n) = \sum_{i=m}^{\infty} \lambda_i(T_{k,D})\,,$$

from which the claim follows. □

[3] Usually, dyadic entropy numbers are denoted by $e_i(\cdot)$, but since this symbol is already used for eigenfunctions, we use $\varepsilon_i(\cdot)$ instead.

To obtain a lower bound, we want to leverage the lower bound by Buchholz (2022) for a certain radial basis function kernel with data generated from an open subset of $\mathbb{R}^d$. However, we want to consider different kernels and distributions on the whole sphere. The following theorem bridges the gap by going to subsets of the data on a sphere cap, projecting them to $\mathbb{R}^d$, and using the kernel equivalence results from Appendix F.2:

Theorem F.12 (Kernel matrix eigenvalue lower bound for Sobolev kernels on the sphere)**.** *Let k be a kernel on $\mathbb{S}^d$ such that its RKHS $\mathcal{H}_k$ is equivalent to a Sobolev space $H^s(\mathbb{S}^d)$ with smoothness $s > d/2$. Moreover, let P_X be a probability distribution on $\mathbb{S}^d$ with lower and upper bounded density. Let the rows of $\boldsymbol{X} \in \mathbb{R}^{n\times d}$ are drawn independently from P_X. Then, for $\varepsilon \in (0, 1/20)$, there exists a constant $c > 0$ and $n_0 \in \mathbb{N}$ such that for all $n \geq n_0$,*

$$\lambda_m(k(\boldsymbol{X}, \boldsymbol{X})/n) \geq cn^{-2s/d}$$

holds with probability $\geq 4/5$ for all $m \in \mathbb{N}$ with $1 \leq m \leq (1 - 11\varepsilon)n$.

Proof. We can choose a suitably large sphere cap T such that $P_X(T) \geq 1-\varepsilon$. Define the conditional distribution $P_T(\cdot) \coloneqq P_X(\cdot|T)$. Out of the points $\boldsymbol{X} = (\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n)$, we can consider the submatrix $\boldsymbol{X}_T = (\boldsymbol{x}_{i_1}, \ldots, \boldsymbol{x}_{i_N})^\top$ of the points lying in T. Conditioned on N, these points are i.i.d. samples from P_T. Moreover, by applying Markov's inequality to a Bernoulli distribution, we obtain $N \geq (1-10\varepsilon)n$ with probability $\geq 9/10$. We fix a value of $N \geq (1-10\varepsilon)n$ in the following and condition on it.

We denote the centered unit ball in $\mathbb{R}^d$ by $B_1(\mathbb{R}^d)$. Using a construction as in Lemma E.1, we can transport k and P_T from T to the unit ball $B_1(\mathbb{R}^d)$ using a rescaled stereographic projection feature map ϕ, such that we obtain a kernel k_ϕ and a distribution $P_\phi = (P_T)_\phi$ on $B_1(\mathbb{R}^d)$ that generate the same distribution of kernel matrices as k with P_T, and such that $\mathcal{H}_{k_\phi} \cong H^s(B_1(\mathbb{R}^d))$. The rows of $\boldsymbol{X}_\phi \coloneqq \phi(\boldsymbol{X}_T)$ are i.i.d. samples from P_ϕ. Moreover, we know that P_ϕ has an lower and upper bounded density w.r.t. the Lebesgue measure on $B_1(\mathbb{R}^d)$.

In order to apply the results from Buchholz (2022), we define a translation-invariant reference kernel on $\mathbb{R}^d$ through the Fourier transform

$$\hat{k}_{\mathrm{ref}}(\xi) = (1 + |\xi|^2)^{-2s} ,$$

see Eq. (3) in Buchholz (2022). The RKHS of k_{ref} on $\mathbb{R}^d$ is equivalent to the Sobolev space $H^s(\mathbb{R}^d)$. Therefore, the RKHS of $k_{\mathrm{ref}}|_{B_1(\mathbb{R}^d),B_1(\mathbb{R}^d)}$ is $H^s(B_1(\mathbb{R}^d))$, cf. the remarks in Appendix B.1 and Lemma F.7.

Now, let $1 \leq m \leq (1 - 11\varepsilon)n$, which implies

$$1 \leq m \leq (1 - 11\varepsilon)n \leq (1 - \varepsilon)(1 - 10\varepsilon)n \leq (1 - \varepsilon)N .$$

We apply Theorem 12 by Buchholz (2022) with bandwidth $\gamma = 1$ and $\alpha = 2s$ to λ_m and obtain with probability at least $1 - 2/N$:

$$\begin{aligned}\lambda_m(k_{\mathrm{ref}}(\boldsymbol{X}_\phi, \boldsymbol{X}_\phi))^{-1} &\leq c_3 \left(\frac{N^{2(\alpha-d)/d}}{(N - m)^{(\alpha-d)/d}} + 1 \right) \leq c_3 \left(\frac{N^{2(\alpha-d)/d}}{(\varepsilon N)^{(\alpha-d)/d}} + 1 \right) \\ &\leq c_4(n^{\alpha/d-1} + 1)\end{aligned}$$

as long as N is large enough such that $(1 - \varepsilon)N < N - 32\ln(N)$, which is the case if n is large enough. Here, the constant c_3 from Buchholz (2022) does not depend on N or m, but only on α, d, and the upper and lower bounds on the density, which in our case depend on ε through the choice of T. Since $\alpha = 2s > d$, we have $n^{\alpha/d-1} > 1$ and therefore

$$\lambda_m(k_{\mathrm{ref}}(\boldsymbol{X}_\phi, \boldsymbol{X}_\phi)/n) \geq c_5 n^{-\alpha/d} = c_5 n^{-2s/d} .$$

Now, we want to translate this to the kernel k. Since the RKHSs of k_ϕ and k_{ref} on $B_1(\mathbb{R}^d)$ are both equivalent to $H^s(B_1(\mathbb{R}^d))$, the kernels themselves are C-equivalent for some constant $C \geq 1$ as defined in Definition F.4. Therefore, Proposition F.8 shows that the corresponding kernel matrices are C^2-equivalent, which implies

$$\lambda_m(k_{*,\phi}(\boldsymbol{X}_\phi, \boldsymbol{X}_\phi)/n) \geq c_5 C^{-2} n^{-2s/d} .$$

By Cauchy's interlacing theorem, we therefore have

$$\lambda_m(k_*(\boldsymbol{X}, \boldsymbol{X})/n) \geq \lambda_m(k_*(\boldsymbol{X}_T, \boldsymbol{X}_T)/n) = \lambda_m(k_{*,\phi}(\boldsymbol{X}_\phi, \boldsymbol{X}_\phi)/n) \geq c_5 C^{-2} n^{-2s/d} .$$

Denoting the event where $\lambda_m(k_*(\boldsymbol{X}, \boldsymbol{X})/n) \geq c_5 C^{-2} n^{-2s/d}$ by A, we thus have

$$\begin{aligned}
P(A) &= P(A|N \geq (1-10\varepsilon)n)P(N \geq (1-10\varepsilon)n) \geq \frac{9}{10} P(A|N \geq (1-10\varepsilon)n) \\
&= \frac{9}{10} \sum_{\hat{N}=\lceil (1-10\varepsilon)n \rceil}^{n} P(N = \hat{N}|N \geq (1-10\varepsilon)n) P(A|N = \hat{N}) \\
&\geq \frac{9}{10} \sum_{\hat{N}=\lceil (1-10\varepsilon)n \rceil}^{n} P(N = \hat{N}|N \geq (1-10\varepsilon)n)(1 - 2/N) \\
&\geq \frac{9}{10} \left(1 - \frac{2}{(1-10\varepsilon)n}\right) \sum_{\hat{N}=\lceil (1-10\varepsilon)n \rceil}^{n} P(N = \hat{N}|N \geq (1-10\varepsilon)n) \\
&= \frac{9}{10} \left(1 - \frac{2}{(1-10\varepsilon)n}\right) \geq \frac{4}{5} ,
\end{aligned}$$

where the last step holds for sufficiently large n. □

F.4 Spectral lower bound for dot-product kernels on the sphere

An application of the spectral generalization bound in Proposition 5 requires a lower bound on eigenvalues of the kernel matrix $k_*(\boldsymbol{X}, \boldsymbol{X})$. To achieve this, we need to understand the properties of the convolution kernel k_*. Since the eigenvalues of T_{k_*,P_X} are the squared eigenvalues of T_{k,P_X}, one might hope that if $\mathcal{H}_k$ is equivalent to a Sobolev space H^s, then $\mathcal{H}_{k_*}$ is equivalent to a Sobolev space H^{2s}. Unfortunately, this is not the case in general, as $\mathcal{H}_{k_*}$ might be a smaller space that involves additional boundary conditions (Schaback, 2018). However, perhaps since the sphere is a manifold without boundary, the desired characterization of $\mathcal{H}_{k_*}$ holds for dot-product kernels on the sphere:

Lemma F.13 (RKHS of convolution kernels)**.** *Let k be a dot-product kernel on $\mathbb{S}^d$ such that its RKHS $\mathcal{H}_k$ is equivalent to a Sobolev space $H^s(\mathbb{S}^d)$ with smoothness $s > d/2$, and let P_X be a distribution on $\mathbb{S}^d$ with lower and upper bounded density. Then, the RKHS $\mathcal{H}_{k_*}$ of the kernel*

$$k_* : \mathbb{S}^d \times \mathbb{S}^d \to \mathbb{R}, k_*(\boldsymbol{x}, \boldsymbol{x}') \coloneqq \int k(\boldsymbol{x}, \boldsymbol{x}'') k(\boldsymbol{x}'', \boldsymbol{x}') \, \mathrm{d}P_X(\boldsymbol{x}'')$$

is equivalent to the Sobolev space $H^{2s}(\mathbb{S}^d)$.

Proof. Define

$$k_{*,\mathrm{unif}}(\boldsymbol{x}, \boldsymbol{x}') = \int k(\boldsymbol{x}, \boldsymbol{x}'') k(\boldsymbol{x}'', \boldsymbol{x}') \, \mathrm{d}\mathcal{U}(\mathbb{S}^d)(\boldsymbol{x}'') .$$

For the corresponding integral operator, we have

$$T_{k_{*,\mathrm{unif}},\mathcal{U}(\mathbb{S}^d)} = T^2_{k,\mathcal{U}(\mathbb{S}^d)} .$$

This means that the corresponding eigenvalues are the squares of the eigenvalues of the corresponding integral operator of k. Especially, we obtain the Mercer representations

$$\begin{aligned}
k(\boldsymbol{x}, \boldsymbol{x}') &= \sum_{l=0}^{\infty} \mu_l \sum_{i=1}^{N_{l,d}} Y_{l,i}(\boldsymbol{x}) Y_{l,i}(\boldsymbol{x}') , \\
k_{*,\mathrm{unif}}(\boldsymbol{x}, \boldsymbol{x}') &= \sum_{l=0}^{\infty} \mu_l^2 \sum_{i=1}^{N_{l,d}} Y_{l,i}(\boldsymbol{x}) Y_{l,i}(\boldsymbol{x}') ,
\end{aligned}$$

where Lemma B.1 yields $\mu_l = \Theta((l+1)^{-2s})$, hence $\mu_l^2 = \Theta((l+1)^{-4s})$ and hence $\mathcal{H}_{k_{*,\mathrm{unif}}} \cong H^{2s}(\mathbb{S}^d)$.

Next, we show the equality of the ranges of the integral operators:

$$R(T_{k,\mathcal{U}(\mathbb{S}^d)}) = R(T_{k,P_X}) \,.$$

Let p_X be a density of P_X w.r.t. the uniform distribution $\mathcal{U}(\mathbb{S}^d)$. If $f \in R(T_{k,\mathcal{U}(\mathbb{S}^d)})$, there exists $g \in L_2(\mathcal{U}(\mathbb{S}^d))$ with $f = T_{k,\mathcal{U}(\mathbb{S}^d)} g$. But then, since p_X is lower bounded, we have $g/p_X \in L_2(P_X)$ and therefore

$$f = T_{k,P_X}(g/p_X) \in R(T_{k,P_X}) \,.$$

An analogous argument shows that $R(T_{k,P_X}) \subseteq R(T_{k,\mathcal{U}(\mathbb{S}^d)})$ since p_X is upper bounded.

The equality of the ranges yields for the RKHSs (as sets)

$$\mathcal{H}_{k_{*,\mathrm{unif}}} = R(T_{k,\mathcal{U}(\mathbb{S}^d)}) = R(T_{k,P_X}) = \mathcal{H}_{k_*} \,,$$

Applying Lemma F.7 twice then shows $\mathcal{H}_{k_*} \cong H^{2s}(\mathbb{S}^d)$. □

Theorem 6 (Inconsistency for Sobolev dot-product kernels on the sphere). *Let k be a dot-product kernel on $\mathbb{S}^d$, i.e., a kernel of the form $k(\boldsymbol{x}, \boldsymbol{x}') = \kappa(\langle \boldsymbol{x}, \boldsymbol{x}' \rangle)$, such that its RKHS $\mathcal{H}_k$ is equivalent to a Sobolev space $H^s(\mathbb{S}^d)$, $s > d/2$. Moreover, let P be a distribution on $\mathbb{S}^d \times \mathbb{R}$ such that P_X has a lower and upper bounded density w.r.t. the uniform distribution $\mathcal{U}(\mathbb{S}^d)$, and such that $\mathrm{Var}(y|\boldsymbol{x}) \geq \sigma^2 > 0$ for P_X-almost all $\boldsymbol{x} \in \mathbb{S}^d$. Then, for every $C > 0$, there exists $c > 0$ independent of σ^2 such that for all $n \geq 1$, $t \in (C^{-1} n^{2s/d}, \infty]$, and $\rho \in [0, C n^{-2s/d})$, the expected excess risk satisfies*

$$\mathbb{E}_D R_P(f_{t,\rho}) - R_P^* \geq c\sigma^2 > 0 \,.$$

Proof. **Step 0: Preparation.** Since the Sobelev space $H^{2s}(\mathbb{S}^d)$ is dense in the space of continuous functions $\mathbb{S}^d \to \mathbb{R}$, the kernel k is universal. Applying (Steinwart and Christmann, 2008, Corollary 5.29 and Corollary 5.34) for the least squares loss thus shows that k is strictly positive definite. If we have mutually distinct $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n$, the corresponding Gram matrix $k((\boldsymbol{x}_i, \boldsymbol{x}_j))_{i,j=1}^n$ is therefore invertible. Now, our assumptions on P guarantee that $\boldsymbol{X}$ consists almost surely of mutually distinct observations, and therefore $k(\boldsymbol{X}, \boldsymbol{X})$ is almost surely invertible.

By Proposition 5, we know that

$$\begin{aligned}
\mathbb{E}_D \mathcal{R}_P(f_{\boldsymbol{X},\boldsymbol{y},t,\rho}) - \mathcal{R}_P^* &\geq \frac{\sigma^2}{n} \sum_{i=1}^n \mathbb{E}_{\boldsymbol{X}} \frac{\lambda_i(k_*(\boldsymbol{X},\boldsymbol{X})/n) \left(1 - e^{-2t(\lambda_i(k(\boldsymbol{X},\boldsymbol{X})/n)+\rho)}\right)^2}{(\lambda_i(k(\boldsymbol{X},\boldsymbol{X})/n) + \rho)^2} \\
&\geq \frac{\sigma^2}{n} \sum_{i=1}^n \mathbb{E}_{\boldsymbol{X}} \frac{\lambda_i(k_*(\boldsymbol{X},\boldsymbol{X})/n) \left(1 - e^{-2C^{-1} n^{2s/d}(\lambda_i(k(\boldsymbol{X},\boldsymbol{X})/n)+0)}\right)^2}{(\lambda_i(k(\boldsymbol{X},\boldsymbol{X})/n) + C n^{-2s/d})^2} \\
&\geq c_n \sigma^2
\end{aligned}$$

for a suitable constant $c_n > 0$ depending on n but not on σ^2, t, ρ, since the kernel matrix eigenvalues are nonzero almost surely. It is therefore sufficient to show the desired statement (with c independent of n, σ^2, t, ρ) for sufficiently large n.

In the following, we assume $n \geq 40$ and set $\varepsilon := 1/100$.

Step 1: Eigenvalue decay for the integral operator. From Lemma B.1, we know that

$$\lambda_i(T_{k,\mathcal{U}(\mathbb{S}^d)}) = \Theta(i^{-2s/d}) \,.$$

Therefore, by Lemma F.9, we know that

$$\lambda_i(T_{k,P_X}) = \Theta(i^{-2s/d}) \,.$$

Step 2: Eigenvalue upper bound. Next, we want to upper-bound suitable eigenvalues of the form $\lambda_i(k(\boldsymbol{X},\boldsymbol{X})/n)$ using Proposition F.11. Using Step 1, we derive

$$\sum_{i=m}^{\infty} \lambda_i(T_{k,P_X}) \leq C_1 \sum_{i=m}^{\infty} i^{-2s/d} \leq C_2 \int_m^{\infty} x^{-2s/d} \,\mathrm{d}x = C_3 m^{1-2s/d}$$

with constants independent of $m \geq 1$. For sufficiently large n, we can choose $m \in \mathbb{N}_{\geq 1}$ such that $\varepsilon n \leq m \leq 2\varepsilon n$. Then, Proposition F.11 yields

$$\mathbb{E}_{\boldsymbol{X}} \sum_{i=m}^{n} \lambda_i(k(\boldsymbol{X}, \boldsymbol{X})/n) \leq \sum_{i=m}^{\infty} \lambda_i(T_k) \leq C_3 m^{1-2s/d} \leq C_4 n^{1-2s/d} .$$

Since $\mathbb{E}_{\boldsymbol{X}} \lambda_i(k(\boldsymbol{X}, \boldsymbol{X})/n)$ is decreasing with i, we have for $i \geq 4\varepsilon n \geq 2m$:

$$\mathbb{E}_{\boldsymbol{X}} \lambda_i(k(\boldsymbol{X}, \boldsymbol{X})/n) \leq C_4 n^{1-2s/d}/m \leq C_5 n^{-2s/d} \leq C_6 \lambda_i(T_{k,P_X}) .$$

Step 3: Eigenvalue lower bounds. From Lemma F.13, we know that $\mathcal{H}_{k_*} \cong H^{2s}(\mathbb{S}^d)$. Therefore, we can apply Lemma F.13 to both k and k_* and obtain for sufficiently large n and suitable constants $c_1, c_2 > 0$ that

$$\lambda_i(k(\boldsymbol{X}, \boldsymbol{X})/n) \geq c_1 n^{-2s/d}$$
$$\lambda_i(k_*(\boldsymbol{X}, \boldsymbol{X})/n) \geq c_2 n^{-4s/d}$$

individually hold with probability $\geq 4/5$ for all $i \in \mathbb{N}$ with $1 \leq i \leq (1 - 11\varepsilon)n$. By the union bound, both bounds hold at the same time with probability $\geq 3/5$.

Step 4: Final result. Now, using the value of m from Step 2, consider an index i with $2m \leq i \leq (1 - 11\varepsilon)n$. Since $2m \leq 4\varepsilon n$ and $\varepsilon = 1/100$, there are at least $n/2$ such indices. By combining Step 3 and Step 1, we have

$$\lambda_i(k(\boldsymbol{X}, \boldsymbol{X})/n) \geq c_3 \lambda_i(T_{k,P_X})$$
$$\lambda_i(k_*(\boldsymbol{X}, \boldsymbol{X})/n) \geq c_4 \lambda_i(T_{k,P_X})^2$$

with probability $\geq 3/5$. By applying Markov's inequality to Step 2, we obtain

$$\lambda_i(k(\boldsymbol{X}, \boldsymbol{X})/n) \leq 10 C_6 \lambda_i(T_{k,P_X})$$

with probability $\geq 9/10$. Therefore, by the union bound, all three inequalities hold simultaneously with probability $\geq 1/2$. Moreover, for $q = 2s/d$, we have $\lambda_i(T_{k,P_X}) \geq c_5 i^{-q}$ by Step 1. We can thus apply the first lower bound from Theorem F.2 to obtain

$$\begin{aligned}
\mathbb{E}_D R_P(f_{\boldsymbol{X},\boldsymbol{y},t,\rho}) - R_P^* &\geq c'\sigma^2 \frac{1}{1 + (\rho + t^{-1})n^{2s/d}} \cdot \frac{|\mathcal{I}(n)|}{n} \\
&\geq c'\sigma^2 \frac{1}{1 + (Cn^{-2s/d} + Cn^{-2s/d})n^{2s/d}} \cdot \frac{n/2}{n} \\
&= \frac{c'}{2 + 2C}\sigma^2 .
\end{aligned}$$

□

G Proof of Theorem 8

Here we denote the solution of kernel ridge regression on D with the kernel function k and regularization parameter $\rho > 0$ as

$$\hat{f}_\rho^k(\boldsymbol{x}) = k(\boldsymbol{x}, \boldsymbol{X}) \left(k(\boldsymbol{X}, \boldsymbol{X}) + \rho \boldsymbol{I}\right)^{-1} \boldsymbol{y},$$

and write $\hat{f}_0^k(\boldsymbol{x}) = k(\boldsymbol{x}, \boldsymbol{X}) k(\boldsymbol{X}, \boldsymbol{X})^+ \boldsymbol{y}$ for the minimum-norm interpolant in the RKHS of k.

While Theorem 1 states that overfitting kernel ridge regression using Sobolev kernels is always inconsistent as long as the derivatives remain bounded by the derivatives of the minimum-norm interpolant of the fixed kernel (Assumption (N)), here we show that consistency over a large class of distributions is achievable by designing a kernel sequence, which can have Sobolev RKHS, that consists of a smooth component for generalization and a spiky component for interpolation.

Recall that $\tilde{k}$ denotes any universal kernel function for the smooth component, and $\check{k}_\gamma$ denotes the kernel function of the spiky component with bandwidth γ. Then we define the *ρ-regularized spiky-smooth kernel with spike bandwidth γ* as

$$k_{\rho,\gamma}(\boldsymbol{x}, \boldsymbol{x}') = \tilde{k}(\boldsymbol{x}, \boldsymbol{x}') + \rho \cdot \check{k}_\gamma(\boldsymbol{x}, \boldsymbol{x}').$$

Let $B_t(\boldsymbol{x}) \coloneqq \{\boldsymbol{y} \in \mathbb{R}^d \mid |\boldsymbol{x} - \boldsymbol{y}| \leq t\}$ denote the Euclidean ball of radius $t \geq 0$ around $\boldsymbol{x} \in \mathbb{R}^d$.

(D2) There exists a constant $\beta_X > 0$ and a continuous function $\phi : [0, \infty) \to [0, 1]$ with $\phi(0) = 0$ such that $P_X(B_t(\boldsymbol{x})) \leq \phi(t) = O(t^{\beta_X})$ for all $\boldsymbol{x} \in \Omega$ and all $t \geq 0$.

The kernel $\check{k}_\gamma$ of the spiky component should fulfill the following weak assumption on its decay behaviour. For example, Laplace, Matérn, and Gaussian kernels all fulfill Assumption (SK).

(SK) There exists a function $\varepsilon : (0, \infty) \times [0, \infty) \to [0, 1]$ such that for any bandwidth $\gamma > 0$ and any $\delta > 0$ it holds that
  (i) $\varepsilon(\gamma, 0) = 1$,
  (ii) $\varepsilon(\gamma, \delta)$ is monotonically increasing in γ,
  (iii) For all $\boldsymbol{x}, \boldsymbol{y} \in \Omega$, if $|\boldsymbol{x} - \boldsymbol{y}| \geq \delta$ then $|\check{k}_\gamma(\boldsymbol{x}, \boldsymbol{y})| \leq \varepsilon(\gamma, \delta)$,
  (iv) For any rates $\beta_X, \beta_k > 0$ there exists a rate $\beta_\gamma > 0$ such that, if $\delta_n = \Omega(n^{-\beta_X})$ and $\gamma_n = O(n^{-\beta_\gamma})$, then $\varepsilon(\gamma_n, \delta_n) = O(n^{-\beta_k})$.

Theorem G.1 (Consistency of spiky-smooth ridgeless kernel regression). *Assume that the training set D consists of n i.i.d. pairs $(\boldsymbol{x}, y) \sim P$ such that the marginal P_X fulfills (D2) and $\mathbb{E}y^2 < \infty$. Let the kernel components satisfy:*

- *$\tilde{k}$ denotes an arbitrary universal kernel, and $\rho_n \to 0$ and $n\rho_n^4 \to \infty$.*
- *$\check{k}_{\gamma_n}$ denotes a kernel function that fulfills Assumption (SK) with a sequence of positive bandwidths (γ_n) fulfilling $\gamma_n = O(\exp(-\beta n))$ for some arbitrary $\beta > 0$.*

Then the minimum-norm interpolant of the ρ_n-regularized spiky-smooth kernel sequence $k_n := k_{\rho_n, \gamma_n}$ is consistent for P.

Remark G.2 (Spike bandwidth scaling). Under stronger assumptions on ϕ and ε in assumptions (D2) and (SK), the spike bandwidths γ_n can be chosen to converge to 0 at a much slower rate. For example, if we choose $\check{k}_\gamma$ to be the Laplace kernel, choosing bandwidths $0 < \gamma_n \leq \frac{\delta}{\beta \ln n}$ yields, for separated points $|\boldsymbol{x} - \boldsymbol{y}| \geq \delta$,

$$\check{k}_{\gamma_n}(\boldsymbol{x}, \boldsymbol{y}) \leq \exp\left(-\frac{\delta}{\gamma_n}\right) \leq n^{-\beta}.$$

For probability measures with upper bounded Lebesgue density, we can choose $\delta_n = n^{-\frac{2+\alpha}{d}}$ and $\beta = \frac{9}{4} + \frac{\alpha}{2}$ for consistency or $\beta = \frac{11}{4} + \frac{\alpha}{2}$ for optimal convergence rates, for any fixed $\alpha > 0$, in the proof of Theorem 8. Hence the Laplace kernel only requires a slow bandwidth decay rate of $\gamma_n = \Omega\left(\frac{n^{-\frac{2+\alpha}{d}}}{\alpha \ln(n)}\right)$, where $\alpha > 0$ arbitrary. For the Gaussian kernel an analogous argument yields $\gamma_n = \Omega\left(\frac{n^{-\frac{4+2\alpha}{d}}}{\alpha \ln(n)}\right)$. The larger the dimension d, the slower the required bandwidth decay. ◀

Remark G.3 (Generalizations). If one does not care about continuous kernels, one could simply take a Dirac kernel as the spike and then obtain consistency for all atom-free P_X. However, we need a continuous kernel to be able to translate it to an activation function for the NTK. Beyond kernel regression, the spike component $\check{k}_\gamma$ does not even need to be a kernel, it just needs to fulfill Assumption (SK) or a similar decay criterion. Then one could still use the 'quasi minimum-norm estimator' $\boldsymbol{x} \mapsto (\tilde{k} + \rho_n \check{k}_{\gamma_n})(\boldsymbol{x}, \boldsymbol{X}) \cdot (\tilde{\boldsymbol{K}} + \rho_n \check{\boldsymbol{K}}_{\gamma_n})^+ \boldsymbol{y}$. ◀

Remark G.4 (Consistency with a single kernel function). Without resorting to kernel sequences as we do, there seems to be no rigorous proof showing that ridgeless kernel regression can be consistent in fixed dimension. In future work, can an analytical expression of such a kernel be found? According to the semi-rigorous results in Mallinar et al. (2022) a spectral decay like $\lambda_k = \Theta(k^{-1} \cdot \log^\alpha(k))$, $\alpha > 1$ could lead to such a kernel. ◀

Proof of Theorem G.1. Given any universal kernel, (Steinwart, 2001, Theorem 3.11 or Example 4.6) implies universal consistency of kernel ridge regression if $\rho_n \to 0$ and $n\rho_n^4 \to \infty$. Hence, for any $\varepsilon > 0$ it holds that

$$\lim_{n \to \infty} P^n \left(D \in (\mathbb{R}^d \times \mathbb{R})^n \mid R_P(\hat{f}^{\tilde{k}}_{\rho_n}) - R_P(f^*_P) = \mathbb{E}_{\boldsymbol{x}}(\hat{f}^{\tilde{k}}_{\rho_n}(\boldsymbol{x}) - f^*_P(\boldsymbol{x}))^2 \geq (\varepsilon/2)^2 \right) = 0.$$

Due to the triangle inequality in $L_2(P_X)$, we know

$$R_P(\hat{f}^{k_n}_0) - R_P(f^*_P) = \mathbb{E}_{\boldsymbol{x}}(\hat{f}^{k_n}_0(\boldsymbol{x}) - f^*_P(\boldsymbol{x}))^2$$

$$\leq \left(\left(\mathbb{E}_{\boldsymbol{x}} (\hat{f}_0^{k_n}(\boldsymbol{x}) - \hat{f}_{\rho_n}^{\tilde{k}}(\boldsymbol{x}))^2 \right)^{1/2} + \left(\mathbb{E}_{\boldsymbol{x}} (\hat{f}_{\rho_n}^{\tilde{k}}(\boldsymbol{x}) - f_P^*(\boldsymbol{x}))^2 \right)^{1/2} \right)^2 .$$

It is left to show that k_n fulfills

$$\lim_{n\to\infty} P^n \left(D \in (\mathbb{R}^d \times \mathbb{R})^n \mid \mathbb{E}_{\boldsymbol{x}} (\hat{f}_0^{k_n}(\boldsymbol{x}) - \hat{f}_{\rho_n}^{\tilde{k}}(\boldsymbol{x}))^2 \geq (\varepsilon/2)^2 \right) = 0.$$

For this purpose we decompose the above difference into the difference of $\check{\boldsymbol{K}}_{\gamma_n} \coloneqq \check{k}_{\gamma_n}(\boldsymbol{X}, \boldsymbol{X})$ and $\boldsymbol{I}_n$ and a remainder term depending on $\check{k}_{\gamma_n}$. We denote the 2-operator norm by $\|\cdot\|$ and the Euclidean norm in $\mathbb{R}^n$ by $|\cdot|$. For any $\boldsymbol{x} \in \mathbb{R}^d$ it holds that

$$\begin{aligned}
|\hat{f}_0^k(\boldsymbol{x}) - \hat{f}_{\rho_n}^{\tilde{k}}(\boldsymbol{x})| &\leq \left| (\tilde{k} + \rho_n \check{k}_{\gamma_n})(\boldsymbol{x}, \boldsymbol{X}) \cdot (\tilde{\boldsymbol{K}} + \rho_n \check{\boldsymbol{K}}_{\gamma_n})^{-1} \boldsymbol{y} - \tilde{k}(\boldsymbol{x}, \boldsymbol{X}) \cdot (\tilde{\boldsymbol{K}} + \rho_n \boldsymbol{I}_n)^{-1} \boldsymbol{y} \right| \\
&\leq \left| \tilde{k}(\boldsymbol{x}, \boldsymbol{X}) \left((\tilde{\boldsymbol{K}} + \rho_n \check{\boldsymbol{K}}_{\gamma_n})^{-1} - (\tilde{\boldsymbol{K}} + \rho_n \boldsymbol{I}_n)^{-1} \right) \boldsymbol{y} \right| \\
&\quad + \rho_n \cdot \left| \check{k}_{\gamma_n}(\boldsymbol{x}, \boldsymbol{X}) (\tilde{\boldsymbol{K}} + \rho_n \check{\boldsymbol{K}}_{\gamma_n})^{-1} \boldsymbol{y} \right| \\
&\leq \|\tilde{k}(\boldsymbol{x}, \boldsymbol{X})\| \cdot \left\| (\tilde{\boldsymbol{K}} + \rho_n \check{\boldsymbol{K}}_{\gamma_n})^{-1} - (\tilde{\boldsymbol{K}} + \rho_n \boldsymbol{I}_n)^{-1} \right\| \cdot |\boldsymbol{y}| \\
&\quad + \rho_n \cdot \|\check{k}_{\gamma_n}(\boldsymbol{x}, \boldsymbol{X})\| \cdot \|(\tilde{\boldsymbol{K}} + \rho_n \check{\boldsymbol{K}}_{\gamma_n})^{-1}\| \cdot |\boldsymbol{y}|.
\end{aligned}$$

Consequently we get

$$\mathbb{E}_{\boldsymbol{x}} (\hat{f}_0^k(\boldsymbol{x}) - \hat{f}_{\rho_n}^{\tilde{k}}(\boldsymbol{x}))^2 \leq 2\, \mathbb{E}_{\boldsymbol{x}} \|\tilde{k}(\boldsymbol{x}, \boldsymbol{X})\|^2 \cdot \left\| (\tilde{\boldsymbol{K}} + \rho_n \check{\boldsymbol{K}}_{\gamma_n})^{-1} - (\tilde{\boldsymbol{K}} + \rho_n \boldsymbol{I}_n)^{-1} \right\|^2 \cdot |\boldsymbol{y}|^2 \quad \text{(G.1)}$$

$$+\, 2\, \rho_n^2 \cdot \mathbb{E}_{\boldsymbol{x}} \|\check{k}_{\gamma_n}(\boldsymbol{x}, \boldsymbol{X})\|^2 \cdot \|(\tilde{\boldsymbol{K}} + \rho_n \check{\boldsymbol{K}}_{\gamma_n})^{-1}\|^2 \cdot |\boldsymbol{y}|^2. \quad \text{(G.2)}$$

We now bound the individual terms in Eq. (G.1) and (G.2). To this end, fix any $\alpha > 0$.

Bounding Eq. (G.1):

Since we assumed y_i i.i.d. and $\mathbb{E} y_1^2 < \infty$, the Markov inequality implies, with $b_n = \mathbb{E} y_1^2 \cdot n^\alpha$,

$$P(|\boldsymbol{y}|^2 \geq b_n n) \leq \frac{\mathbb{E} y_1^2}{b_n} = n^{-\alpha}.$$

Stated differently, with probability at least $1 - n^{-\alpha}$ it holds that $|\boldsymbol{y}|^2 \leq \mathbb{E} y_1^2 \cdot n^{1+\alpha}$.

In order to bound the spectrum of $\check{\boldsymbol{K}}_{\gamma_n}$, Lemma G.7 implies that there exists a positive sequence $\delta_\alpha(n) = n^{-\frac{2+\alpha}{\beta_X}}$ such that with probability at least $1 - O(n^{-\alpha})$ it holds that

$$\min_{i,j \in [n] : i \neq j} |\boldsymbol{x}_i - \boldsymbol{x}_j| \geq \delta_\alpha(n).$$

Since (γ_n) fulfills $\gamma_n = O(n^{-\beta_\gamma})$ for any $\beta_\gamma > 0$, by Assumption (SK) there exists a sequence $\varepsilon_n = o(\rho_n n^{-2-\frac{\alpha}{2}})$ such that $\varepsilon(\gamma_n, \delta_\alpha(n)) \leq \varepsilon_n$. Assumption (SK) further implies that whenever $\min_{i,j \in [n] : i \neq j} |\boldsymbol{x}_i - \boldsymbol{x}_j| \geq \delta_\alpha(n)$ it holds that $(\check{\boldsymbol{K}}_{\gamma_n})_{ii} = 1$ and $0 \leq (\check{\boldsymbol{K}}_{\gamma_n})_{ij} \leq \varepsilon(\gamma_n, \delta_\alpha(n)) \leq \varepsilon_n$ for $i \neq j$. Then Gershgorin's theorem (Gerschgorin, 1931) implies that for all eigenvalues of $\check{\boldsymbol{K}}_{\gamma_n}$

$$|\lambda_i(\check{\boldsymbol{K}}_{\gamma_n}) - 1| \leq (n-1)\varepsilon_n \text{ for all } i \in [n].$$

This in turn implies

$$\|\check{\boldsymbol{K}}_{\gamma_n} - \boldsymbol{I}_n\| \leq (n-1)\varepsilon_n, \qquad \lambda_{\max}(\check{\boldsymbol{K}}_{\gamma_n}) \leq 1 + (n-1)\varepsilon_n, \qquad \lambda_{\min}(\check{\boldsymbol{K}}_{\gamma_n}) \geq 1 - (n-1)\varepsilon_n.$$

Using $\|(\tilde{\boldsymbol{K}} + \rho_n \boldsymbol{I}_n)^{-1}\| \leq \frac{1}{\lambda_{\min}(\tilde{\boldsymbol{K}}) + \rho_n} \leq \rho_n^{-1}$ and $\|\check{\boldsymbol{K}}_{\gamma_n} - \boldsymbol{I}_n\| \leq (n-1)\varepsilon_n$, Lemma G.8 implies

$$\begin{aligned}
\left\| (\tilde{\boldsymbol{K}} + \rho_n \check{\boldsymbol{K}}_{\gamma_n})^{-1} - (\tilde{\boldsymbol{K}} + \rho_n \boldsymbol{I}_n)^{-1} \right\| &\leq \frac{\|(\tilde{\boldsymbol{K}} + \rho_n \boldsymbol{I}_n)^{-1}\|^2 \cdot \rho_n \|\check{\boldsymbol{K}}_{\gamma_n} - \boldsymbol{I}_n\|}{1 - \|(\tilde{\boldsymbol{K}} + \rho_n \boldsymbol{I}_n)^{-1}\| \cdot \rho_n \|\check{\boldsymbol{K}}_{\gamma_n} - \boldsymbol{I}_n\|} \\
&\leq \frac{\rho_n^{-1}(n-1)\varepsilon_n}{1 - (n-1)\varepsilon_n}.
\end{aligned}$$

Using $|\tilde{k}(\boldsymbol{x}, \boldsymbol{X}_i)| \leq 1$ for all $i \in [n]$ yields the naive bound $\|\tilde{k}(\boldsymbol{x}, \boldsymbol{X})\|^2 \leq n$.

Combining all terms in Eq. (G.1) yields its convergence to 0 as the product satifies the rate $O(n^{4+\alpha}\rho_n^{-2}\varepsilon_n^2) = o(1)$ with probability at least $1 - 2n^{-\alpha}$.

Bounding Eq. (G.2):

The analysis below is restricted to the event of probability at least $1 - 2n^{-\alpha}$, on which the bound on Eq. (G.1) holds.

Since $(n-1)\varepsilon_n \to 0$, for any $C > 1$ it holds for n large enough,

$$\rho_n \cdot \|(\tilde{\boldsymbol{K}} + \rho_n \check{\boldsymbol{K}}_{\gamma_n})^{-1}\| \leq \frac{\rho_n}{\lambda_{\min}(\tilde{\boldsymbol{K}}) + \rho_n(1-(n-1)\varepsilon_n)} \leq \frac{1}{(1-(n-1)\varepsilon_n)} \leq C.$$

Finally we show $\sup_{\boldsymbol{x}' \in \mathbb{R}^d} \mathbb{E}_{\boldsymbol{x}} \check{k}_{\gamma_n}(\boldsymbol{x}, \boldsymbol{x}')^2 \leq 2n^{-(2+\alpha)}$ for n large enough.

Fix an arbitrary $\boldsymbol{x}' \in \mathbb{R}^d$. Then by construction of $\delta_\alpha(n)$ and ε_n it holds that

$$\begin{aligned}
\mathbb{E}_{\boldsymbol{x}} \check{k}_{\gamma_n}(\boldsymbol{x}, \boldsymbol{x}')^2 &\leq 1 \cdot P_X(\{\boldsymbol{x} \in \mathbb{R}^d : \ \check{k}_{\gamma_n}(\boldsymbol{x}, \boldsymbol{x}')^2 \geq \varepsilon_n^2\}) + \varepsilon_n^2 \\
&\leq P_X(\{\boldsymbol{x} \in \mathbb{R}^d : \ |\boldsymbol{x} - \boldsymbol{x}'| < \delta_\alpha(n)\}) + \varepsilon_n^2 \\
&\leq \phi(\delta_\alpha(n)) + \varepsilon_n^2 \leq n^{-(2+\alpha)} + \varepsilon_n^2.
\end{aligned}$$

Since $\varepsilon_n^2 = o(\rho_n^2 n^{-4-\alpha})$, we get $\mathbb{E}_{\boldsymbol{x}} \check{k}_{\gamma_n}(\boldsymbol{x}, \boldsymbol{x}')^2 \leq 2n^{-(2+\alpha)}$ for n large enough.

Combining all terms in Eq. (G.2) yields its convergence to 0 with the rate $O(n^{-(2+\alpha)} \cdot 1 \cdot n^{1+\alpha}) = O(n^{-1})$ with probability at least $1 - 2n^{-\alpha}$, which concludes the proof. □

The following theorem shows that the minimum-norm interpolants of the spiky-smooth kernel sequence can achieve optimal convergence rates for Sobolev target functions, as long as ρ_n is properly chosen. We therefore introduce Assumption (D3), which resembles Assumption (D1) but allows more general target functions $f^* \in H^{s^*}(\Omega) \backslash \{0\}$, $s^* > 0$, that may lie outside of the RKHS.

(D3) Let $\Omega = \mathbb{S}^d$ or let $\Omega \subseteq \mathbb{R}^d$ be a bounded open Lipschitz domain. Let P_X be a distribution on Ω with lower- and upper-bounded Lebesgue density. Consider i.i.d. data sets $D = \{(\boldsymbol{x}_1, y_1), \ldots, (\boldsymbol{x}_n, y_n)\} \subseteq \Omega \times \mathbb{R}$, where $\boldsymbol{x}_i \sim P_X$, $f^*(\boldsymbol{x}) = \mathbb{E}[y|\boldsymbol{x}] \in H^{s^*}(\Omega) \backslash \{0\}$, $s^* > 0$, with $\|f^*\|_{L^\infty(P_X)} < B_\infty$ for some constant $B_\infty > 0$, $\mathbb{E}y^2 < \infty$ and there are constants $\sigma, L > 0$ such that

$$\mathbb{E}\Big[|y - f^*(\boldsymbol{x})|^m \,\Big|\, \boldsymbol{x}\Big] \leq \frac{1}{2} m! \, \sigma^2 \, L^{m-2},$$

for P_X-almost all $\boldsymbol{x} \in \Omega$ and all $m \geq 2$.

The above moment condition holds for additive Gaussian noise with variance $\sigma^2 > 0$. Hence Assumption (D1) is strictly stronger than Assumption (D3). The spike components $\check{k}_{\gamma_n}$ can also be chosen as in Theorem G.1.

Theorem G.5. *Assume Assumption (D3) holds and that the kernel components satisfy:*

- *the RKHS $\tilde{\mathcal{H}}$ of $\tilde{k}$ satisfies $\tilde{\mathcal{H}} = H^s$ as sets with $s > \max(s^*, d/2)$,*
- *$\check{k}_{\gamma_n}$ denotes the Laplace kernel with a sequence of positive bandwidths (γ_n) fulfilling $\gamma_n \leq n^{-\frac{2+\alpha}{d}} \left((\frac{11}{4} + \frac{\alpha}{2}) \ln n\right)^{-1}$, where $\alpha > 0$ is arbitrary.*

Then there exists a constant $C > 0$ independent of n and there is a sequence $(\rho_n)_{n \in \mathbb{N}}$ of order $n^{-s/(s^*+d/2)}$ such that the minimum-norm interpolant f_{ρ_n, γ_n} of the ρ_n-regularized spiky-smooth kernel sequence $k_n := k_{\rho_n, \gamma_n}$ fulfills, with probability at least $1 - 6n^{-(1 \wedge \alpha)}$, for n large enough,

$$R_P(f_{\rho_n, \gamma_n}) - R_P(f^*) \leq C \, n^{-\frac{s^*}{(s^*+d/2)}} \log^2(n).$$

Proof. **Step 1: Kernel ridge regression $\hat{f}_{\rho_n}^{\tilde{k}}$ with optimal regularization achieves the desired convergence rate, with high probability.**

We slightly modify the proof of Theorem 8. Instead of using (Steinwart, 2001, Theorem 3.11 or Example 4.6), we use results of Fischer and Steinwart (2020). Here we first note that in the case

$\mathcal{H} = H^s(\Omega), \Omega \subseteq \mathbb{R}^d$ as RKHSs, we could directly use (Fischer and Steinwart, 2020, Corollary 5). Since we only have equivalent norms and also want to consider $\Omega = \mathbb{S}^d$, see Lemma F.7, we need to resort to the underlying more general result (Fischer and Steinwart, 2020, Theorem 1). To this end, we first need to verify its Assumptions (EMB), (EVD), (SRC), and (MOM).

Step 1.1: Verifying (MOM). The moment condition (MOM) on the noise distributions holds since we assumed it in Assumption (D3).

Step 1.2: Simpler equivalent spaces. We verify the remaining conditions by analyzing them for a nicer equivalent RKHS $\mathcal{H}$ and with uniform distribution ν on Ω. For the non-spherical case $\Omega \subseteq \mathbb{R}^d$, we choose $\mathcal{H} := H^s(\Omega)$. For the case $\Omega = \mathbb{S}^d$, we choose $\mathcal{H}$ as an RKHS associated to a dot-product kernel k with $\mu_l = \Theta((l+1)^{-2s})$, such that $\mathcal{H} \cong H^s \cong \tilde{\mathcal{H}}$ by Lemma B.1. In each case, $\mathcal{H} = \tilde{\mathcal{H}}$ with equivalent norms, and $L_2(\nu) = L_2(P_X)$ with equivalent norms since we assumed in (D3) that P_X has an upper- and lower-bounded density.

Step 1.3: Verifying (EVD+). It suffices to verify the eigenvalue decay condition (EVD+) for $\mathcal{H}$ and ν, since Lemma F.10 and Lemma F.9 then allow to transfer it to $\tilde{\mathcal{H}}$ and P_X.

For $\Omega \subseteq \mathbb{R}^d$, as pointed out in front of (Fischer and Steinwart, 2020, Corollary 5), it is well-known that $\mathcal{H}$ satisfies the polynomial eigenvalue decay assumption (EVD+) for $p := \frac{d}{2s}$.

For $\Omega = \mathbb{S}^d$, our definition of $\mathcal{H}$ together with Lemma B.1 directly yields (EVD+) for $\mathcal{H}$ and ν with $p := \frac{d}{2s}$.

Step 1.4: Verifying (EMB) and (SRC). The remaining two conditions of (Fischer and Steinwart, 2020, Theorem 1) are stated in terms of so-called power spaces, which in turn can be described by interpolation spaces of the real method. We therefore quickly recall these spaces. To this end, let us assume that we have two Banach spaces E and F such that $F \subset E$ and the corresponding inclusion map is continuous. Then the so-called K-functional of an $x \in E$ is defined by

$$K(x, t, E, F) := \inf_{y \in F} \left(t\|y\|_F + \|x - y\|_E \right), \qquad t > 0.$$

For $q \in (0,1)$ and $x \in E$ we then define

$$\|x\|_{q,2}^2 := \int_0^\infty t^{-2q-1} K^2(x, t, E, F)\, dt$$

and $[E, F]_{q,2} := \{x \in E : \|x\|_{q,2} < \infty\}$. Let us now consider the cases $(E, F) = (L_2(\nu), \mathcal{H})$ and $(E, F) = (L_2(P_X), \tilde{\mathcal{H}})$. Now, for a suitable constant C, $\mathcal{H}$ and $\tilde{\mathcal{H}}$ are C-equivalent, and $L_2(\nu)$ and $L_2(P_x)$ are also C-equivalent. We then find

$$C^{-1} K(f, t, L_2(\Omega), \mathcal{H}) \leq K(f, t, L_2(\Omega), \tilde{\mathcal{H}}) \leq C K(f, t, L_2(\Omega), \mathcal{H})$$

for all $t > 0$ and $f \in L_2(\Omega)$, and consequently we have

$$[L_2(\nu), \mathcal{H}]_{q,2} = [L_2(P_X), \tilde{\mathcal{H}}]_{q,2}$$

for all $q \in (0,1)$ with C-equivalent norms. Now, (Steinwart and Scovel, 2012, Theorem 4.6) shows that the power space $[\mathcal{H}]_\nu^q$ defined in (Steinwart and Scovel, 2012, Equation (36)) satisfies

$$[\mathcal{H}]_\nu^q = [L_2(\nu), \mathcal{H}]_{q,2}$$

with equivalent norms, and an analogous result is true for $\tilde{\mathcal{H}}$ and P_X. Moreover, $\mathcal{H}$ is dense in $L_2(\nu)$, and therefore (Steinwart and Scovel, 2012, Equations (36) and (18)) together with (Steinwart and Scovel, 2012, Lemma 2.2) show $[\mathcal{H}]_\nu^0 = L_2(\nu)$ as spaces. Again, we analogously find $[\tilde{\mathcal{H}}]_{P_X}^0 = L_2(P_X)$ Consequently, for all $0 \leq q < 1$ we have

$$[\mathcal{H}]_\nu^q = [\tilde{\mathcal{H}}]_{P_X}^q$$

with equivalent norms. From this we easily deduce that the Assumptions (EMB) and (SRC) are satisfied for $(\tilde{\mathcal{H}}, P_X)$ if and only if they are satisfied for $(\mathcal{H}, \nu)$.

Step 1.4.1: Non-spherical case. Now, let $\Omega \subseteq \mathbb{R}^d$. Then, (EMB) and (SRC) are satisfied for $(\mathcal{H}, \nu)$ for $\beta := s^*/s$ and an arbitrary but fixed $\alpha \in (p, \min\{1, p+\beta\})$ as outlined in front of (Fischer and Steinwart, 2020, Corollary 5). Applying Part ii) of (Fischer and Steinwart, 2020, Theorem 1) for

$\gamma = 0$ then shows that there exists a sequence $(\rho_n)_{n\in\mathbb{N}}$ of order $n^{-s/(s^*+d/2)}$ and a constant $K_1 > 0$ independent of n such that, for n large enough,

$$P^n\left(D \in (\mathbb{R}^d \times \mathbb{R})^n \mid \mathbb{E}_{\boldsymbol{x}}(\hat{f}^{\tilde{k}}_{\rho_n}(\boldsymbol{x}) - f^*(\boldsymbol{x}))^2 \geq K_1 n^{-\frac{s^*}{(s^*+d/2)}} \log^2(n)\right) \leq 4n^{-1}. \quad \text{(G.3)}$$

Step 1.4.2: Spherical case. Suppose $\Omega = \mathbb{S}^d$. Using the differently normalized spherical harmonics $Y_{l,i}$ and $\tilde{Y}_{l,i}$ from Appendix B.3, we obtain for the power spaces:

$$\begin{aligned}
[\mathcal{H}]^q_\nu &= \left\{ \sum_{l=0}^{\infty} \sum_{i=1}^{N_{l,d}} a_{li} \tilde{\mu}_i^{q/2} [\tilde{Y}_{l,i}]_\sim \mid (a_{li}) \in \ell_2 \right\} \\
&= \left\{ \sum_{l=0}^{\infty} \sum_{i=1}^{N_{l,d}} a_{l,i} (l+1)^{-qs} [Y_{l,i}]_\sim \mid (a_{li}) \in \ell_2 \right\} \\
&= \left\{ \sum_{l=0}^{\infty} \sum_{i=1}^{N_{l,d}} b_{l,i} [Y_{l,i}]_\sim \mid \sum_{l=0}^{\infty} \sum_{i=1}^{N_{l,d}} b_{l,i}^2 (l+1)^{2qs} < \infty \right\} \\
&= H^{qs}(\mathbb{S}^d) \,,
\end{aligned}$$

where the first equation follows from (Steinwart and Scovel, 2012, Eq. (36)), the second one from our definition of the μ_l in Step 1.2, and the last one from (Hubbert et al., 2023, Section 3). Again, we can choose an arbitrary but fixed $\alpha \in (p, \min\{1, p+\beta\})$. Then, $\alpha s > ps = d/2$, which means that $[\mathcal{H}]^\alpha_\nu = H^{\alpha s}$ is an RKHS with bounded kernel (De Vito et al., 2021, Theorem 8), hence the embedding condition (EMB) holds. Similarly, (SRC) holds for $\beta := s^*/s$ and the result follows as above.

Step 2: $\hat{f}^{k_n}_0$ and $\hat{f}^{\tilde{k}}_{\rho_n}$ are close in $L_2(P_X)$, with high probability.

Since $\frac{s^*}{(s^*+d/2)} < 1$, it suffices to show that k_n fulfills, for some constant $K_2 > 0$,

$$P^n\left(D \in (\mathbb{R}^d \times \mathbb{R})^n \mid \mathbb{E}_{\boldsymbol{x}}(\hat{f}^{k_n}_0(\boldsymbol{x}) - \hat{f}^{\tilde{k}}_{\rho_n}(\boldsymbol{x}))^2 \geq K_2 n^{-1}\right) \leq 2n^{-\alpha}. \quad \text{(G.4)}$$

Since $\gamma_n \leq n^{-\frac{2+\alpha}{d}} \left((\frac{11}{4} + \frac{\alpha}{2}) \ln n\right)^{-1}$, it holds that $|\check{k}_{\gamma_n}(\boldsymbol{x}, \boldsymbol{y})| \leq \varepsilon_n := \rho_n n^{-\frac{5+\alpha}{2}}$ (cf. Remark G.2). Then the product of all terms in Eq. (G.1) satisfies $O(n^{4+\alpha}\rho_n^{-2}\varepsilon_n^2) = o(n^{-1})$ with probability at least $1 - 2n^{-\alpha}$. On the same event, the bound on Eq. (G.2) remains of order $O(n^{-1})$, which shows Eq. (G.4). Combining (G.3) and (G.4) with the triangle inequality in $L_2(P_X)$ concludes the proof. □

Remark G.6 (Optimality of the rates)**.** In the setting of Theorem G.5, we can apply (Fischer and Steinwart, 2020, Theorem 2) in order to obtain lower bounds on the achievable rates. We have already verified the conditions (MOM), (EVD+), (EMB), and (SRC) in the proof of Theorem G.5. In the case $s^* > d/2$, we have $\beta = s^*/s > d/(2s) = p$ and can therefore choose $\alpha \in (p, \beta)$ such that $\beta > \alpha$. Then, (Fischer and Steinwart, 2020, Theorem 2) yields a lower bound on the rate of the form (with constant probability)

$$n^{-\frac{\beta}{\beta+p}} = n^{-\frac{s^*}{s^*+d/2}} \,,$$

which matches the rates in Theorem G.5 up to log terms. ◀

G.1 Auxiliary results for the proof of Theorem 8

The distributional Assumption (D2) immediately implies that the training points are separated with high probability.

Lemma G.7. *Assume (D2) is fulfilled with $\beta_X > 0$. Then with probability at least $1 - O(n^{-\alpha})$,*

$$\min_{i,j\in[n]:i\neq j} |\boldsymbol{x}_i - \boldsymbol{x}_j| \geq n^{-\frac{2+\alpha}{\beta_X}}.$$

Proof. For any $i \in [n]$, the union bound implies

$$P(\min_{j\in[n]:i\neq j} |\boldsymbol{x}_i - \boldsymbol{x}_j| \leq \delta) = P\left(\bigcup_{j\in[n]:j\neq i} \{\boldsymbol{x}_j \in B_\delta(\boldsymbol{x}_i)\}\right) \leq (n-1)\phi(\delta).$$

Another union bound yields

$$P(\min_{i,j\in[n]:i\neq j} |\boldsymbol{x}_i - \boldsymbol{x}_j| \leq \delta) \leq n(n-1)\phi(\delta).$$

Choosing $\delta_\alpha(n) = n^{-\frac{2+\alpha}{\beta_X}}$ yields $\phi(\delta_\alpha(n)) = O(\frac{1}{n^{2+\alpha}})$, which concludes the proof. □

The following lemma bounds $\|\boldsymbol{A}^{-1} - \boldsymbol{B}^{-1}\|$ via $\|\boldsymbol{A}^{-1}\|$ and $\|\boldsymbol{A} - \boldsymbol{B}\|$. Similar results can for example be found in (Horn and Johnson, 2013, Section 5.8).

Lemma G.8. *Let $\boldsymbol{A}, \boldsymbol{B} \in \mathbb{R}^{n\times n}$ be invertible matrices and let $\|\cdot\|$ be a submultiplicative matrix norm with $\|\boldsymbol{I}_n\| = 1$. If $\boldsymbol{A}$ and $\boldsymbol{B}$ fulfill $\|\boldsymbol{A}^{-1}\|\|\boldsymbol{A} - \boldsymbol{B}\| < 1$, then it holds that*

$$\|\boldsymbol{B}^{-1} - \boldsymbol{A}^{-1}\| \leq \frac{\|\boldsymbol{A}^{-1}\|^2 \cdot \|\boldsymbol{A} - \boldsymbol{B}\|}{1 - \|\boldsymbol{A}^{-1}\| \cdot \|\boldsymbol{A} - \boldsymbol{B}\|}\,.$$

Proof. Because of $\|\boldsymbol{A}^{-1}(\boldsymbol{A} - \boldsymbol{B})\| \leq \|\boldsymbol{A}^{-1}\|\|\boldsymbol{A} - \boldsymbol{B}\| < 1$ we get

$$\|\boldsymbol{I} - \boldsymbol{A}^{-1}(\boldsymbol{A} - \boldsymbol{B})\| \geq 1 - \|\boldsymbol{A}^{-1}\|\|\boldsymbol{A} - \boldsymbol{B}\|.$$

Writing $\boldsymbol{B} = \boldsymbol{A}(\boldsymbol{I} - \boldsymbol{A}^{-1}(\boldsymbol{A} - \boldsymbol{B}))$ yields $\boldsymbol{B}^{-1} = (\boldsymbol{I} - \boldsymbol{A}^{-1}(\boldsymbol{A} - \boldsymbol{B}))^{-1}\boldsymbol{A}^{-1}$ which implies

$$\|\boldsymbol{B}^{-1}\| \leq \frac{\|\boldsymbol{A}^{-1}\|}{1 - \|\boldsymbol{A}^{-1}\|\|\boldsymbol{A} - \boldsymbol{B}\|}.$$

Now write $\boldsymbol{B}^{-1} - \boldsymbol{A}^{-1} = \boldsymbol{A}^{-1}(\boldsymbol{A} - \boldsymbol{B})\boldsymbol{B}^{-1}$ to get

$$\|\boldsymbol{B}^{-1} - \boldsymbol{A}^{-1}\| \leq \|\boldsymbol{A}^{-1}\|\|\boldsymbol{A} - \boldsymbol{B}\|\|\boldsymbol{B}^{-1}\|.$$

Combining the last two inequalities concludes the proof. □

G.2 RKHS norm bounds

Here we show that if $\tilde{k}$ and $\check{k}_\gamma$ have RKHS equivalent to some Sobolev space H^s, $s > d/2$, then the RKHS of the spiky-smooth kernel $k_{\rho,\gamma}$ is also equivalent to H^s, for any fixed $\rho, \gamma > 0$. Hence all members of the spiky-smooth kernel sequence may have RKHS equivalent to a Sobolev space H^s and are individually inconsistent due to Theorem 1; yet the sequence is consistent. This shows that when arguing about generalization properties based on RKHS equivalence, the constants matter and the narrative that depth does not matter in the NTK regime as in Bietti and Bach (2021) is too simplified.

The following proposition states that the sum of kernels with equivalent RKHS yields an RKHS that is equivalent to the RKHS of the summands. For example, the spiky-smooth kernel with Laplace components possesses an RKHS equivalent to the RKHS of the Laplace kernel.

Proposition G.9. *Let $\mathcal{H}_1$ and $\mathcal{H}_2$ denote the RKHS of k_1 and k_2 respectively. If $\mathcal{H}_1 = \mathcal{H}_2$ then the RKHS $\mathcal{H}$ of $k = k_1 + k_2$ fulfills $\mathcal{H} = \mathcal{H}_1$. Moreover, if $C \geq 1$ is a constant with $\frac{1}{C}\|f\|_{\mathcal{H}_2} \leq \|f\|_{\mathcal{H}_1} \leq C\|f\|_{\mathcal{H}_2}$, then we have $\frac{1}{\sqrt{2}C}\|f\|_{\mathcal{H}_1} \leq \|f\|_{\mathcal{H}} \leq \|f\|_{\mathcal{H}_1}$.*

Proof. The RKHS of $k = k_1 + k_2$ is given by $\mathcal{H} = \mathcal{H}_1 + \mathcal{H}_2$ with norm

$$\|f\|_{\mathcal{H}}^2 = \min\{\|f_1\|_{\mathcal{H}_1}^2 + \|f_2\|_{\mathcal{H}_2}^2 :\ f = f_1 + f_2, f_1 \in \mathcal{H}_1, f_2 \in \mathcal{H}_2\}\,.$$

To see this we consider the map $\Phi : X \to \mathcal{H}_1 \times \mathcal{H}_2$ defined by $\Phi(\boldsymbol{x}) := (\Phi_1(\boldsymbol{x}, \cdot), \Phi_2(\boldsymbol{x}, \cdot))$ for all $\boldsymbol{x} \in X$, where X is the set, the spaces $\mathcal{H}_i$ live on and $\Phi_i(\boldsymbol{x}) := k_i(\boldsymbol{x}, \cdot)$. The reproducing property of k_1 and k_2 immediately ensures that Φ is a feature map of $k_1 + k_2$ and Theorem E.3 then shows

$$\begin{aligned}\mathcal{H} &= \left\{\langle w, \Phi(\cdot)\rangle_{\mathcal{H}_1\times\mathcal{H}_2} : w \in \mathcal{H}_1 \times \mathcal{H}_2\right\} \\ &= \left\{\langle w_1, \Phi_1(\cdot)\rangle_{\mathcal{H}_1} + \langle w_2, \Phi_2(\cdot)\rangle_{\mathcal{H}_2} : w_1 \in \mathcal{H}_1, w_2 \in \mathcal{H}_2\right\} = \mathcal{H}_1 + \mathcal{H}_2\end{aligned}$$

as well as the formula for the norm on $\mathcal{H}$. Now let $f \in \mathcal{H}$. Considering the decomposition $f = f_1 + 0$ then gives $\|f\|_{\mathcal{H}} \leq \|f\|_{\mathcal{H}_1}$. Moreover, for $f = f_1 + f_2$ with $f_i \in \mathcal{H}_i$ we have

$$\|f\|_{\mathcal{H}_1} \leq \|f_1\|_{\mathcal{H}_1} + \|f_2\|_{\mathcal{H}_1} \leq \|f_1\|_{\mathcal{H}_1} + C\|f_2\|_{\mathcal{H}_2} \leq \sqrt{2}C\big(\|f_1\|_{\mathcal{H}_1}^2 + \|f_2\|_{\mathcal{H}_1}^2\big)^{1/2}.$$

Taking the infimum over all decomposition then yields the estimate $\|f\|_{\mathcal{H}_1} \leq \sqrt{2}C\|f\|_{\mathcal{H}}$. □

H Spiky-smooth activation functions induced by Gaussian components

Here we explore the properties of the NNGP and NTK activation functions induced by spiky-smooth kernels with Gaussian components.

To offer some more background, it is well-known that NNGPs and NTKs on the sphere $\mathbb{S}^d$ are dot-product kernels, i.e., kernels of the form $k_d(\boldsymbol{x}, \boldsymbol{x}') = \kappa(\langle \boldsymbol{x}, \boldsymbol{x}' \rangle)$, where the function κ has a series representation $\kappa(t) = \sum_{i=0}^{\infty} b_i t^i$ with $b_i \geq 0$ and $\sum_{i=0}^{\infty} b_i < \infty$. The function κ is independent of the dimension d of the sphere. Conversely, all such kernels can be realized as NNGPs or NTKs (Simon et al., 2022, Theorem 3.1).

As dot-product kernel $k(\boldsymbol{x}, \boldsymbol{y}) = \kappa(\langle \boldsymbol{x}, \boldsymbol{y} \rangle)$ on the sphere, the Gaussian kernel has the simple analytic expression,

$$\kappa_\gamma^{Gauss}(z) = \exp\left(\frac{2(z-1)}{\gamma}\right),$$

with Taylor expansion

$$\kappa_\gamma^{Gauss}(z) = \sum_{i=0}^{\infty} \underbrace{\frac{2^i}{\gamma^i i!} \exp(-2/\gamma)}_{b_i^{Gauss}} \; z^i.$$

For spiky-smooth kernels $k = \tilde{k} + \rho \check{k}_\gamma$ with Gaussian components $\tilde{k}$ and $\check{k}_\gamma$ of width $\tilde{\gamma}$ and γ respectively, we get Taylor series coefficients

$$b_i = \frac{\exp(-2/\tilde{\gamma})}{i!} \left(\frac{2}{\tilde{\gamma}}\right)^i + \rho \frac{\exp(-2/\gamma)}{i!} \left(\frac{2}{\gamma}\right)^i. \tag{H.1}$$

Now Theorem 11 states that as soon as κ induces a dot-product kernel for every input dimension d, then the dot-product kernels can be written as the NNGP kernel of a 2-layer fully-connected network without biases and with the induced activation function

$$\phi_{NNGP}^{\kappa}(x) = \sum_{i=0}^{\infty} s_i b_i^{1/2} h_i(x),$$

or as the NTK of a 2-layer fully-connected network without biases and with the induced activation function

$$\phi_{NTK}^{\kappa}(x) = \sum_{i=0}^{\infty} s_i \left(\frac{b_i}{i+1}\right)^{1/2} h_i(x),$$

where h_i denotes the i-th Probabilist's Hermite polynomial normalized such that $\|h_i\|_{L_2(\mathcal{N}(0,1))} = 1$ and $s_i \in \{-1, +1\}$ are arbitrarily chosen for all $i \in \mathbb{N}_0$.

Now we can study the induced activation functions if we know the kernel's Taylor coefficients $(b_i)_{i \in \mathbb{N}_0}$. If infinitely many $b_i > 0$, then infinitely many activation functions induce the same dot-product kernel, with different choices of the signs s_i. For alternating signs $s_i = (-1)^i$, the symmetry property $h_i(-x) = (-1)^i h_i(x)$ of the Hermite polynomials implies

$$\phi_{NNGP,+-}(x) = \phi_{NNGP,+}(-x), \qquad \phi_{NTK,+-}(x) = \phi_{NTK,+}(-x).$$

To form an orthonormal basis of $L_2(\mathcal{N}(0,1))$ the unnormalized Probabilist's Hermite polynomials He_i have to be normalized by $h_i(x) = \frac{1}{\sqrt{i!}} He_i(x)$. We can use the identity $\exp(xt - \frac{t^2}{2}) = \sum_{i=0}^{\infty} He_i(x) \frac{t^i}{i!}$ with $t = \sqrt{2/\gamma}$ to analytically express the NNGP activation of the Gaussian kernel with all $s_i = +1$ as the exponential function

$$\phi_{NNGP,+}^{Gauss}(x) = \exp(-1/\gamma) \sum_{i=0}^{\infty} \frac{1}{i!} \left(\frac{2}{\gamma}\right)^{\frac{i}{2}} h_i(x) = \exp\left(\left(\frac{2}{\gamma}\right)^{\frac{1}{2}} \cdot x - \frac{2}{\gamma}\right). \tag{H.2}$$

Remarkably, the Gaussian kernel can not only be induced by an exponential activation function, but also by a single shifted sine activation function. This is shown in the following proposition.

Proposition H.1 (**Trigonometric Gaussian NNGP activation functions**). *For any $\gamma > 0$ and the bi-alternating choice of signs $\{(-1)^{\lfloor i/2 \rfloor}\}_{i=0,1,2,\ldots}$, the Gaussian kernel of bandwidth γ can be realized as the NNGP kernel of a two-layer fully-connected network without biases and with activation function*

$$\phi^{Gauss}_{NNGP,++--}(x) = \sin((2/\gamma)^{1/2}x) + \cos((2/\gamma)^{1/2}x).$$

Proof. We write $c = 2/\gamma$. We need to show that

$$\sin(c^{1/2}x) + \cos(c^{1/2}x) = e^{-c/2}\sum_{i=0}^{\infty}(-1)^{\lfloor i/2 \rfloor}\frac{c^{i/2}}{i!}He_i(x).$$

We will use the fact that

$$e^{2xz-z^2} = \sum_{i=0}^{\infty}\frac{z^i}{i!}H_i(x),$$

with the choices $z_1 = i\sqrt{c/2}$ and $z_2 = -i\sqrt{c/2}$. Now, using $e^{iax+b} = e^b(\cos(ax) + i\sin(ax))$, observe that

$$\begin{aligned}
\sin(\sqrt{c}x) = \sin(\sqrt{2c}x/\sqrt{2}) &= \frac{1}{2ie^{c/2}}\left(e^{ix\sqrt{c}+c/2} - e^{ix\sqrt{c}+c/2}\right) \\
&= \frac{1}{2ie^{c/2}}\left(\sum_{i=0}^{\infty}\frac{(i\sqrt{c/2})^i}{i!}H_i(x/\sqrt{2})(1-(-1)^i)\right) \\
&= e^{-c/2}\sum_{i=0}^{\infty}(-1)^i\frac{(\sqrt{c/2})^{2i+1}}{(2i+1)!}H_{2i+1}(x/\sqrt{2}).
\end{aligned}$$

An analogous calculation yields

$$\cos(c^{1/2}x) = e^{-c/2}\sum_{i=0}^{\infty}(-1)^i\frac{(\sqrt{c/2})^{2i}}{(2i)!}H_{2i}(x/\sqrt{2}).$$

Finally, using $H_i(x/\sqrt{2}) = 2^{i/2}He_i(x)$, we get

$$\begin{aligned}
\sin(c^{1/2}x) + \cos(c^{1/2}x) &= e^{-c/2}\sum_{i=0}^{\infty}(-1)^{\lfloor i/2 \rfloor}\frac{(c/2)^{i/2}}{i!}H_i(x/\sqrt{2}) \\
&= e^{-c/2}\sum_{i=0}^{\infty}(-1)^{\lfloor i/2 \rfloor}\frac{c^{i/2}}{i!}He_i(x).
\end{aligned}$$

□

For $\phi_{NNGP}(x) = \sum_{i=0}^{\infty} s_i\sqrt{b_i}h_i(x)$, we get $\|\phi\|^2_{L_2(\mathcal{N}(0,1))} = \sum_{i=0}^{\infty} b_i$ invariant to the choice $\{s_i\}_{i\in\mathbb{N}}$. For Gaussian NNGP activation components with bandwidth $\gamma > 0$ this yields

$$\|\phi^{Gauss}_{NNGP}\|^2_{L_2(\mathcal{N}(0,1))} = \exp(-2/\gamma)\sum_{i=0}^{\infty}\frac{1}{i!}\left(\frac{2}{\gamma}\right)^i = 1, \tag{H.3}$$

because $\{h_i\}_{i\in\mathbb{N}_0}$ is an ONB of $L_2(\mathcal{N}(0,1))$. Analogously, for Gaussian NTK activation components, we get

$$\begin{aligned}
\|\phi^{Gauss}_{NTK}\|^2_{L_2(\mathcal{N}(0,1))} &= \exp(-2/\gamma)\sum_{i=0}^{\infty}\frac{1}{(i+1)!}\left(\frac{2}{\gamma}\right)^i \\
&= \exp(-2/\gamma)\frac{\gamma}{2}\sum_{i=1}^{\infty}\frac{1}{i!}\left(\frac{2}{\gamma}\right)^i = \frac{\gamma}{2}\left(1-\exp\left(-\frac{2}{\gamma}\right)\right).
\end{aligned} \tag{H.4}$$

This implies that the average amplitude of NNGP activation functions does not depend on γ, while the average amplitude of NTK activation functions decays with $\gamma \to 0$.

By the fact $h_n'(x) = \sqrt{n} h_{n-1}(x)$, we know that any activation function $\phi(x) = \sum_{i=0}^{\infty} s_i a_i h_i(x)$ has the derivative $\phi'(x) = \sum_{i=0}^{\infty} s_i a_{i+1} \sqrt{i+1} \cdot h_i(x)$ as long as $\sum_{i=0}^{\infty} |a_{i+1}\sqrt{i+1}| < \infty$.

The following proposition formalizes the additive approximation $\phi^k \approx \phi^{\tilde{k}} + \rho^{1/2}\phi^{\check{k}_\gamma}$, and quantifies the necessary scaling of γ for any demanded precision of the approximation.

Proposition H.2. *Fix $\tilde{\gamma}, \rho > 0$ arbitrary. Let $k = \tilde{k} + \rho\check{k}_\gamma$ denote the spiky-smooth kernel where $\tilde{k}$ and $\check{k}_\gamma$ are Gaussian kernels of bandwidth $\tilde{\gamma}$ and γ, respectively. Assume that we choose the activation functions ϕ^k_{NTK}, $\phi^{\tilde{k}}_{NTK}$ and $\phi^{\check{k}_\gamma}_{NTK}$ as in Theorem 11 with same signs $\{s_i\}_{i\in\mathbb{N}}$. Then, for $\gamma > 0$ small enough, it holds that*

$$\|\phi^k_{NTK} - (\phi^{\tilde{k}}_{NTK} + \sqrt{\rho}\cdot\phi^{\check{k}_\gamma}_{NTK})\|^2_{L_2(\mathcal{N}(0,1))} \leq 2^{1/2}\rho\gamma^{3/2}\exp\left(-\frac{1}{\gamma}\right) + \frac{4\pi\gamma(1+\tilde{\gamma})}{\tilde{\gamma}},$$
$$\|\phi^k_{NNGP} - (\phi^{\tilde{k}}_{NNGP} + \sqrt{\rho}\cdot\phi^{\check{k}_\gamma}_{NNGP})\|^2_{L_2(\mathcal{N}(0,1))} \leq 2^{3/2}\rho\gamma^{1/2}\exp\left(-\frac{1}{\gamma}\right) + \frac{8\pi\gamma(1+\tilde{\gamma})}{\tilde{\gamma}^2}.$$

Proof. Let $b_{i,\gamma} = \frac{2^i}{\gamma^i i!}\exp(-2/\gamma)$ denote the Taylor coefficients of the Gaussian kernel. All considered infinite series converge absolutely.

$$\begin{aligned}
&\|\phi^{\tilde{\gamma},\gamma,\rho}_{NNGP} - (\phi^{\tilde{\gamma}}_{NNGP} + \sqrt{\rho}\cdot\phi^{\gamma}_{NNGP})\|^2_{L_2(\mathcal{N}(0,1))} \\
=&\|\sum_{i=0}^{\infty} s_i\sqrt{b_{i,\tilde{\gamma}} + \rho b_{i,\gamma}}h_i(x) - \sum_{i=0}^{\infty} s_i(\sqrt{b_{i,\tilde{\gamma}}} + \sqrt{\rho b_{i,\gamma}})h_i(x)\|^2_{L_2(\mathcal{N}(0,1))} \\
=&\sum_{i=0}^{\infty}\left(\sqrt{b_{i,\tilde{\gamma}} + \rho b_{i,\gamma}} - (\sqrt{b_{i,\tilde{\gamma}}} + \sqrt{\rho b_{i,\gamma}})\right)^2 \\
\leq& \underbrace{2\sum_{i=0}^{I}(\sqrt{b_{i,\tilde{\gamma}} + \rho b_{i,\gamma}} - b^{1/2}_{i,\tilde{\gamma}})^2}_{(I)} + \underbrace{2\rho\sum_{i=0}^{I} b_{i,\gamma}}_{(II)} + \underbrace{2\sum_{i=I+1}^{\infty}(\sqrt{b_{i,\tilde{\gamma}} + \rho b_{i,\gamma}} - \rho^{1/2}b^{1/2}_{i,\gamma})^2}_{(III)} + \underbrace{2\sum_{i=I+1}^{\infty} b_{i,\tilde{\gamma}}}_{(IV)},
\end{aligned}$$

for any $I \in \mathbb{N}$. To bound (I) observe

$$\sum_{i=0}^{I}(\sqrt{b_{i,\tilde{\gamma}} + \rho b_{i,\gamma}} - b^{1/2}_{i,\tilde{\gamma}})^2 = \sum_{i=0}^{I}\left(\rho b_{i,\gamma} + 2b_{i,\tilde{\gamma}}\left(1 - \sqrt{1 + \frac{\rho b_{i,\gamma}}{b_{i,\tilde{\gamma}}}}\right)\right) \leq \rho\sum_{i=0}^{I} b_{i,\gamma}.$$

An analogous calculation for (III) yields

$$\sum_{i=I+1}^{\infty}(\sqrt{b_{i,\tilde{\gamma}} + \rho b_{i,\gamma}} - \rho^{1/2}b^{1/2}_{i,\gamma})^2 \leq \sum_{i=I+1}^{\infty} b_{i,\tilde{\gamma}}.$$

So overall we get the bound

$$\|\phi^{\tilde{\gamma},\gamma,\rho}_{NNGP} - (\phi^{\tilde{\gamma}}_{NNGP} + \sqrt{\rho}\cdot\phi^{\gamma}_{NNGP})\|^2_{L_2(\mathcal{N}(0,1))} \leq 4\rho\sum_{i=0}^{I} b_{i,\gamma} + 4\sum_{i=I+1}^{\infty} b_{i,\tilde{\gamma}}. \tag{H.5}$$

Now, defining $c \coloneqq 2/\gamma$,

$$\sum_{i=0}^{I} b_{i,\gamma} = \exp(-c)\sum_{i=0}^{I}\frac{1}{i!}c^i = \frac{\Gamma(I+1,c)}{I!},$$

where $\Gamma(k+1,c)$ denotes the upper incomplete Gamma function. Choosing $I = \lfloor\frac{c}{2\pi}\rfloor$, (Pinelis, 2020, Theorem 1.1) yields, for $c \geq 121$,

$$\begin{aligned}
\frac{\Gamma(I+1,c)}{I!} &\leq \exp(-c)\frac{(c + (I+1)!^{1/I})^{I+1}}{(I+1)!\cdot(I+1)!^{1/I}} \leq \frac{\exp(-c)(c+I)^{I+1}}{(I+1)!^{(I+1)/I}} \\
&\leq \frac{\exp(-c)(c+I)^{I+1}}{(2\pi(I+1))^{1/2}\left(\frac{I+1}{e}\right)^{(I+1)^2/I}}
\end{aligned}$$

$$\leq \frac{1}{\sqrt{2\pi(I+1)}} \exp\Big(-c + (I+1)\big(\ln(c+I) - \ln(I+1) + 1\big)\Big)$$
$$\leq \frac{1}{\sqrt{c}} \exp\Big(-c + (\frac{c}{2\pi}+1)\big(\ln(\frac{2\pi+1}{2\pi}c) - \ln(\frac{c}{2\pi}) + 1\big)\Big)$$
$$\leq \frac{1}{\sqrt{c}} \exp\Big(-c + (\frac{c}{2\pi}+1)\big(\ln(2\pi+1) + 1\big)\Big) \leq \frac{\exp\big(-\frac{c}{2}\big)}{\sqrt{c}}, \tag{H.6}$$

where we used $(I+1)!^{1/I} \leq I$ for $I \geq 3$ in the first line, Stirling's approximation in the second line, and $(\frac{c}{2\pi}+1)\big(\ln(2\pi+1)+1\big) \leq c/2$ for $c \geq 121$ in the last line.

It is obvious that

$$\sum_{i=I+1}^{\infty} b_{i,\tilde{\gamma}} \to 0, \quad \text{for } I = \lfloor \frac{c}{2\pi} \rfloor \to \infty.$$

To quantify the rate of convergence, we use the bound $\Gamma(I+1, c_0) \geq e^{-c_0} I!(1 + c_0/(I+1))^I$, which follows from applying Jensen's inequality to $\Gamma(I+1, c_0) = e^{-c_0} I! \mathbb{E}(1 + c_0/G)^I$, where $G \sim \Gamma(I+1, 1)$ and $\mathbb{E}G = I+1$. Defining $c_0 = 2/\tilde{\gamma}$, it holds that

$$\sum_{i=I+1}^{\infty} b_{i,\tilde{\gamma}} \leq 1 - \frac{\Gamma(I+1, c_0)}{I!} \leq 1 - e^{-c_0}\left(1 + \frac{c_0}{I+1}\right)^I \leq 1 - e^{-c_0}\left(1 + \frac{c_0}{I+1}\right)^{I+1}.$$

Taking the first two terms of the Laurent series expansion of $n \mapsto \big(1 + \frac{c_0}{n}\big)^n$ about $n = \infty$ yields $\Big(1 + \frac{c_0}{I+1}\Big)^{I+1} > e^{c_0}(1 - \frac{c_0^2}{2(I+1)})$ for I large enough (where we demand $\gamma \in o(\tilde{\gamma}^2)$), thus

$$\sum_{i=I+1}^{\infty} b_{i,\tilde{\gamma}} \leq 1 - e^{-c_0}\left(1 + \frac{c_0}{I+1}\right)^{I+1} \cdot \left(1 + \frac{c_0}{I+1}\right)^{-1}$$
$$\leq \frac{c_0/(I+1) + c_0^2/(2(I+1))}{1 + c_0/(I+1)} \leq \frac{c_0}{I+1} + \frac{c_0^2}{2(I+1)} \leq \frac{4\pi}{\tilde{\gamma}c} + \frac{4\pi}{\tilde{\gamma}^2 c}. \tag{H.7}$$

Plugging (H.6) and (H.7) into (H.5) yields, for $\gamma \leq 1/61$,

$$\|\phi^{\tilde{\gamma},\gamma,\rho}_{NNGP} - (\phi^{\tilde{\gamma}}_{NNGP} + \sqrt{\rho} \cdot \phi^{\gamma}_{NNGP})\|^2_{L_2(\mathcal{N}(0,1))} \leq 2^{3/2}\rho\gamma^{1/2} \exp\left(-\frac{1}{\gamma}\right) + \frac{8\pi\gamma(1+\tilde{\gamma})}{\tilde{\gamma}^2}.$$

For the NTK we get

$$\|\phi^{\tilde{\gamma},\gamma,\rho}_{NTK} - (\phi^{\tilde{\gamma}}_{NTK} + \sqrt{\rho} \cdot \phi^{\gamma}_{NTK})\|^2_{L_2(\mathcal{N}(0,1))}$$
$$= \|\sum_{i=0}^{\infty} s_i \sqrt{\frac{b_{i,\tilde{\gamma}} + \rho b_{i,\gamma}}{i+1}} h_i(x) - \sum_{i=0}^{\infty} s_i \left(\sqrt{\frac{b_{i,\tilde{\gamma}}}{i+1}} + \sqrt{\frac{\rho b_{i,\gamma}}{i+1}}\right) h_i(x)\|^2_{L_2(\mathcal{N}(0,1))}$$
$$= \sum_{i=0}^{\infty} \frac{1}{i+1}\left(\sqrt{b_{i,\tilde{\gamma}} + \rho b_{i,\gamma}} - (\sqrt{b_{i,\tilde{\gamma}}} + \sqrt{\rho b_{i,\gamma}})\right)^2.$$

We can proceed exactly as for the NNGP, but choose $I = \lfloor \frac{c}{2\pi} \rfloor - 1$ to get

$$\sum_{i=0}^{I} \frac{b_{i,\gamma}}{i+1} = \exp(-c) \sum_{i=0}^{I} \frac{c^i}{(i+1)!} = \frac{\exp(-c)}{c}\left(\sum_{i=0}^{I+1} \frac{c^i}{i!} - 1\right) \leq \frac{\exp(-c/2)}{c^{3/2}} - \frac{\exp(-c)}{c},$$

and replace (H.7) with

$$\sum_{i=I+1}^{\infty} \frac{b_{i,\tilde{\gamma}}}{i+1} = \frac{\exp(-c_0)}{c_0} \sum_{i=I+2}^{\infty} \frac{c_0^i}{i!} \leq \frac{1}{I+2} + \frac{c_0}{2(I+2)} \leq \frac{\pi\gamma(1+\tilde{\gamma})}{\tilde{\gamma}}. \qquad \square$$

I Additional experimental results

The code to reproduce all our experiments is provided in the supplementary material and under

`https://github.com/moritzhaas/mind-the-spikes`

Our implementations rely on `PyTorch` (Paszke et al., 2019) for neural networks and `mpmath` (Johansson et al., 2023) for high-precision calculations.

I.1 Experimental details of Figure 1

For the kernel experiment (Figure 1a), we used the Laplace kernel with bandwidth 0.4 and the spiky-smooth kernel (4) with Laplace components with $\rho = 1, \tilde{\gamma} = 1, \gamma = 0.01$.

For the neural network experiment (Figure 1b,c) we initialize 2-layer networks with NTK parametrization (Jacot et al., 2018) and He initialization (He et al., 2015). Using the antisymmetric initialization trick from Zhang et al. (2020) doubles the network width from 10000 to 20000 and helps to prevent errors induced by the random initialization function. It might also be helpful to increase the initialization variance (Chizat et al., 2019). We train the network with stochastic gradient descent of batch size 1 over the 15 training samples with learning rate 0.04 for 2500 epochs. Training with gradient descent and learning rate 0.4 produces similar results. We use the spiky-smooth activation function given by $x \mapsto ReLU(x) + 0.01 \cdot (\sin(100x) + \cos(100x))$, which corresponds to $x \mapsto ReLU(x) + \omega_{NTK}(x, \frac{1}{5000})$, including both even and uneven Hermite coefficients.

I.2 Disentangling signal from noise in neural networks with spiky-smooth activation functions

Since our spiky-smooth activation function has the additive form $\sigma_{spsm}(x) = ReLU(x) + \omega_{NTK}(x; \frac{1}{5000})$, we can dissect the learned neural network

$$f_{spsm}(\boldsymbol{x}) = \boldsymbol{W}_2 \cdot \sigma_{spsm}(\boldsymbol{W}_1 \cdot \boldsymbol{x} + \boldsymbol{b}_1) + b_2 = f_{ReLU}(\boldsymbol{x}) + f_{spikes}(\boldsymbol{x}) \tag{I.1}$$

into its $ReLU$-component

$$f_{ReLU}(\boldsymbol{x}) = \boldsymbol{W}_2 \cdot ReLU(\boldsymbol{W}_1 \cdot \boldsymbol{x} + \boldsymbol{b}_1) + b_2,$$

and its spike component

$$f_{spikes}(\boldsymbol{x}) = \boldsymbol{W}_2 \cdot \omega_{NTK}(\boldsymbol{W}_1 \cdot \boldsymbol{x} + \boldsymbol{b}_1; \frac{1}{5000}).$$

If the analogy to the spiky-smooth kernel holds and f_{spikes} fits the noise in the labels while having a small L_2-norm, then f_{ReLU} would have learned the signal in the data. Indeed Figure I.1 demonstrates that this simple decomposition is useful to disentangle the learned signal from the spike component in our setting. The figure also suggests that the oscillations in the activations of the hidden layer constructively interfere to interpolate the training points, while the differing frequencies and phases approximately destructively interfere on most of the remaining covariate support. Figure I.2 shows some of the functions generated by the hidden layer neurons of the spike component f_{spikes}. Both the phases and frequencies vary. Destructive interference in sums of many oszillations occurs, for example, under a uniform phase distribution.

An exciting direction of future work will be to understand when and why the neural networks with spiky-smooth activation functions learn the target function well, and when the decomposition into $ReLU$- and spike component succeeds to disentangle the noise from the signal. Particular challenges will be to design architectures and learning algorithms that provably work on complex data sets and to determine their statistical convergence rates. A different line of work could evaluate whether there exist useful spike components for deep and narrow networks beyond the pure infinite-width limit. Maybe for deep architectures is suffices to apply spiky-smooth activation functions only between the penultimate and the last layer.

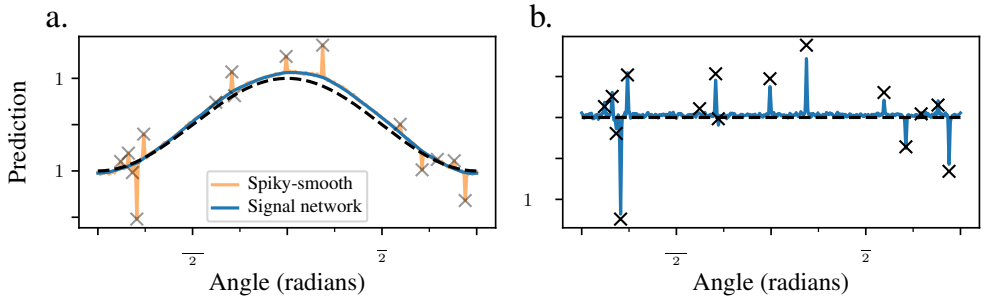


Figure I.1: **a.** The $ReLU$-component f_{ReLU} (blue) and the full spiky-smooth network f_{spsm} (orange) of the learned neural network from Figure 1. **b.** The spike component f_{spikes} of the learned neural network from Figure 1 against the label noise in the training set, derived by subtracting the signal from the training points. Observe that the $ReLU$-component has learned the signal, while the spike component has fitted the noise in the data while regressing to 0 between data points.

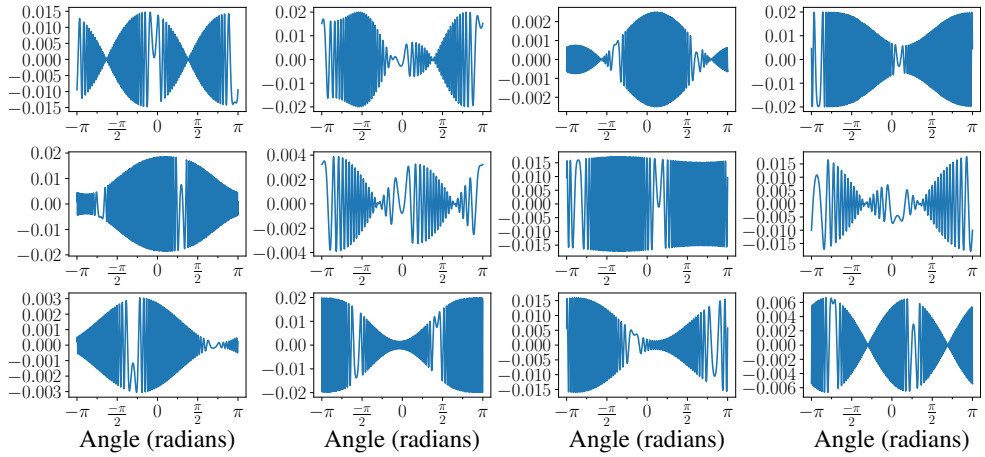


Figure I.2: Here we plot the functions learned by 12 random hidden layer neurons of the spike component network f_{spikes} corresponding to Figure 1.

Of course an analogous additive decomposition exists for the minimum-norm interpolant $\hat{f}_0^k$ of the spiky-smooth kernel,

$$\hat{f}_0^k(\boldsymbol{x}) = (\tilde{k} + \rho_n \check{k}_{\gamma_n})(\boldsymbol{x}, \boldsymbol{X}) \cdot (\tilde{\boldsymbol{K}} + \rho_n \check{\boldsymbol{K}}_{\gamma_n})^{-1} \boldsymbol{y} = f_{signal}(\boldsymbol{x}) + f_{spikes}(\boldsymbol{x}), \tag{I.2}$$

where

$$f_{signal}(\boldsymbol{x}) = \tilde{k}(\boldsymbol{x}, \boldsymbol{X}) \cdot (\tilde{\boldsymbol{K}} + \rho_n \check{\boldsymbol{K}}_{\gamma_n})^{-1} \boldsymbol{y}, \qquad f_{spikes}(\boldsymbol{x}) = \rho_n \check{k}_{\gamma_n}(\boldsymbol{x}, \boldsymbol{X}) \cdot (\tilde{\boldsymbol{K}} + \rho_n \check{\boldsymbol{K}}_{\gamma_n})^{-1} \boldsymbol{y}.$$

We plot the results in Figure I.3. Observe that the spikes f_{spikes} regress to 0 more reliably than in the neural network.

Although spiky-smooth estimators can be consistent, any method that interpolates noise cannot be adversarially robust. The signal component f_{signal} may be a simple correction towards robust estimators. Figure I.4 suggests that the signal components of spiky-smooth estimators behave more robustly than ReLU networks or minimum-norm interpolants of Laplace kernels in terms of finite-sample variance.

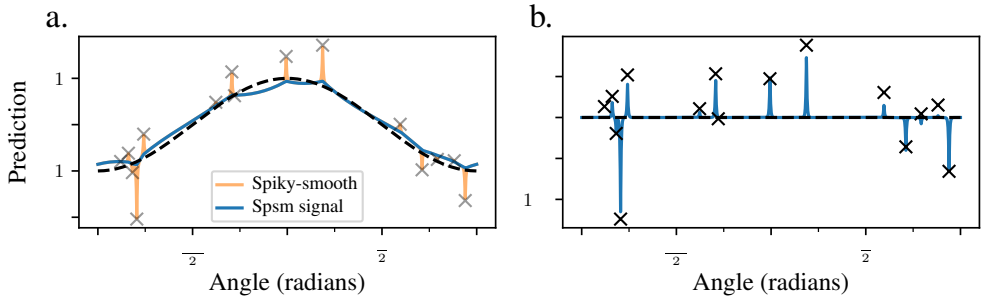


Figure I.3: **a.** The signal component f_{signal} (blue) and the full minimum-norm interpolant $\hat{f}_0^k$ (orange) of the spiky-smooth kernel from Figure 1. **b.** The spike component f_{spikes} of the spiky-smooth kernel from Figure 1 against the label noise in the training set, derived by subtracting the signal from the training points.

I.3 Repeating the finite-sample experiments

We repeat the experiment from Figure 1 100 times, both randomizing with respect to the training set and with respect to neural network initialization.

For the kernels (Figure I.4a), observe that all minimum-norm kernel interpolants are biased towards 0. While the Laplace kernel and the signal component (I.2) of the spiky-smooth kernel have similar averages, the spiky-smooth kernel has a slightly larger bias. However, both the spiky-smooth kernel as well as its signal component produces lower variance estimates than the Laplace kernel.

Considering the trained neural networks (Figure I.4b), the ReLU networks are approximately unbiased, but have large variance. The neural networks with spiky-smooth activation function as well as the extracted signal network (I.1) are similar on average: They are slighly biased towards 0, but have much smaller variance than the ReLU networks.

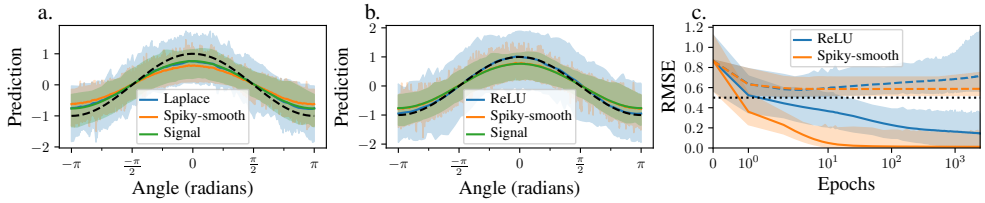


Figure I.4: We repeat the experiment from Figure 1 100 times and report the mean values (lines). Confidence bands denote the interval between the empirical 2.5%- and 97.5%-quantiles from the 100 independent runs.

The training curves (Figure I.4c) offer similar conclusions as Figure 1: While the ReLU networks harmfully overfit over the course of training, the neural networks with spiky-smooth activation function quickly overfit to 0 training error with monotonically decreasing test error, which on average is almost optimal, already with only 15 training points. The spiky-smooth networks have smaller confidence bands, indicating increased robustness compared to the ReLU networks. If the ReLU networks would be early-stopped with perfect timing, they would generalize similarly well as the networks with spiky-smooth activation function.

I.4 Spiky-smooth activation functions

In Figures I.5 and I.6 we plot the 2-layer NTK activation functions induced by spiky-smooth kernels with Gaussian components, where $\tilde{k}$ has bandwidth 1, and in the first figure $\rho = 1$ while in the second figure $\rho = 0.1$.

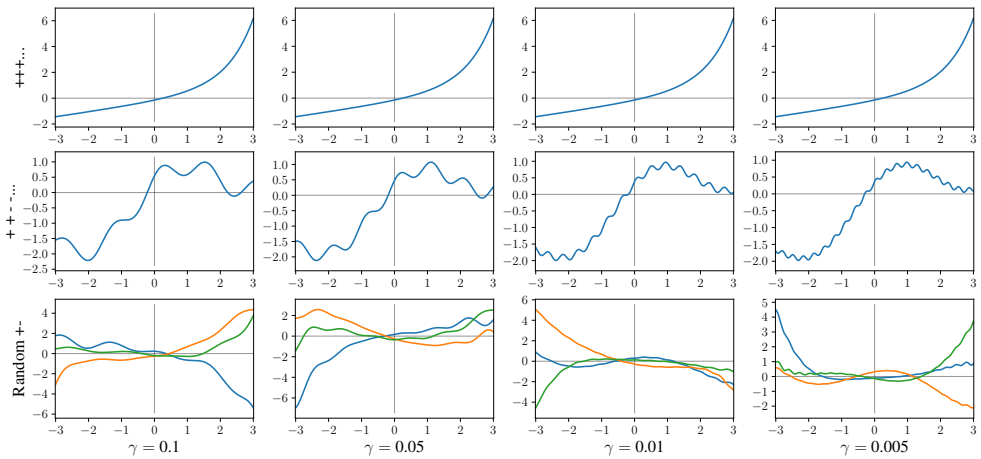


Figure I.5: The 2-layer NTK activation functions for Gaussian-Gaussian spiky-smooth kernels with varying γ (columns) with $k_{max} = 1000$, $\tilde{k}$ has bandwidth 1, $\rho = 1$. Top: all $s_i = +1$, middle: $+, +, -, -, +, +, ...$, bottom: Random $+1$ and -1. Although the activation function induced by the spiky-smooth kernel is not exactly the sum of the activation functions induced by its components, this approximation is accurate because the spike components approximately live in a subspace of higher frequency in the Hermite basis orthogonal to the low-frequency subspace of the smooth component.

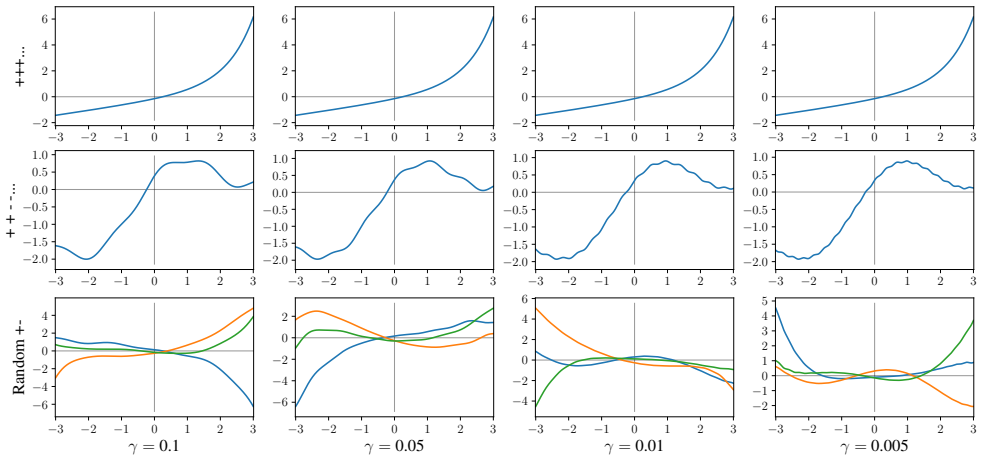


Figure I.6: Same as above but $\rho = 0.1$. The high-frequency fluctuations are much smaller compared to Figure I.5.

In Figure I.7 we plot the corresponding 2-layer NNGP activation functions with $\rho = 1$. In contrast to the NTK activation functions the amplitudes of the fluctuations only depends on ρ and not on γ. Our intuition is the following: Since the first layer weights are not learned in case of the NNGP, the first layer cannot learn constructive interference, so that the oscillations in the activation function need to be larger.

The additive approximation $\phi^k \approx \phi^{\tilde{k}} + \rho^{1/2}\phi^{\check{k}_\gamma}$ remains accurate in all considered cases (Appendix I.6).

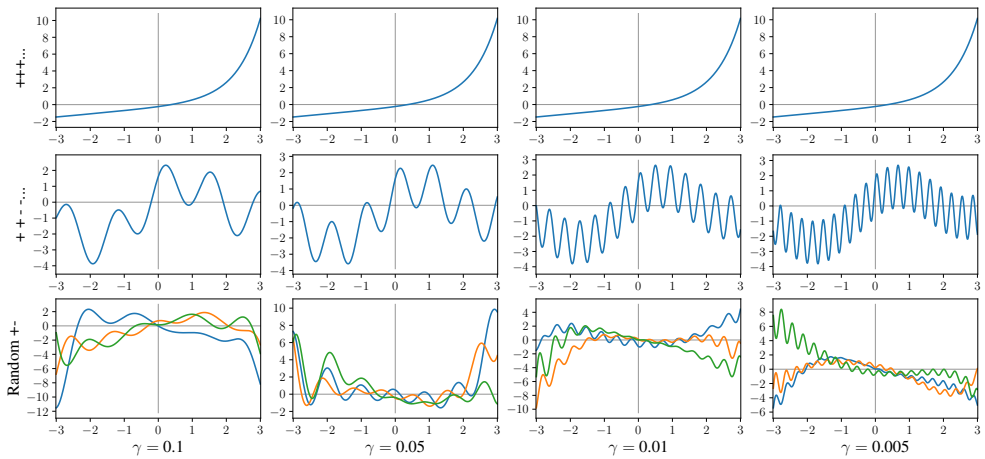


Figure I.7: Same as above but NNGP and $\rho = 1$. As expected from the isolated spike plots: Spikes essentially add fluctuations that increase in frequency and stay constant in amplitude for $\gamma \to 0$, ρ regulates the amplitude.

I.5 Isolated spike activation functions

Figure I.8 is the equivalent of Figure 3 for the NNGP.

By plotting the NTK activation components corresponding to Gaussian spikes $\phi^{\check{k}_\gamma}$ with varying choices of the signs s_i in Figure I.9, we observe the following properties:

1. All $s_i = +1$ leads to exponentially exploding activation functions, cf. Eq. (H.2).
2. If the signs s_i alternate every second i, i.e. $s_i = +1$ iff $\lfloor \frac{i}{2} \rfloor$ even, $\phi^{\check{k}_\gamma}$ is approximately a single shifted sin-curve with increasing frequency and decreasing/constant amplitude for NTK/NNGP activation functions, cf. Eq. (6).
3. If s_i is chosen uniformly at random, with high probability, $\phi^{\check{k}_\gamma}$ both oscillates at a high frequency around 0 and explodes for $|x| \to \infty$.

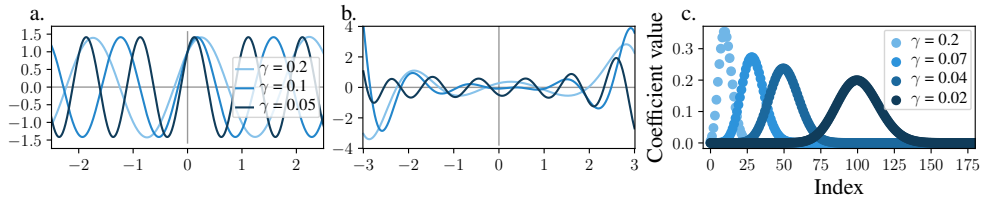


Figure I.8: Same as Figure 3 but for the NNGP. In contrast to the NTK, the amplitudes of the oscillations in a. do not shrink with $\gamma \to 0$. Otherwise the behaviour is analogous. For example, the Hermite coefficients peak at $2/\gamma$. The squared coefficients sum to 1 (Eq. (6)).

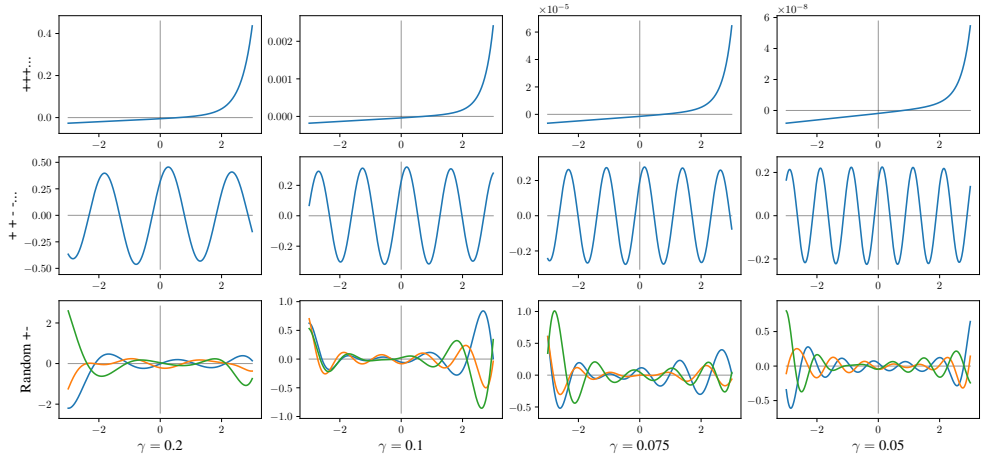


Figure I.9: The spike activation components of 2-layer NTK for Gaussian spikes with varying γ (columns), $k_{max} = 1000$, top: all $s_i = +1$, middle: signs alternate every second index, bottom: 3 draws from uniformly random signs. With $\gamma \to 0$, the amplitude shrinks, while the frequency increases.

Figure I.10 visualizes NNGP activation functions induced by Gaussian spikes with varying bandwidth γ. Observe similar behaviour as for the NTK but amplitudes invariant to γ as predicted by Eq. (6). For smaller γ the explosion of (all+) activation functions starts at larger x, but appears sharper as can be seen in the analytic expression (H.2).

Figure I.11 resembles Figure I.9 but plotted on a larger range to visualize the exploding behaviour for $|x| \to \infty$.

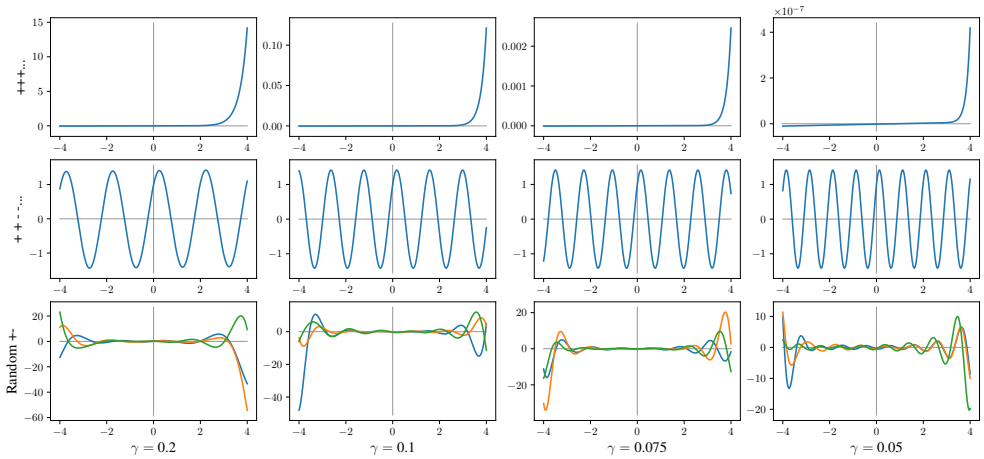


Figure I.10: Spike activation components as in Figure I.9, but for the NNGP with x between $[-4, 4]$.

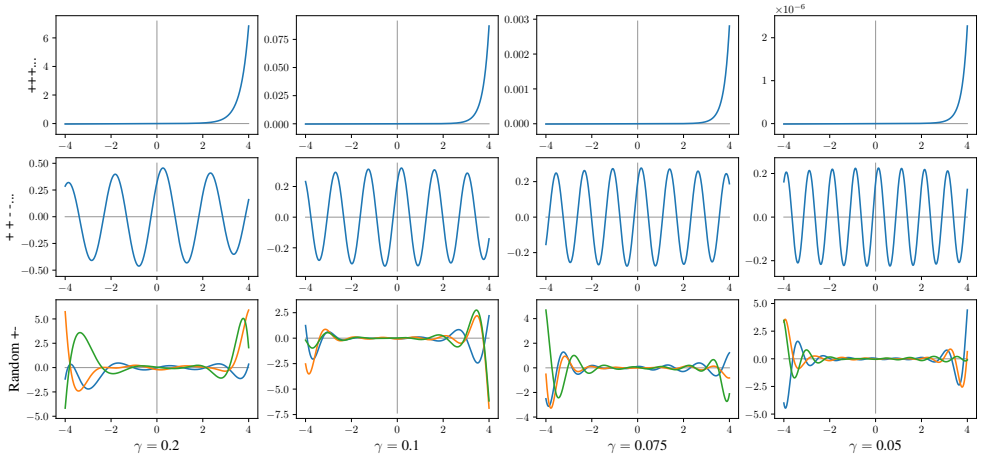


Figure I.11: Spike NTK activation components as in Figure I.9 but with x between $[-4, 4]$. The all+ NTK activation explodes exponentially. While random sign activations explode as well, $++--$-activations remain stable sin-fluctuations with slowly decaying amplitude for $|x| \to \infty$.

I.6 Additive decomposition and sin-fit

Here we quantify the error of the sin-approximation (8) of Gaussian NTK activation components. The additive decomposition $\phi^k \approx \phi^{\tilde{k}} + \rho^{1/2}\phi^{\check{k}_\gamma}$ quickly becomes accurate in the limit $\gamma \to 0$ (Figures I.12 and I.13), the sin-approximation seems to converge pointwise at rate $\Theta(|x|\gamma)$, where a good approximation can be expected when $|x| \ll 1/\gamma$. The error at large $|x|$ arises because the spike component decays for $|x| \to \infty$. For $O(1)$ inputs, we conjecture that this inaccuracy does not dramatically affect the test error of neural networks when γ is chosen to be small.

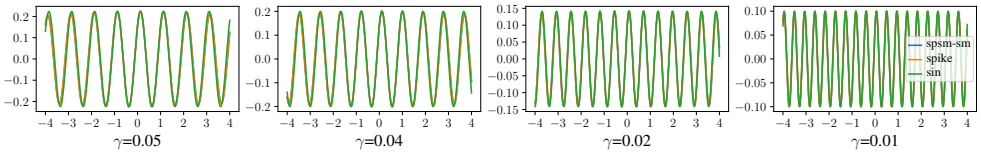


Figure I.12: The isolated NTK spike activation function (orange), the difference between spiky-smooth and smooth activation function (blue) and a fitted sin-curve (8) (green). All curves roughly align, in particular for $\gamma \to 0$.

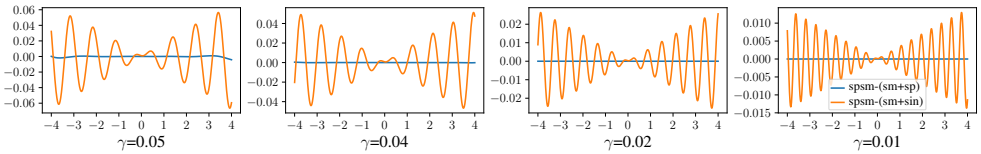


Figure I.13: The error of the additive decomposition $\phi^k \approx \phi^{\tilde{k}} + \rho^{1/2}\phi^{\check{k}_\gamma}$ (blue) and the sin-fit (8) (orange) for the NTK. While the additive decomposition makes errors of order $10^{-3}, 10^{-4}, 10^{-9}, 10^{-15}$ (from left to right) in the domain $[-4, 4]$, the sin-fit is increasingly inaccurate for $|x| \to \infty$, and increasingly accurate for $\gamma \to 0$.

Now we evaluate the numerical approximation quality of the sin-fits (7) and (8) to the isolated spike activation components $\phi^{\check{k}_\gamma}$. As expected by Proposition H.1, the NNGP oscillating activation function $\phi^{\check{k}_\gamma}$ of a Gaussian spike component corresponds to Eq. (7) up to numerical errors. Both for the NNGP and for the NTK, the approximations become increasingly accurate with smaller bandwidths $\gamma \to 0$ (Figure I.14). Again the approximation quality suffers for $|x| \to \infty$, since $\phi^{\check{k}_\gamma}_{NTK}$ slowly decay to 0 for $|x| \to \infty$.

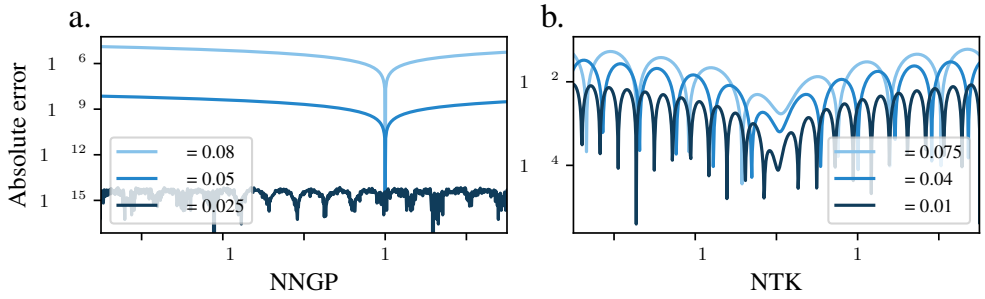


Figure I.14: Absolute numerical error between the oscillating activation function $\phi^{\check{k}_\gamma}$ of a Gaussian spike component and **a.** its analytical expression Eq. (7) for the NNGP and **b.** the approximation Eq. (8) for the NTK with varying bandwidth γ.

I.7 Spiky-smooth kernel hyper-parameter selection

To understand the empirical performance of spiky-smooth kernels on finite data sets, we generate i.i.d. data where $\boldsymbol{x} \sim \mathcal{U}(\mathbb{S}^d)$ and

$$y = \boldsymbol{x}_1 + \boldsymbol{x}_2^2 + \sin(\boldsymbol{x}_3) + \prod_{i=1}^{d+1} \boldsymbol{x}_i + \varepsilon,$$

with $\varepsilon \sim \mathcal{N}(0, \sigma^2)$ independent of $\boldsymbol{x}$ and evaluate the least squares excess risk of the minimum-norm interpolant. Figure I.15 shows that

- the smaller the spike bandwidth γ, the better. At some point, the improvement saturates,
- ρ should be carefully tuned, it has large impact. As with $\gamma \to 0$ ridgeless regression with the spiky-smooth kernel approximates ridge regression with $\tilde{k}$ and regularization ρ, simply choose the optimal regularization ρ^{opt} of ridge regression.
- The spiky-smooth kernel with Gaussian components exhibits catastrophic overfitting, when γ is too large (cf. Mallinar et al. (2022)), the Laplace kernel is more robust with respect to γ.
- With sufficiently thin spikes and properly tuned ρ, spiky-smooth kernels with Gaussian components outperform the Laplace counterparts.

We repeat the experiment in Figure I.16 with a slightly more complex generating function and come to the same conclusions.

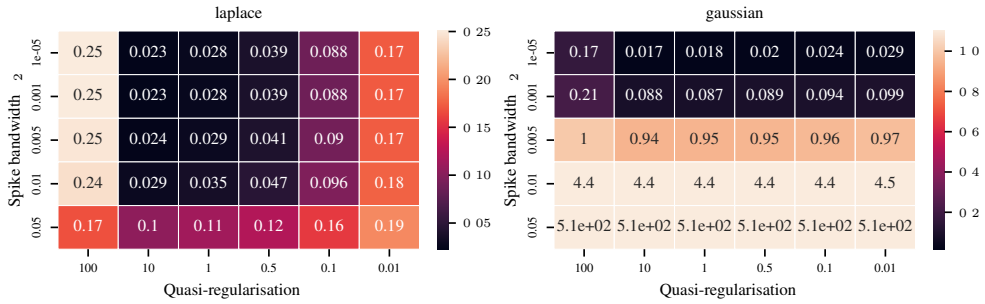


Figure I.15: Least squares excess risk for spiky-smooth kernel ridgeless regression with Laplace components (*left*) and Gaussian components (*right*), with $n = 1000, d = 2$, estimated on 10000 independent test points, $\sigma^2 = 0.5, \tilde{\gamma} = 1$. The smaller the spike bandwidth γ, the better. Properly tuning ρ is important.

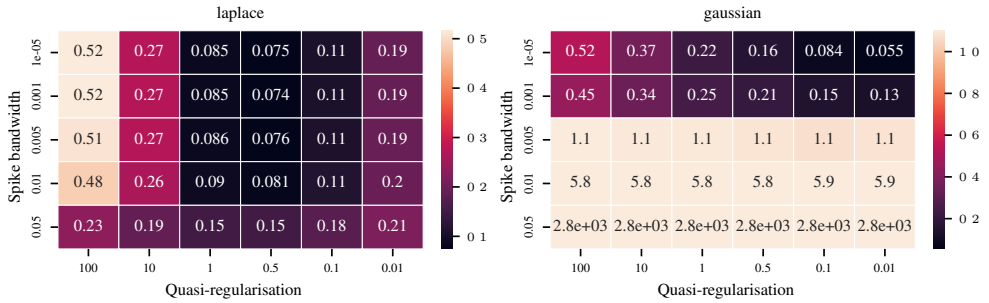


Figure I.16: Same as Figure I.15 but with the more complex generating function $y = |\boldsymbol{x}_1| + \boldsymbol{x}_2^2 + \sin(2\pi\boldsymbol{x}_3) + \prod_{i=1}^{d+1} \boldsymbol{x}_i + \varepsilon$. The errors are larger compared to Figure I.15 and the optimal values of ρ are smaller, but the conceptual conclusions remain the same.

Chapter 5

μP^2: Effective Sharpness Aware Minimization Requires Layerwise Perturbation Scaling

μP^2: Effective Sharpness Aware Minimization Requires Layerwise Perturbation Scaling

Moritz Haas[1] **Jin Xu**[2] **Volkan Cevher**[3,4] **Leena Chennuru Vankadara**[4]

[1]University of Tübingen, Tübingen AI Center* [2]University of Oxford*
[3]LIONS, EPFL* [4]AGI Foundations, Amazon

Abstract

Sharpness Aware Minimization (SAM) enhances performance across various neural architectures and datasets. As models are continually scaled up to improve performance, a rigorous understanding of SAM's scaling behaviour is paramount. To this end, we study the infinite-width limit of neural networks trained with SAM, using the Tensor Programs framework. Our findings reveal that the dynamics of standard SAM effectively reduce to applying SAM solely in the last layer in wide neural networks, even with optimal hyperparameters. In contrast, we identify a stable parameterization with layerwise perturbation scaling, which we call *Maximal Update and Perturbation Parameterization* (μP^2), that ensures all layers are both feature learning and effectively perturbed in the limit. Through experiments with MLPs, ResNets and Vision Transformers, we empirically demonstrate that μP^2 achieves hyperparameter transfer of the joint optimum of learning rate and perturbation radius across model scales. Moreover, we provide an intuitive condition to derive μP^2 for other perturbation rules like Adaptive SAM and SAM-ON, also ensuring balanced perturbation effects across all layers.

1 Introduction

Sharpness Aware Minimization (SAM) (Foret et al., 2021) and its variants (Kwon et al., 2021; Müller et al., 2024) improves generalization performance across a wide range of neural architectures and datasets (Chen et al., 2021; Kaddour et al., 2022). In the SAM formulation, we minimize a given loss L between our prediction and the data y as a function of the architecture's weights W, where an adversary simultaneously maximizes the same loss by perturbing the weights within a budget ρ.

A standard SAM update for an L-hidden layer multi layer perceptron (MLP) is given by

$$W_{t+1}^l = W_t^l - \eta_l \nabla_{W^l} \mathcal{L}\left(f\left(\xi_t; W_t + \boldsymbol{\varepsilon}_t\right), y_t\right), \quad \text{with} \quad \boldsymbol{\varepsilon}_t^l = \rho \cdot \frac{\nabla_{W^l} \mathcal{L}(f(\xi_t; W_t), y_t)}{\|\nabla_{\mathbf{W}} \mathcal{L}(f(\xi_t; W_t), y_t)\|_F}, \tag{SAM}$$

where $\mathbf{W} = [W^1, \ldots, W^{L+1}]$, t is the iteration count and $\boldsymbol{\varepsilon}_t^l$ denotes the perturbation in the l-th MLP layer with width $n \in \mathbb{N}$, and where we define an L-hidden layer MLP iteratively via

$$h^1(\xi) := W^1 \xi, \qquad x^l(\xi) := \phi(h^l(\xi)), \qquad h^{l+1}(\xi) := W^{l+1} x^l(\xi), \qquad f(\xi) := W^{L+1} x^L(\xi),$$

for inputs $\xi \in \mathbb{R}^{d_{\text{in}}}$ with trainable weight matrices $W^1 \in \mathbb{R}^{n \times d_{in}}$, $W^l \in \mathbb{R}^{n \times n}$ for $l \in [2, L]$, and $W^{L+1} \in \mathbb{R}^{d_{\text{out}} \times n}$. We call h^l preactivations, x^l activations, and $f(\xi)$ output function. Despite the inherent difficulty of non-convex, non-concave optimization, SAM is quite successful in practice.

*This work was conducted during Moritz', Jin's and Volkan's time at Amazon. Correspondence to: mo.haas@uni-tuebingen.de

38th Conference on Neural Information Processing Systems (NeurIPS 2024).

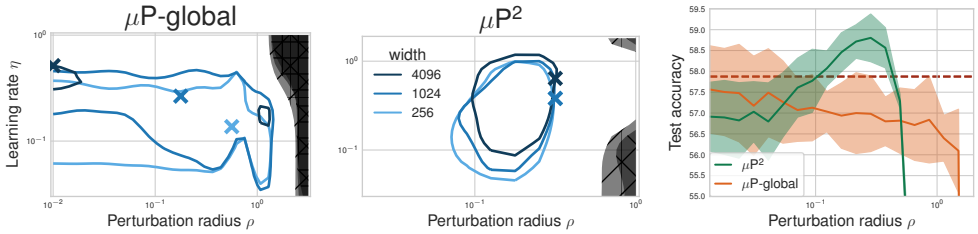


Figure 1: *Left and center* (**μP^2 transfers both η and ρ**): Test accuracy as a function of learning rate η and perturbation radius ρ of a 3-layer MLP in μP trained with SAM on CIFAR10 for various widths with global perturbation scaling $\rho \cdot n^{-1/2}$ (*left*) and our layerwise perturbations scaling μP^2 (*right*), averaged over 3 independent runs.'$\times$' denotes the optimum. Blue contours (the darker, the wider) denote the region within 1% of the optimal test accuracy smoothened with a Gaussian filter. Grey regions (the lighter, the wider) denote the unstable regime below 30% test accuracy. *Right* (**μP^2 improves generalization**): Same as left but sliced at the optimal learning rate of both parameterizations for width 4096 with the base optimizer SGD in μP (dashed line) as a baseline. Average and 2σ-CI from 16 independent runs. Global perturbation scaling $\rho \cdot n^{-1/2}$ achieves a width-independent critical perturbation radius at which training becomes unstable, but does not consistently improve over SGD in μP and does not transfer the optimal (η, ρ). μP^2 achieves joint transfer in (η, ρ) and improves generalization performance.

On the other hand, the steadily growing scale of foundation models has sparked considerable interest in scaling laws of model size and dataset size (Kaplan et al., 2020; Zhai et al., 2022). To rigorously understand learning dynamics under width scaling, Yang and Hu (2021) have recently provided general infinite-width theory for SGD, which has since been shown to be a good model for understanding the properties of large models (Vyas et al., 2024). Yang and Hu (2021) show that standard parameterizations (SP), including He or LeCun initialization (He et al., 2015; LeCun et al., 2002) with a global learning rate, do not learn features in the infinite-width limit.

Instead, a different scaling of layerwise initialization variances and learning rates, termed *Maximal Update Parameterization* (μP), is necessary to achieve feature learning in all layers in wide networks. A crucial practical benefit of μP is the transferability of the optimal learning rate across model scales (Yang et al., 2022). This can drastically reduce computational costs as it allows to tune hyperparameters on smaller representative models and then to train the large model only once.

Contributions. In this paper, we adopt a scaling perspective to understand SAM's learning dynamics. Using the Tensor Programs framework (Yang, 2019; Yang and Hu, 2021; Yang and Littwin, 2023), this work provides the first infinite-width theory for SAM with important practical consequences:

1. We show that training an MLP with the standard (SAM) update rule is equivalent to applying perturbations only in the last layer in the infinite-width limit, even if the perturbation radius is properly tuned. This holds for any width-dependent scaling of layerwise initialization variances and learning rates, including SP and μP.
2. We demonstrate that the optimal perturbation radius can shift significantly in μP (Figure 1).
3. We postulate that jointly transferring the optimal learning rate η and perturbation radius ρ requires width-independent feature learning and *effective perturbations in every layer* in the infinite-width limit. We define the perturbation of a trainable weight tensor to be *effective* iff its effect on the output function scales width-independently. We show that this can be achieved with *layerwise scalings* of the perturbation radius, and provide a complete characterization of perturbation scaling parameterizations into four regimes: unstable, vanishing, nontrivial and effective perturbations.
4. We derive the *Maximal Update and Perturbation Parameterization* (μP^2) that achieves both feature learning and effective perturbations in all layers in the infinite-width limit. We empirically demonstrate that μP^2 achieves hyperparameter transfer in both learning rate η and perturbation radius ρ (Figure 1).
5. We provide a versatile (spectral) scaling condition ($*$) applicable to architectures such as ResNets and Vision Transformers (ViTs), and to various SAM variants like SAM-ON and Adaptive SAM (ASAM), and any SAM updates modeled in a Tensor Program.

2 Background and related work

We here provide a short summary of related work. A more detailed account is provided in Appendix B.

Sharpness Aware Minimization. SAM was motivated as an inductive bias towards flatter minima and it provably reduces properties of the Hessian that are related to sharpness in simpler settings (Bartlett et al., 2023; Wen et al., 2023; Monzio Compagnoni et al., 2023). However a full understanding of why SAM works so well remains elusive (Andriushchenko et al., 2023b; Wen et al., 2024). For example, applying SAM on only the normalization layers (SAM-ON) often improves generalization further despite increasing sharpness (Müller et al., 2024). A plethora of SAM variants have recently been proposed with the purpose of even stronger performance or reducing SAM's computational and memory complexity. We focus on two variants of Adaptive SAM (ASAM) (Kwon et al., 2021) which achieve the overall strongest results in Müller et al. (2024) (see Appendix F.4 for more details).

Tensor Programs. We build on the Tensor Programs framework (Yang, 2019; Yang and Hu, 2021; Yang and Littwin, 2023; Yang et al., 2022, 2023b), which covers many modern deep learning architectures, optimization algorithms and arbitrary abc-parameterizations. Each abc-parameterization is essentially defined by a layerwise scaling of initialization variance and learning rate as a function of network width. Beyond pure infinite-width limits, the simple $\frac{1}{\sqrt{L}}$-scaling allows depth-scaling in ResNets and unlocks hyperparameter transfer across depths (Hayou et al., 2021; Li et al., 2021; Bordelon et al., 2023; Yang et al., 2023b). Noci et al. (2022, 2024) provide infinite width and depth analyses for Transformers with the goal of preventing rank collapse.

Look-LayerSAM (Liu et al., 2022) already considers layerwise perturbation scaling with the goal of preserving good performance under large batch training. However, achieving μP^2 with Look-LayerSAM requires nontrivial layerwise learning rate and perturbation rescaling (see Appendix B).

3 SAM induces vanishing perturbations in wide neural networks

This section shows that under the standard (SAM) update rule, weight perturbations induced by SAM vanish in the infinite-width limit in every layer except the output layer. We later demonstrate that other SAM variants also selectively perturb other subsets of layers. For enhanced readability of some formulae, we use colors to distinguish four regimes of perturbation behaviour: Unstable, vanishing, nontrivial and effective perturbations.

While our theory covers any stable parameterization including He and LeCun initializations, for concreteness and for the clarity of exposition, we first present our results for MLPs under μP:

$$\text{initialize} \quad W^1 \sim \mathcal{N}(0, 1/d_{in}),\ W^l \in \mathbb{R}^{n\times n} \sim \mathcal{N}(0,1/n) \text{ for } l \in [2, L],\ W^{L+1} \sim \mathcal{N}(0, 1/n^2)$$
$$\text{with layerwise SGD learning rates} \quad \eta_1 = \eta n,\ \eta_l = \eta, \text{ for } l \in [2, L],\ \eta_{L+1} = \eta n^{-1}.$$

By analyzing the infinite-width behaviour of the SAM update rule, we show that the training dynamics under standard (SAM) become unstable as the network width increases. This result is first stated informally below in Proposition 1 and then more formally in the next section.

Proposition 1 (Instability of standard SAM parameterization in wide neural networks). *Under μP with the standard (SAM) update rule and default perturbation given in (SAM), the output function becomes unbounded after the first update step in the infinite-width limit for any fixed, positive learning rate $\eta > 0$ and perturbation radius $\rho > 0$.*

Hence, to achieve stable optimization, it is necessary to introduce some width-dependent perturbation scaling ρn^{-d} for some suitable $d > 0$. To understand the layerwise scaling behaviour of SAM under this scaling, we define the notion of *vanishing perturbations*.

Vanishing perturbations. The weight perturbation $\boldsymbol{\varepsilon}^l$ perturbs the l-th layer's activations as

$$x^l + \tilde{\delta}x^l = \phi((W^l + \boldsymbol{\varepsilon}^l)(x^{l-1} + \tilde{\delta}x^{l-1})),$$

where $\tilde{\delta}x^l$ denotes the perturbation of the l-th layer's activations accumulated from the weight perturbations $\{\boldsymbol{\varepsilon}^{l'}\}_{l'\in[l]}$ in all previous layers. We say a layer l has *vanishing perturbations* if $\tilde{\delta}x^l \to 0$ as the width approaches infinity. This occurs if the weight perturbations in all previous layers are too small when measured in spectral norm, that is if $\|\boldsymbol{\varepsilon}^{l'}\|_*/\|W^{l'}\|_* \to 0$ for all $l' \in [l]$.

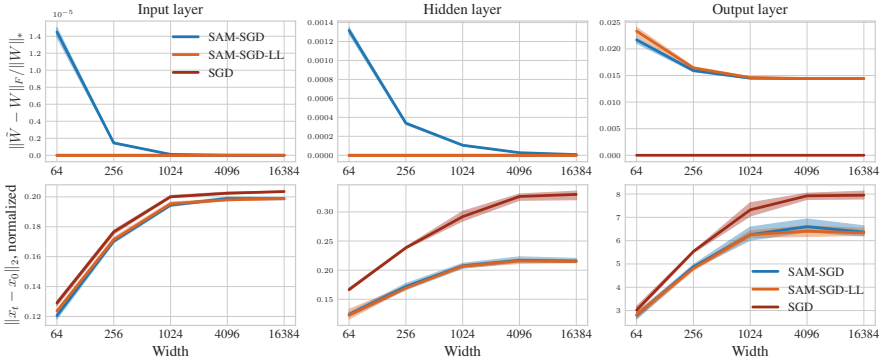


Figure 2: ((SAM) **effectively only perturbs the last layer)** Layerwise weight perturbations (top) and normalized activation updates $\|\Delta x^l\|_2$ (bottom) for SAM, last-layer SAM and SGD as a baseline across widths after training a 3-layer MLP in μP with global perturbation scaling $\rho \cdot n^{-1/2}$ for 20 epochs on CIFAR10. Average and CI are computed from 4 independent runs. Perturbations are normalized by the weight spectral norm to measure their effect on the layer's output. Activation updates are normalized by $\sqrt{\dim(\Delta x^l)}$ to measure coordinatewise updates. We provide more neural network statistics in Appendix H.1.

Informally, Proposition 2 below shows that for every choice of a decay parameter $d > 0$, either the training dynamics of SAM are unstable or all the hidden layers of the network have vanishing perturbations in the limit. The formal results are stated in the next section.

Proposition 2 (**Global perturbation scaling is unstable or induces vanishing perturbations**). *Fix $\rho > 0$ and $t \in \mathbb{N}$. Let $\mathring{f}_t$ denote the infinite-width limit of the output function after training an MLP of width n with the SAM update rule* (SAM) *with perturbation radius ρn^{-d} for t steps. If $d < 1/2$, then output perturbations blow up, and $\mathring{f}_t$ is unstable. If $d > 1/2$, then the perturbations in all layers vanish and $\mathring{f}_t$ corresponds to the limit after t steps of SGD. If $d = 1/2$, then only the last layer is effectively perturbed, all other layers have vanishing perturbations.*

Figure 2 shows statistics of an MLP trained with (SAM) with global width-dependent scaling $\rho n^{-1/2}$ versus the same MLP trained with SAM where *only the last-layer weights are perturbed* and $\boldsymbol{\varepsilon}^l = 0$ for all $l \in [L]$. As predicted by Proposition 2, both training algorithms produce equivalent training dynamics, already at moderate width, and last-layer perturbations are scaled correctly.

Remark 3 (**Practical implications**). According to Proposition 2, for sufficiently wide models, any performance gains from standard SAM are primarily due to applying the SAM update rule to the last layer even with a properly tuned perturbation radius. This implies that, when applying the standard SAM update rule (SAM), one can remove the inner backward pass beyond the last layer and nearly recover the computational cost of SGD. However, it may be undesirable for optimal generalization if only the last layer is perturbed (Figure 1). ◀

Layerwise perturbation scaling. In the next section, we show that correcting the (SAM) update rule to achieve *effective perturbations* in every single layer requires introducing additional hyperparameters — layerwise width-dependent scaling of the perturbation radius. This is similar in spirit to μP which corrects standard parameterization by introducing layerwise scaling of the learning rates. We postulate that achieving width-independent scaling of both updates and perturbations is a necessary condition for hyperparameter transfer under SAM. We also lay all theoretical foundations and derive the stable parameterization, we call *maximal update and perturbation parameterization* (μP^2) that achieves both **feature learning and effective perturbations in all layers in the infinite-width limit**.

Figure 1 shows that μP^2 indeed achieves hyperparameter transfer in the optimal joint choice of (η, ρ), while also achieving the best generalization performance (Table 2).

General perturbation scaling condition. For intuitive understanding and a generalization to other perturbation rules, a simple condition for achieving effective perturbations in any layer follows from our results: **in every layer, perturbations should scale like updates in μP**.

The reason is that both updates and perturbations are gradient-based $\nabla_{W^l}\mathcal{L} = \nabla_{h^l}\mathcal{L} \cdot (x^{l-1})^\top$, and thus low-rank and correlated with the incoming activations x^{l-1}. Therefore updates and perturbations introduce the same LLN-like scaling factors, and require the same layerwise scaling corrections. Like Yang et al. (2023a), we can rephrase this condition in terms of weight spectral norms to: For a weight matrix $W_t^l \in \mathbb{R}^{\texttt{fan\_out}\times\texttt{fan\_in}}$, its update δW_t^l and its perturbation $\boldsymbol{\varepsilon}_t^l$, it should hold at all times t that

$$\|\boldsymbol{\varepsilon}_t^l\|_* = \Theta\left(\|\delta W_t^l\|_*\right) = \Theta\left(\|W_t^l\|_*\right) = \Theta\left(\sqrt{\texttt{fan\_out}/\texttt{fan\_in}}\right). \tag{$*$}$$

with big-O notation that only tracks dependence on network width (Definition C.1). We discuss the spectral perspective in more detail in Appendix F.7.

4 Sharpness Aware Minimization in the infinite-width limit

4.1 Characterization of layerwise perturbation scaling: Unstable, vanishing, nontrivial and effective perturbations

To systematically and rigorously understand the width-scaling behaviour of neural networks trained under the SAM update rule, we propose a new class of parameterizations, which we refer to as *bcd-parameterizations*. Motivated by the analysis in Section 3, the class of *bcd*-parameterizations naturally extends *abc*-parameterizations (Yang and Hu, 2021) by including layerwise scaling of the perturbation radius. By setting all weight multiplier exponents $a_l = 0$, we do not need to modify the MLP architecture and recover representatives of each *abc*-parameterization that capture their essence and condense all equations: Ignoring numerical considerations (Blake et al., 2024), each *abc*-parameterization is essentially a layerwise initialization and learning rate scaling. The effects of weight multipliers on SAM are more nuanced than for SGD or Adam (see Remark 12 and Appendix F.6).

To study the infinite-width behaviour of networks trained with SAM in any *bcd*-parameterization, we utilize the theoretical framework of NE⊗OR⊤programs (Yang et al., 2023b). We write the two forward and backward passes for each SAM update (ascent/perturbation step then descent/update step) using the NE⊗OR⊤computation rules and rigorously track all relevant scalings as provided by the NE⊗OR⊤master theorem. All proofs are provided in Appendix E. The full formal result statements can be found in Appendix D. Further theoretical considerations and generalizations around perturbation scaling are provided in Appendix F.

Assumptions. For clarity of exposition, we present our main results for MLPs. Their extension to other architectures is discussed in Appendix F.5. For all of the results in this section, we assume that the used activation function is either `tanh` or σ-`gelu` for $\sigma > 0$ sufficiently small. For small enough $\sigma > 0$, σ-`gelu` (Definition C.9) approximates ReLU arbitrarily well. We also assume constant training time as width $n \to \infty$. We assume batch size 1 for clarity, but our results can be extended without further complications to arbitrary fixed batch size as well as differing fixed batch sizes for the ascent/perturbation and the descent/update step, as sometimes used for SAM (Foret et al., 2021). Considering small perturbation batch size is practical, as it has been observed to enhance SAM's generalization properties (Andriushchenko and Flammarion, 2022).

Definition 4 (*bcd*-**parametrization**)**.** A *bcd-parametrization* $\{b_l\}_{l\in[L+1]} \cup \{c_l\}_{l\in[L+1]} \cup \{d_l\}_{l\in[L+1]} \cup \{d\}$ defines the training of an MLP with SAM in the following way:

(a) Initialize weights iid as $W_{ij}^l \sim \mathcal{N}(0, n^{-2b_l})$.
(b) Train the weights using the SAM update rule with layerwise learning rates,

$$W_{t+1}^l = W_t^l - \eta n^{-c_l} \nabla_{W^l}\mathcal{L}\left(f\left(\xi_t; W_t + \boldsymbol{\varepsilon}_t\right), y_t\right),$$

with the scaled perturbation $\boldsymbol{\varepsilon}_t$ via layerwise perturbation radii,

$$\boldsymbol{\varepsilon}_t := \rho n^{-d} \frac{v_t}{\|v_t\|}, \quad \text{with} \quad v_t = (v_t^1, \ldots, v_t^{L+1}), \quad v_t^l := n^{-d_l} \cdot \nabla_{W^l}\mathcal{L}(f(\xi_t; W_t), y_t), \tag{LP}$$

W.l.o.g. we set $\|v_t\| = \Theta(1)$, which prevents nontrivial width-dependence from the denominator. This imposes the constraints: $d_1 \geq 1/2 - \min(b_{L+1}, c_{L+1})$, $d_l \geq 1 - \min(b_{L+1}, c_{L+1})$ for $l \in [2, L]$, and $d_{L+1} \geq 1/2$, with at least one equality required to hold (see Appendix E.1.3). The normalization $v_t/\|v_t\|$ removes one degree of freedom from $\{d_l\}_{l\in[L+1]}$ via the equivalence $\{d_l'\}_{l\in[L+1]} \cong \{d_l\}_{l\in[L+1]}$ iff there exists a $C \in \mathbb{R}$ such that $d_l' = d_l + C$ for all $l \in [L+1]$. ◀

Stability. To ensure that the training dynamics of SAM are well-behaved with scale, we require *bcd*-parameterizations to satisfy conditions of stability. Perturbed weights $\tilde{W}^l = W^l + \boldsymbol{\varepsilon}^l$ induce perturbed activations $x^l + \tilde{\delta}x^l$ and a perturbed output function $\tilde{f}_t(\xi) := f_{\tilde{W}_t}(\xi)$. We call a *bcd*-parameterization *stable* (Definition C.3) if the hidden activations have width-independent scaling $\Theta(1)$ at initialization and during training, and neither the updates nor the perturbations $\tilde{\delta}x^l$ of the activations or output logits $\tilde{f}_t - f_t$ blow up at any point in training.

For stating the conditions that characterize the class of stable *bcd*-parameterizations, we define the *maximal feature perturbation scaling* $\tilde{r}$ of a *bcd*-parameterization as

$$\tilde{r} := \min(b_{L+1}, c_{L+1}) + d + \min_{l=1}^{L}(d_l - \mathbb{I}(l \neq 1)).$$

Similar to the maximal feature update scaling r from Yang and Hu (2021), $\tilde{r}$ describes how much the last hidden-layer activations are perturbed as a function of width, $x^L + \tilde{\delta}x^L = \Theta(n^{-\tilde{r}})$. Hidden-layer activation perturbations do not explode with width if and only if $\tilde{r} \geq 0$. The output perturbations not to blow up if and only if $d + d_{L+1} \geq 1$ and $b_{L+1} + \tilde{r} \geq 1$. In particular, this implies that any stable *bc*-parameterization together with naive perturbation scaling $d_l = d = 0$ for all $l \in [L+1]$ is *unstable due to blowup in the last layer*. We formally state the stability characterization in Theorem D.2. Ideally, we will later require width-independent perturbation scaling which is attained iff $\tilde{r} = 0$.

Effective SGD dynamics. Within the class of stable parameterizations, there are parameterizations in which perturbations in the output vanish in the infinite-width limit at any point during training. In other words, SAM training dynamics collapses to SGD dynamics with scale. We are mostly interested in the opposing class of parameterizations with non-vanishing perturbations. We characterize this class in Theorem 6 and refer to them as *perturbation nontrivial* (Definition 5).

Definition 5 (**Perturbation nontriviality**)**.** We say that a stable *bcd*-parametrization is *perturbation nontrivial* if there exists a training routine, $t \in \mathbb{N}_0$ and $\xi \in \mathbb{R}^{d_{\text{in}}}$ such that $\tilde{\delta}f_t(\xi) := f_{\tilde{W}_t}(\xi) - f_{W_t}(\xi) = \Omega(1)$. Otherwise, the *bcd*-parametrization is *perturbation trivial*. ◀

Theorem 6 (**Perturbation nontriviality characterization**)**.** *A stable bcd-parametrization is perturbation nontrivial if and only if* $d + d_{L+1} = 1$ *or* $\min(b_{L+1}, c_{L+1}) + \tilde{r} = 1$.

For the class of stable and perturbation nontrivial *bcd*-parameterizations, SAM learning is both stable and deviates from SGD dynamics. A natural question to ask here is: what should be the ideal SAM behaviour in the infinite-width limit? To address this question, we make the following crucial distinction between *non-vanishing* and *effective perturbations*.

Non-vanishing versus effective perturbations. Recall that the weight perturbation $\boldsymbol{\varepsilon}^l$ perturbs the l-th layer's activations as

$$x^l + \tilde{\delta}x^l = \phi((W^l + \boldsymbol{\varepsilon}^l)(x^{l-1} + \tilde{\delta}x^{l-1})),$$

where $\tilde{\delta}x^l$ denotes the perturbation of the l-th layer's activations accumulated from the weight perturbations $\{\boldsymbol{\varepsilon}^{l'}\}_{l' \in [l]}$ in all previous layers. Therefore, perturbations $\tilde{\delta}x^l$ can stem both from weight perturbations $\boldsymbol{\varepsilon}^{l'}$ in a previous layer $l' < l$ and/or from weight perturbations $\boldsymbol{\varepsilon}^l$ in the current layer l. Intuitively, if we perturb a layer, we want this to affect the next layer's activations and thereby have a nontrivial effect on the output function. Otherwise one can simply set the layer's perturbations to 0 by design and not change the learning algorithm in the infinite-width limit. This motivates the definition of *effective perturbations*, which demands the weight perturbations of the current layer to contribute non-vanishingly. From the weight perspective ($*$), effective l-th layer perturbations are achieved if and only if weight perturbations scale like the weights in spectral norm, $\|\boldsymbol{\varepsilon}^l\|_* / \|W^l\|_* = \Theta(1)$. Without an effective perturbation $\boldsymbol{\varepsilon}^l$ of the l-th layer, this layer does not inherit SAM's inductive bias towards low spectral norm of the Hessian or enhanced sparsity and does not improve generalization performance. We provide empirical evidence for these claims in Appendix H.2. Therefore a distinction between *non-vanishing* and *effective perturbations* is crucial.

Definition 7 (**Non-vanishing perturbations**)**.** For $l \in [L]$, we say that a stable parameterization has *non-vanishing perturbations in the l-th layer* if there exists a $t \in \mathbb{N}$ such that $\tilde{\delta}x^l_t = \Omega(1)$. ◀

Definition 8 (**Effective perturbations**)**.** For $l \in [L+1]$, we say that a stable parameterization *effectively perturbs the l-th layer* if there exists a $t \in \mathbb{N}$ such that $\boldsymbol{\varepsilon}^l_t(x^{l-1}_t + \tilde{\delta}x^{l-1}_t) = \Theta(1)$, where $x^0_t + \tilde{\delta}x^0_t = \xi_t$. ◀

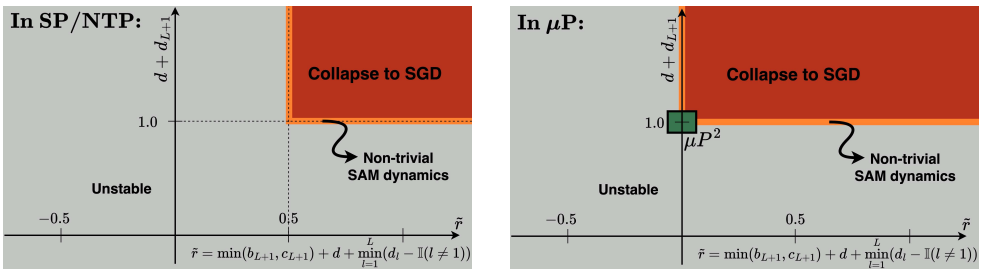


Figure 3: **(Perturbation phase characterization of bcd-parameterizations)** Given a choice of layerwise initialization and learning rate scalings $\{b_l, c_l\}_{l\in[L+1]}$, the maximal feature perturbation scaling $\tilde{r}$ and the last-layer perturbation scaling $d + d_{L+1}$ completely determine whether a *bcd*-parameterization is unstable, has effective SGD dynamics, effective perturbations in some but not all layers or effective perturbations in all layers. In SP or NTP (left), there does not exist a choice of perturbation scalings that achieves effective perturbations in all layers, whereas in μP (right), there is a unique choice as provided in Theorem 11.

Theorem 9 provides a characterization of stable *bcd*-parameterizations with vanishing perturbations in any given layer.

Theorem 9 (Vanishing perturbation characterization). *For any $l_0 \in [L]$, the following statements are equivalent:*

(a) A stable bcd-parametrization has vanishing perturbations in layer l_0.
(b) A stable bcd-parametrization has vanishing perturbations in layer l for all $1 \leq l \leq l_0$.
(c) $\tilde{r}_{l_0} := \min(b_{L+1}, c_{L+1}) + d + \min_{m=1}^{l_0}(d_m - \mathbb{I}(m \neq 1)) > 0$.

It follows from Theorem 9 that any stable *bcd*-parameterization that performs updates in the original gradient direction (i.e., $d_l = C$ for all $l \in [L+1]$ for some $C \in \mathbb{R}$) has vanishing perturbations in all input and hidden layers $l \in [L]$, and the last layer $l = L+1$ is effectively perturbed if and only if $d = 1/2$. This covers the case of both standard and maximal update parameterizations with global scaling of the perturbation radius discussed in Section 3. Negating the conditions of Theorem 9 implies that a stable *bcd*-parameterization has non-vanishing perturbations in layer l_0 if and only if $\tilde{r}_{l_0} = 0$. Achieving *effective perturbations* is a stronger requirement for which Theorem 10 provides the necessary and sufficient conditions.

Theorem 10 (Effective perturbation characterization). *For $l \in [L]$, a stable bcd-parametrization effectively perturbs the l-th layer if and only if $\min(b_{L+1}, c_{L+1}) + d + d_l - \mathbb{I}(l \neq 1) = 0$.*

A stable bcd-parametrization effectively perturbs the last layer if and only if $d + d_{L+1} = 1$.

4.2 Maximal Update and Perturbation Parameterization (μP^2)

We postulate that just as the optimal learning rate transfers across widths under μP for SGD and Adam due to non-vanishing width-independent feature evolution in all layers, the optimal learning rate and perturbation radius may be jointly transferable across widths if additionally the weight perturbations induce width-independent perturbations of the activations in all layers. Here, we show that, for every stable initialization and learning rate scaling with $b_{L+1} \geq 1$, there exists a unique stable layerwise perturbation scaling which effectively perturbs every single layer. We term this layerwise perturbation scaling $\{d_l\}_{l\in[L+1]} \cup \{d\}$ the Maximal Perturbation Parameterization (MPP). This concludes the phase characterization of perturbation scaling behaviours (Figure 3).

Theorem 11 (Maximal Perturbation Parameterization (MPP)). *Consider any stable bcd-parametrization $\{b_l\}_{l\in[L+1]} \cup \{c_l\}_{l\in[L+1]} \cup \{d_l\}_{l\in[L+1]} \cup \{d\}$. If $b_{L+1} < 1$, then there does not exist a stable choice of $\{d_l\}_{l\in[L+1]} \cup \{d\}$ that achieves effective perturbations before the last layer. If $b_{L+1} \geq 1$, then up to the equivalence $d_l' = d_l + C$, $C \in \mathbb{R}$, $\forall l \in [L+1]$, the unique stable choice $\{d_l\}_{l\in[L+1]} \cup \{d\}$ that effectively perturbs all layers $l \in [L+1]$ is given by*

$$d = -1/2, \qquad d_l = \begin{cases} 1/2 - \min(b_{L+1}, c_{L+1}) & l = 1, \\ 3/2 - \min(b_{L+1}, c_{L+1}) & l \in [2, L], \\ 3/2 & l = L+1. \end{cases} \tag{1}$$

Maximal Update and Perturbation Parameterization μP^2. To achieve feature learning in every layer and hyperparameter transfer in the learning rate, μP is the unique[2] choice of layerwise initialization variance and learning rate scalings $\{b_l, c_l\}_{l\in[L+1]}$ (Yang and Hu, 2021). Together with Theorem 11, this shows that there exists a *unique*[2] bcd-parameterization that achieves both feature learning and effective perturbations in all layers, we call *maximal update and perturbation parametrization*, μP^2 for short. Now that we have found a parameterization that achieves width-independent scaling of both activation updates and activation perturbations, μP^2 fulfills essential necessary conditions for hyperparameter transfer to occur in both η and ρ.

Remark 12 (Achieving μP^2 with weight multipliers). Appendix F.6 covers the extension of our results to nontrivial weight multipliers. We show that, for each choice of weight multipliers $\{a_l\}_{l\in[L+1]}$, there is a unique[2] choice of bcd-hyperparameters that achieves effective perturbations in all layers. But unlike for SGD or Adam, these parameterizations lead to slightly different training algorithms, because differing subsets of layers contribute non-vanishingly to the joint gradient normalization term $\|\nabla_{\mathbf{W}}\mathcal{L}\|_F$ in (SAM). The term $\|\nabla_{\mathbf{W}}\mathcal{L}\|_F$ couples all layers so that there do not exist layerwise but only layer-coupled equivalence classes for (SAM). Most importantly, **instead of adapting (SAM), we can adapt the architecture with the weight multipliers $n^{-a_l}\cdot W^l$ with**

$$a_l = -1/2\cdot\mathbb{I}(l=1) + 1/2\cdot\mathbb{I}(l=L+1) \qquad (a\text{-}\mu P^2)$$

to achieve effective perturbations in all layers with naive perturbation and learning rate scaling such that all layers contribute non-vanishingly to the joint gradient norm (Appendix F.6). One downside of (a-μP^2), that also applies to naive weight multipliers $a_l = 0$, is its incompatibility with unit scaling considerations for low precision training (Blake et al., 2024). ◀

Alternative perturbation scaling definitions. Scaling equivalent to (a-μP^2) can be achieved without multipliers by scaling the numerator and denominator terms in (LP) independently, and choosing to scale all denominator terms to be width-independent (see perturbation rule (DP) and Appendix F.7 for more details). The ablations in Appendix H.4 suggest that this has a negligible effect on the optimal generalization performance of μP^2, but can be more stable given suboptimal hyperparameters. Gradient normalization in each layer separately is uncommon and performs slightly worse (Appendix H.5). Appendix F.2 discusses further considerations that led to Definition 4.

Trivial, lazy, and feature learning regimes. A small last-layer initialization variance $b_{L+1} \geq 1$ is required for stable feature learning. Theorem 11 shows that $b_{L+1} \geq 1$ is also required for effective hidden-layer perturbations. Beyond this condition, the choice of $\{b_l\}$ and $\{c_l\}$ is decoupled from that of perturbation scalings $\{d_l\}\cup\{d\}$ for stable bcd-parameterizations, because the scale of the activations of a layer l is entirely determined by the scale of initialization b_l and learning rates c_l, given stability. Consequently, whether a parameterization is *trivial*, in the *lazy regime,* or in the *feature learning regime* is independent of the choice of d_l's provided that all stability constraints are met. A complete characterization of these regimes for the class of bc-parameterizations has been provided in Yang and Hu (2021) and remains unchanged for the class of stable bcd-parameterizations. For completeness, formal definitions and the corresponding results are stated in Appendices C and D.

4.3 Generalizations to other architectures and SAM variants

Generalization to other architectures. Our results can be extended to other common layer types, that are representable as a NE⊗OR⊤program, including all ResNet and Transformer components (Appendix F.5). All considered layer types behave like input, hidden or output layers. Most importantly, normalization layer weights and biases scale like input layer weights to the input 1.

Generalization to other SAM variants. We would like to find the correct layerwise perturbation scaling without writing out the NE⊗OR⊤program for every perturbation rule individually. Formally justified by our proof in Appendix E, we rephrase our equivalent spectral scaling condition ($*$) from Section 3 to: maximal stable perturbations are achieved in μP if and only if $\boldsymbol{\varepsilon}^l = \Theta(\delta W^l)$. This condition holds as soon as weight updates δW^l and perturbations $\boldsymbol{\varepsilon}^l$ are both correlated with the incoming activations x^{l-1}, for example if both are gradient-based. Table 1 summarizes the application of this condition to two ASAM variants that perform well empirically but cannot be written as a NE⊗OR⊤program. Additional details are provided in Appendix F.4. We demonstrate that these scalings perform well and transfer hyperparameters in the next section. Note that for hidden layers in

[2]Strictly speaking, unique up to smaller last-layer initialization $b_{L+1} \geq 1$.

	Perturbed under global scaling?			**For effective perturbations with μP^2:**			
	Input, biases, norm.	Other hidden layers	Output layer	Global ρ	Input-like	Hidden-like	Output-like
SAM	✗	✗	✓	$n^{1/2}$	$n^{1/2}$	$n^{-1/2}$	$n^{-3/2}$
Layer. ASAM	✗	✓	✗	1	1	n^{-1}	1
Elem. ASAM	✓	✓	✓	$n^{1/2}$	1	1	1
SAM-ON	✓	-	-	$n^{1/2}$	1	-	-

Table 1: **(Layerwise perturbation scaling for effective perturbations in μP)** Without layerwise perturbation scaling (*left*), each SAM variant perturbs a different subset of layers at large width $n \to \infty$, but we provide the unique layerwise perturbation rescaling μP^2 (*right*) that achieves effective perturbations in all layers. This parameterization transfers both the optimal η and ρ across widths.

μP^2, it holds that $\varepsilon^l = \Theta(n^{-1})$ but $W^l = \Theta(n^{-1/2})$ entrywise, due to large initialization, showing that it is crucial to compare perturbations to updates or to measure weight scalings in spectral norm.

5 The maximal update and perturbation parameterization μP^2 achieves hyperparameter transfer and improved generalization

In this section, we provide experimental results showing that μP^2 achieves hyperparameter transfer in both η and ρ across architectures, and that μP^2 also improves generalization over SP and μP with global perturbations – even after multi-epoch training to convergence. We train MLPs and ResNets (He et al., 2016) on CIFAR10 (Krizhevsky et al., 2009) and Vision Transformers (ViTs) (Dosovitskiy et al., 2021) on Imagenet1K (Deng et al., 2009). While we directly implement *bcd*-parameterizations for MLPs and ResNets in PyTorch (Paszke et al., 2019), we use the `mup`-package (Yang et al., 2022) as a basis for ViT experiments. Pseudocode and a spectral derivation of our μP^2-implementation for ViTs, which is equivalent to (a-μP^2), are provided in Appendix F.7. All experimental details are stated in Appendix G and all supplemental experiments can be found in Appendix H.

Comparing candidate parameterizations in MLPs. Figure 1 shows test accuracy as a function of learning rate and perturbation radius for MLPs of varying width. While previous μP-literature mostly focuses on the more immediate transfer in training error, for SAM it is crucial to consider optimality in test error as the perturbation radius acts as a regularizer, so that optimality in test error typically coincides with suboptimal training error. In μP without perturbation scaling, the regime of stable perturbation radii shrinks (Figure H.6). In μP with global perturbation scaling $\rho \cdot n^{-1/2}$, the regime of stable ρ remains invariant under width scaling, but there is no significant improvement of SAM beyond SGD, so that the optimal perturbation radius fluctuates within its stable regime due to noise. Only μP^2 consistently achieves hyperparameter transfer across widths, and achieves significant improvement over its base optimizer SGD in μP at scale. The full hyperparameter landscapes are provided in Appendix H.3.

ρ-transfer in ViTs. Figure 4 shows that the optimal perturbation radius transfers for ViT-S/16 on Imagenet1K trained with SAM in μP^2. While Andriushchenko and Flammarion (2022, Appendix E.3) observe diminishing benefits of SAM at large widths in SP, here the improvements beyond the base optimizer AdamW in μP are particularly large.

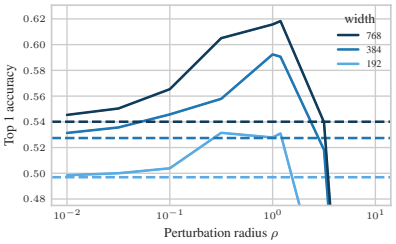


Figure 4: **(ρ-transfer in ViTs)** Training a ViT with SAM in μP^2 on ImageNet1K from scratch for 100 epochs yields ρ-transfer and large improvements over AdamW in μP (dashed lines).

ρ-transfer for SAM variants in μP^2. Figure 6 shows that training a ResNet-18 in μP^2 achieves hyperparameter transfer in ρ for all considered SAM variants with varying width. μP with global perturbation scaling (μP-global) has a width-invariant stability threshold in ρ and the optimal ρ clearly shifts toward that threshold. It would be interesting to see whether this shift continues with larger width and leads to suboptimal performance of μP-global in wider ResNets. Table 2 shows that all SAM variants perform similarly well in μP^2, some slightly outperforming the best-

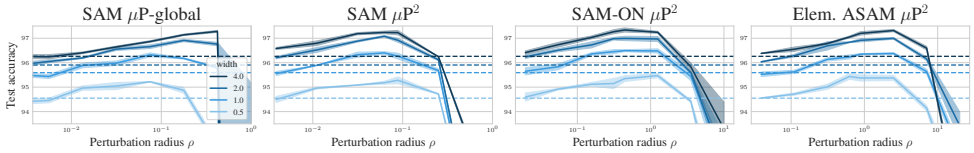


Figure 6: (**ρ-transfer of ASAM variants in μP^2**) Test error as a function of perturbation radius ρ after 200 epochs of training a ResNet-18 in μP^2 on CIFAR10 with various SAM variants (see subplot title). CI over 2 independent runs. Darker lines correspond to larger width multipliers. Other hyperparameters are tuned at base width multiplier 0.5. μP^2 achieves transfer in ρ and large improvements over the base optimizer (dashed lines) SGD in μP with momentum and weight decay.

	SAM global	SAM μP^2	SAM-ON μP^2	Elem. ASAM μP^2
SP	$97.00_{\pm 0.03}(+0.96)$	$97.00_{\pm 0.03}(+0.96)$	$\mathbf{97.29}_{\pm 0.06}(\mathbf{+1.26})$	$97.15_{\pm 0.01}(+1.11)$
μP	$\mathbf{97.19}_{\pm 0.05}(\mathbf{+0.93})$	$\mathbf{97.23}_{\pm 0.08}(\mathbf{+0.97})$	$\mathbf{97.34}_{\pm 0.08}(\mathbf{+1.08})$	$\mathbf{97.32}_{\pm 0.05}(\mathbf{+1.06})$

Table 2: (**Performance of μP^2**) Average test accuracy$_{\pm \text{standard deviation across 4 runs}}$ (+ improvement of SAM over SGD) for ResNet-18 with width multiplier 4 on CIFAR10 using SGD as a base optimizer. In bold, all parameterizations within a 2σ-CI from the best-performing variant SAM-ON in μP^2.

performing variant SAM-ON in SP. This suggests that for ResNets, even with a proper layerwise balance, normalization layer perturbations may suffice, and performance differences in SP are primarily caused by varying degrees to which the normalization layers are perturbed.

Without providing an explanation, Müller et al. (2024, Section 5.3) observe that only SAM-ON and elementwise ASAM sufficiently perturb normalization layers in SP. Table 1 (*left*) explains these observations by showing that only these two SAM variants effectively perturb normalization layers under global perturbation scaling. Table 1 (*right*) also provides full control over which layers to perturb. For transferring the optimal ρ with SAM-ON in μP, our theory predicts the global scaling $\rho = \Theta(n^{1/2})$ which is confirmed by our empirical observations (Figure 6). However, properly understanding the role of normalization layer perturbations remains an important question for future work. Note that we report results after fine-tuning all hyperparameters. The performance gain of μP^2 over SP and μP-global is likely much higher in larger models, for which fine-tuning is infeasible and the lack of feature learning and effective perturbations is more pronounced. Even under optimal HPs, μP^2 appears to stabilize SAM's training dynamics compared to SP (Figure 5).

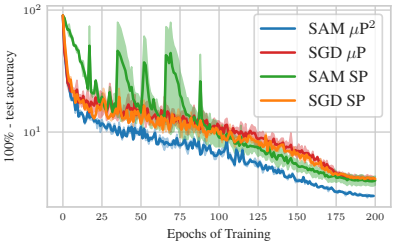


Figure 5: **(Stable training dynamics)** SAM in μP^2 stabilizes training dynamics for a ResNet-18 with width multiplier 2.

6 Future work

This study may serve as an inspiration of how scaling theory can be used to understand and improve training procedures in minimax optimization and beyond. To reach a fully practical theory of deep learning, it will be necessary to take data distributions and training dynamics into account in more detail than it is possible with current Tensor Program theory (Everett et al., 2024). Existing Tensor Program theory assumes constant batch size and training time, and does not make statements about generalization. For example, we observe that MLPs and ResNets in SP can sometimes display HP transfer in η and ρ after multi-epoch training to convergence (Appendix H.3.2). This goes beyond the observations by Everett et al. (2024) as we observe transfer even without tuning layerwise learning rates or weight multipliers. This transfer strongly contradicts the infinite-width theory from Yang and Hu (2021) which predicts output blowup under large learning rates, and it shows that the exact conditions which enable hyperparameter transfer in practice are not fully understood. It also remains unclear how to optimally adapt (SAM) when increasing network depth. We plan to address some of these questions in upcoming work.

References

Maksym Andriushchenko and Nicolas Flammarion. Towards understanding sharpness-aware minimization. In *International Conference on Machine Learning (ICML)*, 2022. Cited on page 5, 9, 17, 37, 38, 48.

Maksym Andriushchenko, Dara Bahri, Hossein Mobahi, and Nicolas Flammarion. Sharpness-aware minimization leads to low-rank features. *arXiv:2305.16292*, 2023a. Cited on page 38, 53.

Maksym Andriushchenko, Francesco Croce, Maximilian Müller, Matthias Hein, and Nicolas Flammarion. A modern look at the relationship between sharpness and generalization. In Andreas Krause, Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato, and Jonathan Scarlett, editors, *Proceedings of the 40th International Conference on Machine Learning (ICML)*, 2023b. Cited on page 3, 17.

Sanjeev Arora, Zhiyuan Li, and Abhishek Panigrahi. Understanding gradient descent on the edge of stability in deep learning. In Kamalika Chaudhuri, Stefanie Jegelka, Le Song, Csaba Szepesvari, Gang Niu, and Sivan Sabato, editors, *Proceedings of the 39th International Conference on Machine Learning (ICML)*, 2022. Cited on page 17.

Peter L. Bartlett, Philip M. Long, and Olivier Bousquet. The dynamics of sharpness-aware minimization: Bouncing across ravines and drifting towards wide minima. *Journal of Machine Learning Research (JMLR)*, 24(316):1–36, 2023. Cited on page 3, 17.

Jeremy Bernstein, Arash Vahdat, Yisong Yue, and Ming-Yu Liu. On the distance between two neural networks and the stability of learning. *Advances in Neural Information Processing Systems (NeurIPS)*, 33, 2020. Cited on page 17.

Tamay Besiroglu, Ege Erdil, Matthew Barnett, and Josh You. Chinchilla scaling: A replication attempt. *arXiv:2404.10102*, 2024. Cited on page 17.

Charlie Blake, Constantin Eichenberg, Josef Dean, Lukas Balles, Luke Y Prince, Björn Deiseroth, Andres Felipe Cruz-Salinas, Carlo Luschi, Samuel Weinbach, and Douglas Orr. u-μP: The unit-scaled maximal update parametrization. *arXiv:2407.17465*, 2024. Cited on page 5, 8, 45, 49.

Blake Bordelon, Lorenzo Noci, Mufan Bill Li, Boris Hanin, and Cengiz Pehlevan. Depthwise hyperparameter transfer in residual networks: Dynamics and scaling limit. *arXiv:2309.16620*, 2023. Cited on page 3, 16.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in Neural Information Processing Systems (NeurIPS)*, 33:1877–1901, 2020. Cited on page 60.

Xiangning Chen, Cho-Jui Hsieh, and Boqing Gong. When vision transformers outperform resnets without pre-training or strong data augmentations. *arXiv:2106.01548*, 2021. Cited on page 1, 17.

Jeremy Cohen, Simran Kaur, Yuanzhi Li, J Zico Kolter, and Ameet Talwalkar. Gradient descent on neural networks typically occurs at the edge of stability. In *International Conference on Learning Representations (ICLR)*, 2020. Cited on page 17.

Yan Dai, Kwangjun Ahn, and Suvrit Sra. The crucial role of normalization in sharpness-aware minimization. *Advances in Neural Information Processing Systems (NeurIPS)*, 36, 2024. Cited on page 17, 39.

Yann N Dauphin, Atish Agarwala, and Hossein Mobahi. Neglected hessian component explains mysteries in sharpness regularization. *arXiv:2401.10809*, 2024. Cited on page 17.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2009. Cited on page 9, 51, 71.

Laurent Dinh, Razvan Pascanu, Samy Bengio, and Yoshua Bengio. Sharp minima can generalize for deep nets. In *Proceedings of the 34th International Conference on Machine Learning (ICML)*, 2017. Cited on page 17.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations (ICLR)*, 2021. Cited on page 9.

Katie E Everett, Lechao Xiao, Mitchell Wortsman, Alexander A Alemi, Roman Novak, Peter J Liu, Izzeddin Gur, Jascha Sohl-Dickstein, Leslie Pack Kaelbling, Jaehoon Lee, and Jeffrey Pennington. Scaling exponents across parameterizations and optimizers. In *Proceedings of the 41st International Conference on Machine Learning (ICML)*, 2024. Cited on page 10, 49, 61.

Pierre Foret, Ariel Kleiner, Hossein Mobahi, and Behnam Neyshabur. Sharpness-aware minimization for efficiently improving generalization. In *International Conference on Learning Representations (ICLR)*, 2021. Cited on page 1, 5, 17, 37, 48.

Boris Hanin and David Rolnick. How to start training: The effect of initialization and architecture. *Advances in Neural Information Processing Systems (NeurIPS)*, 31, 2018. Cited on page 16.

Soufiane Hayou and Greg Yang. Width and depth limits commute in residual networks. In *International Conference on Machine Learning (ICML)*, 2023. Cited on page 17.

Soufiane Hayou, Eugenio Clerico, Bobby He, George Deligiannidis, Arnaud Doucet, and Judith Rousseau. Stable resnet. In *International Conference on Artificial Intelligence and Statistics*, 2021. Cited on page 3, 16.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *IEEE international conference on computer vision (ICCV)*, 2015. Cited on page 2, 51.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. Cited on page 9.

Sepp Hochreiter and Jürgen Schmidhuber. Flat minima. *Neural Comput.*, 9(1):1–42, 1997. Cited on page 17.

Jordan Hoffmann, Sebastian Borgeaud, Arthur Mensch, Elena Buchatskaya, Trevor Cai, Eliza Rutherford, Diego de Las Casas, Lisa Anne Hendricks, Johannes Welbl, Aidan Clark, et al. Training compute-optimal large language models. *arXiv:2203.15556*, 2022. Cited on page 17.

Satoki Ishikawa and Ryo Karakida. On the parameterization of second-order optimization effective towards the infinite width. *arXiv preprint arXiv:2312.12226*, 2023. Cited on page 17.

Arthur Jacot, Franck Gabriel, and Clément Hongler. Neural Tangent Kernel: Convergence and generalization in neural networks. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2018. Cited on page 16.

Yiding Jiang*, Behnam Neyshabur*, Hossein Mobahi, Dilip Krishnan, and Samy Bengio. Fantastic generalization measures and where to find them. In *International Conference on Learning Representations (ICLR)*, 2020. Cited on page 17.

Jean Kaddour, Linqing Liu, Ricardo Silva, and Matt J Kusner. When do flat minima optimizers work? *Advances in Neural Information Processing Systems (NeurIPS)*, 35, 2022. Cited on page 1, 17.

Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. Scaling laws for neural language models. *arXiv:2001.08361*, 2020. Cited on page 2, 17.

Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009. Cited on page 9, 51, 71.

Jungmin Kwon, Jeongseop Kim, Hyunseo Park, and In Kwon Choi. Asam: Adaptive sharpness-aware minimization for scale-invariant learning of deep neural networks. In *International Conference on Machine Learning (ICML)*, 2021. Cited on page 1, 3, 17, 37, 39, 48.

Yann LeCun, Léon Bottou, Genevieve B Orr, and Klaus-Robert Müller. Efficient backprop. In *Neural networks: Tricks of the trade*. Springer, 2002. Cited on page 2.

Mufan Li, Mihai Nica, and Dan Roy. The future is log-gaussian: Resnets and their infinite-depth-and-width limit at initialization. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2021. Cited on page 3, 16.

Yong Liu, Siqi Mai, Xiangning Chen, Cho-Jui Hsieh, and Yang You. Towards efficient and scalable sharpness-aware minimization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022. Cited on page 3, 17.

Philip M. Long and Peter L. Bartlett. Sharpness-aware minimization and the edge of stability. *arXiv:2309.12488*, 2023. Cited on page 17.

Song Mei, Andrea Montanari, and Phan-Minh Nguyen. A mean field view of the landscape of two-layer neural networks. *Proceedings of the National Academy of Sciences*, 115(33):E7665–E7671, 2018. Cited on page 16.

Enea Monzio Compagnoni, Luca Biggio, Antonio Orvieto, Frank Norbert Proske, Hans Kersting, and Aurelien Lucchi. An SDE for modeling SAM: Theory and insights. In Andreas Krause, Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato, and Jonathan Scarlett, editors, *Proceedings of the 40th International Conference on Machine Learning (ICML)*, 2023. Cited on page 3, 17, 39.

Maximilian Müller, Tiffany Vlaar, David Rolnick, and Matthias Hein. Normalization layers are all that sharpness-aware minimization needs. *Advances in Neural Information Processing Systems (NeurIPS)*, 36, 2024. Cited on page 1, 3, 10, 17, 35, 37, 39, 41, 48, 51, 52.

Radford M. Neal. *Priors for Infinite Networks*. Springer New York, 1996. Cited on page 16.

Lorenzo Noci, Sotiris Anagnostidis, Luca Biggio, Antonio Orvieto, Sidak Pal Singh, and Aurelien Lucchi. Signal propagation in transformers: Theoretical perspectives and the role of rank collapse. *Advances in Neural Information Processing Systems (NeurIPS)*, 35, 2022. Cited on page 3, 16.

Lorenzo Noci, Chuning Li, Mufan Li, Bobby He, Thomas Hofmann, Chris J Maddison, and Dan Roy. The shaped transformer: Attention models in the infinite depth-and-width limit. *Advances in Neural Information Processing Systems (NeurIPS)*, 36, 2024. Cited on page 3, 16.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2019. Cited on page 9, 71.

Ben Poole, Subhaneil Lahiri, Maithra Raghu, Jascha Sohl-Dickstein, and Surya Ganguli. Exponential expressivity in deep neural networks through transient chaos. *Advances in Neural Information Processing Systems (NeurIPS)*, 29, 2016. Cited on page 16.

David Samuel. (Adaptive) SAM Optimizer (PyTorch). `https://github.com/davda54/sam`, 2022. Cited on page 37, 48, 49, 71.

Samuel S Schoenholz, Justin Gilmer, Surya Ganguli, and Jascha Sohl-Dickstein. Deep information propagation. *arXiv:1611.01232*, 2016. Cited on page 16.

Sungbin Shin, Dongyeop Lee, Maksym Andriushchenko, and Namhoon Lee. The effects of overparameterization on sharpness-aware minimization: An empirical and theoretical analysis. *arXiv:2311.17539*, 2023. Cited on page 17.

Leena Chennuru Vankadara, Jin Xu, Moritz Haas, and Volkan Cevher. On feature learning in structured state space models. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems (NeurIPS)*, 2024. Cited on page 17.

Nikhil Vyas, Alexander Atanasov, Blake Bordelon, Depen Morwani, Sabarish Sainathan, and Cengiz Pehlevan. Feature-learning networks are consistent across widths at realistic scales. *Advances in Neural Information Processing Systems (NeurIPS)*, 36, 2024. Cited on page 2, 16, 60.

Kaiyue Wen, Tengyu Ma, and Zhiyuan Li. How sharpness-aware minimization minimizes sharpness? In *The Eleventh International Conference on Learning Representations (ICLR)*, 2023. Cited on page 3, 17.

Kaiyue Wen, Zhiyuan Li, and Tengyu Ma. Sharpness minimization algorithms do not only minimize sharpness to achieve better generalization. *Advances in Neural Information Processing Systems (NeurIPS)*, 36, 2024. Cited on page 3, 17.

Jonathan Wenger, Felix Dangel, and Agustinus Kristiadi. On the disconnect between theory and practice of overparametrized neural networks. *arXiv:2310.00137*, 2023. Cited on page 16.

Bichen Wu, Chenfeng Xu, Xiaoliang Dai, Alvin Wan, Peizhao Zhang, Zhicheng Yan, Masayoshi Tomizuka, Joseph Gonzalez, Kurt Keutzer, and Peter Vajda. Visual transformers: Token-based image representation and processing for computer vision. *arXiv:2006.03677*, 2020. Cited on page 51.

Lechao Xiao, Jeffrey Pennington, and Samuel Schoenholz. Disentangling trainability and generalization in deep neural networks. In *International Conference on Machine Learning (ICML)*, 2020. Cited on page 16.

Greg Yang. Wide feedforward or recurrent neural networks of any architecture are gaussian processes. *Advances in Neural Information Processing Systems (NeurIPS)*, 32, 2019. Cited on page 2, 3, 16, 42.

Greg Yang. Tensor programs iii: Neural matrix laws. *arXiv:2009.10685*, 2021. Cited on page 24.

Greg Yang and Edward J. Hu. Tensor programs iv: Feature learning in infinite-width neural networks. In *International Conference on Machine Learning (ICML)*, 2021. Cited on page 2, 3, 5, 6, 8, 10, 16, 18, 20, 21, 29, 31, 32, 33, 41.

Greg Yang and Etai Littwin. Tensor programs ivb: Adaptive optimization in the infinite-width limit. *arXiv:2308.01814*, 2023. Cited on page 2, 3, 16, 17, 19, 41.

Greg Yang, Edward J Hu, Igor Babuschkin, Szymon Sidor, Xiaodong Liu, David Farhi, Nick Ryder, Jakub Pachocki, Weizhu Chen, and Jianfeng Gao. Tensor programs v: Tuning large neural networks via zero-shot hyperparameter transfer. *arXiv:2203.03466*, 2022. Cited on page 2, 3, 9, 16, 19, 43, 51, 57, 60, 61, 71.

Greg Yang, James B. Simon, and Jeremy Bernstein. A spectral condition for feature learning. *arXiv:2310.17813*, 2023a. Cited on page 5, 17, 39, 47.

Greg Yang, Dingli Yu, Chen Zhu, and Soufiane Hayou. Tensor programs vi: Feature learning in infinite-depth neural networks. *arXiv:2310.02244*, 2023b. Cited on page 3, 5, 16, 17, 43.

Yang You, Jing Li, Sashank Reddi, Jonathan Hseu, Sanjiv Kumar, Srinadh Bhojanapalli, Xiaodan Song, James Demmel, Kurt Keutzer, and Cho-Jui Hsieh. Large batch optimization for deep learning: Training bert in 76 minutes. In *International Conference on Learning Representations (ICLR)*, 2020. Cited on page 17.

Xiaohua Zhai, Alexander Kolesnikov, Neil Houlsby, and Lucas Beyer. Scaling vision transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022. Cited on page 2, 17.

Appendices

Appendix Contents.

A Notation **16**

B Detailed related work **16**

C Definitions **18**

D Extensive main results **19**

E Proof of main results **23**
E.1 Tensor program formulation 23
E.2 The infinite-width limit 29
E.3 Concluding the proof of all main results 31
E.4 Analytic expression of the features after first SAM update 33

F Generalizations and further perturbation scaling considerations **35**
F.1 Overview over choices of d_l and d 35
F.2 Other ways to introduce layerwise perturbation scaling 37
F.3 Extension to SAM without gradient normalization 38
F.4 Extension to Adaptive SAM 39
F.5 Representing general architectures and adaptive optimizers as Tensor Programs . . 41
F.6 Influence of width-dependent weight multipliers on bcd-parameterizations 43
F.7 The spectral perspective on μP^2 47

G Experimental details **51**

H Supplemental experiments **53**
H.1 SAM is approximately LL-SAM in μP with global perturbation scaling 53
H.2 Propagating perturbations from the first layer does not inherit SAM's benefits . . . 54
H.3 Hyperparameter transfer 57
H.4 Gradient norm contributions have negligible effects on generalization performance 64
H.5 SAM with layerwise gradient normalization 65
H.6 Test error over the course of training 66

A Notation

Symbol	Meaning
n, η, ρ	width, learning rate, perturbation radius
$\phi, \mathcal{L}, (\xi_t, y_t)$	activation function, loss function, input and label at time t
$\|v\| := \|v\|_2, \|W\| := \|W\|_F$	2-norm as standard for vectors, Frobenius norm as standard for matrices
$\|W\|_*$	spectral norm for matrices (also called operator norm)
W_t^l	trainable weights at time t in layer l
δW_t^l	weight updates at time t in layer l
$\boldsymbol{\varepsilon}_t^l$	weight perturbations at time t in layer l
$\|v_t\|$	norm of the rescaled gradient in the perturbation denominator
h_t^l, x_t^l	preactivations and activations at time t in layer l
δx_t^l	activation updates at time t in layer l
$\tilde{\delta} x_t^l$	activation perturbations at time t in layer l
$\delta f_t, \tilde{\delta} f_t$	update/perturbation of the output function at time t
$\chi_t = \mathcal{L}'(f_t(\xi_t), y_t)$	derivative of loss w.r.t. output function at time t
$\tilde{\delta} W_t^l = \boldsymbol{\varepsilon}_t^l$	weight perturbations at time t in layer l (with $\tilde{\delta}$ for consistency)
$\odot$	elementwise multiplication
$dz_t = \theta_\nabla^{-1} \nabla_z f$	derivative of output function w.r.t. $z \in \{h^l, x^l\}$ at time t, normalized to $\Theta(1)$
$dz_{SAM,t}$	derivative of perturbed output function w.r.t. perturbed $z \in \{\tilde{h}^l, \tilde{x}^l\}$ at time t, normalized to $\Theta(1)$
θ_∇	scaling of the activation gradients
$\theta_l, \tilde{\theta}_l$	update and perturbation scaling of h_t^l and x_t^l
$\theta_{W^l}, \tilde{\theta}_{W^l}$	update and perturbation scaling of W_t^l
$\mathring{\theta}$	limit scaling; under stability, all considered scalings $\mathring{\theta} \in \{0, 1\}$
Z^z	random variable distributed according to the limiting distribution for the entries of the TP vector z specified by the TP Master Theorem

Table A.1: **(Notation)** Overview over notation used in the main paper (top) and in the appendix (bottom).

B Detailed related work

Signal propagation. Our work can be seen as scaling theory with the goal of preventing both vanishing and exploding signals in forward and backward passes, where the analysis of SAM requires considering stability of perturbations in each layer as well. In this sense, we build on a rich literature, often restricted to an analysis at initialization (Schoenholz et al., 2016; Poole et al., 2016; Hanin and Rolnick, 2018; Xiao et al., 2020). For scaling neural networks to infinite depth, residual connections have been found to be beneficial for stabilizing signal propagation while retaining expressivity. The simple $\frac{1}{\sqrt{L}}$-scaling allows depth-scaling in ResNets and unlocks hyperparameter transfer (Hayou et al., 2021; Li et al., 2021; Bordelon et al., 2023; Yang et al., 2023b). Noci et al. (2022, 2024) provide infinite width and depth analyses for Transformers with the goal of preventing rank collapse and attaining a limit that has behaviour consistent with that of moderately large networks.

Tensor Programs. After kernel-based approaches to understand infinite-width limits of neural networks (Neal, 1996; Jacot et al., 2018) and applications of mean-field theory (Mei et al., 2018), the Tensor Program series (Yang, 2019; Yang and Hu, 2021; Yang and Littwin, 2023; Yang et al., 2022, 2023b) marks the first important break through in the theory of large neural networks. The framework covers many modern deep learning architectures, optimization algorithms and arbitrary abc-parameterizations, where each abc-parameterization is essentially defined by a layerwise scaling of initialization variance and learning rate as a function of network width. Yang and Hu (2021) propose the *maximal update parameterization* (μP) and show that it is the unique stable parameterization that achieves feature learning in all layers in the limit of infinite width. In this framework, training neural networks with a global learning rate $\eta > 0$ for all layers and with He or LeCun initialization falls under the category of so called *standard parameterization* (SP). The neural tangent parameterization (NTP), studied in the neural tangent kernel literature, differs but does not achieve feature learning in any layer, and is therefore less useful to describe the behaviour of finite width networks than μP (Wenger et al., 2023; Vyas et al., 2024). Yang and Littwin (2023) characterize stable learning with adaptive optimizers at infinite width into a feature learning versus a (nonlinear) operator regime. SAM

is not covered by the update rule definition in Yang and Littwin (2023) since the nested application of the gradient w.r.t. the weights is not a coordinatewise optimizer anymore. Yang et al. (2023a) show that μP is equivalent to the spectral scaling conditions on the weights $\|\Delta W^l\| = \Theta(\sqrt{n_l/n_{l-1}})$ and $\|\Delta W^l\| = \Theta(\sqrt{n_l/n_{l-1}})$. Hence Bernstein et al. (2020) would have achieved their goal of an optimizer with automatic update scaling, if they had normalized by the spectral instead of the Frobenius norm and multiplied by $\sqrt{\texttt{fan\_out}/\texttt{fan\_in}}$ in each layer. While recent works have considered joint limits of infinite width and depth (Yang et al., 2023b; Hayou and Yang, 2023), the data distribution has not been taken into account in Tensor Program literature. The study of scaling laws of jointly scaling model size, data set size and training time has predominantly been empirical (Kaplan et al., 2020; Zhai et al., 2022; Hoffmann et al., 2022; Besiroglu et al., 2024). Developing theory to inform Pareto optimal trade offs in a principled manner constitutes an important direction for future work.

As an example of scaling theory for second order optimization, Ishikawa and Karakida (2023) derive μP for KFAC and Shampoo. This scaling rule differs from μP for SGD. Similarly, Vankadara et al. (2024) show that maximal updates are achieved by another different scaling rule for non-standard architectures like structured state space models.

Sharpness Aware Minimization. Sharpness aware minimization (SAM) (Foret et al., 2021) has shown to be extremely effective and robust in improving generalization performance across a wide range of architectures and settings (Chen et al., 2021; Kaddour et al., 2022). SAM was motivated as an inductive bias towards flatter minima and it has been understood to have an gradient-norm adaptive edge of stability at which it drifts towards minima with smaller spectral norm of the Hessian (Long and Bartlett, 2023; Bartlett et al., 2023). However a full understanding of why SAM works so well remains elusive. While correlations between flatness and generalization have been observed in some settings (Hochreiter and Schmidhuber, 1997; Jiang* et al., 2020), other studies have questioned the usefulness of sharpness as a measure for generalization, especially for modern architectures (Dinh et al., 2017; Andriushchenko et al., 2023b; Wen et al., 2024). Applying SAM on only the normalization layers often even improves generalization in vision tasks depsite increasing sharpness (Müller et al., 2024). Adaptive SAM (ASAM) (Kwon et al., 2021) is a variant of SAM derived from a sharpness definition that is invariant to weight rescalings with respect to a chosen normalization operator that leave the output function invariant. The results in Müller et al. (2024) suggest that two of the most promising normalization operators are elementwise normalization $T^l_w(x) = |W^l| \odot x$ and layerwise normalization $T^l_w(x) = \|W^l\|_F \cdot x$. We state the resulting update rules and a scaling analysis in Appendix F.4. A variant of SAM that is often studied theoretically because of its simplicity does not normalize the gradient of the perturbation. Our theory covers this variant too (Appendix F.3), but Dai et al. (2024) argue that normalizing the gradients for the perturbation is crucial. Monzio Compagnoni et al. (2023) find that unnormalized SAM gets stuck around saddles while SAM slowly escapes through additional Hessian-induced noise. This suggests that the additional effort of analysing the original SAM update rule with gradient normalization is necessary for practically useful theory. Dauphin et al. (2024) draw connections between SAM and other second order optimizers like gradient penalties and weight noise. They show that SAM is able to effectively use second order information implicitly using ReLU, whereas the other two methods close the gap to SAM when using GeLU since they require the localized second order information that GeLU provides in contrast to ReLU. Wen et al. (2023) show that worst-case, ascent and average case sharpness are biased towards minimizing the maximal eigenvalue, minimal non-zero eigenvalue and trace of the Hessian, respectively. With an architecture-agnostic analysis, they show that 1-SAM minimizes the trace of Hessian like average-case sharpness, for small enough η and ρ. Similarly, the theoretical results by Andriushchenko and Flammarion (2022) rely on the assumption that learning rate η and perturbation radius ρ are chosen sufficiently close to 0. Arguably, the empirically optimal choice of η and ρ lies outside of this gradient flow-like regime and has qualitatively different properties (see e.g. edge of stability literature (Cohen et al., 2020; Arora et al., 2022)).

Scaling theory for SAM. Shin et al. (2023) suggest that the generalisation improvement by SAM continues to increase with growing overparametrization. This corroborates empirical observations that performance monotonically improves with scale, and understanding the infinite-width limit is not only of theoretical interest but entails immediate practical benefits.

Liu et al. (2022) introduce Look-LayerSAM with layerwise perturbation scaling for preserving good performance under large batch training for enhanced training parallelization. They use LAMB (You et al., 2020) for layerwise learning rate scaling for large batch training. The update scaling strategy in

these kinds of algorithms follows

$$W_{t+1}^l = W_t^l - \eta_t \phi(\|W_t^l\|_F) \frac{\nabla_{W^l}\mathcal{L}}{\|\nabla_{W^l} L\|_F},$$

with some $\phi : \mathbb{R}_+ \to \mathbb{R}_+$ and where $\nabla_{W^l}\mathcal{L}$ may be replaced by Adam's $\frac{m_t}{\sqrt{v_t}+\boldsymbol{\epsilon}}$. In practice, often simple functions like $\phi(x) = \max(c, \min(x, C))$ or $\phi(x) = x$ are used. The idea is to ensure that the update has the same order of magnitude as the weights. Look-LayerSAM follows an analogous approach for layerwise perturbation scaling. A derivation of μP for LAMB could also yield feature learning in all layers in the infinite-width limit as well as hyperparameter transfer. It certainly requires layerwise learning rate scaling. In the case $\phi(x) = x$, following a heuristic scaling derivation as in Appendix F.4 leads to layerwise learning rate scalings $\eta_1 = \eta_{L+1} = \Theta(1)$ and $\eta_l = \Theta(n^{-1/2})$ for hidden layers $l \in [2, L]$. With a bounded function like $\phi(x) = \max(c, \min(x, C))$, the scalings become $\eta_1 = \Theta(n^{1/2})$, $\eta_{L+1} = \Theta(n^{-1/2})$ and $\eta_l = \Theta(1)$ for hidden layers $l \in [2, L]$. We leave a closer investigation of feature learning and hyperparameter transfer with LAMB and Look-LayerSAM in SP and μP to future work.

C Definitions

In this section, we collect all definitions that do not appear in the main text. With minor modifications, we adopt all definitions from Yang and Hu (2021). If not stated otherwise, limits are taken with respect to width $n \to \infty$.

Definition C.1 (Big-O Notation). Given a sequence of scalar random variables $c = \{c_n \in \mathbb{R}\}_{n=1}^{\infty}$, we write $c = \Theta(n^{-a})$ if there exist constants A, B such that for almost every instantiation of $c = \{c_n \in \mathbb{R}\}_{n=1}^{\infty}$, for n large enough, $An^{-a} \leq |c_n| \leq Bn^{-a}$. Given a sequence of random vectors $x = \{x_n \in \mathbb{R}^n\}_{n=1}^{\infty}$, we say x has coordinates of size $\Theta(n^{-a})$ and write $x = \Theta(n^{-a})$ to mean the scalar random variable sequence $\left\{\sqrt{\|x_n\|^2/n}\right\}_n$ is $\Theta(n^{-a})$. For the definition of $c = O(n^{-a})$ and $c = \Omega(n^{-a})$, adapt the above definition of $c = \Theta(n^{-a})$ by replacing $An^{-a} \leq |c_n| \leq Bn^{-a}$ with $|c_n| \leq Bn^{-a}$ and $An^{-a} \leq |c_n|$, respectively. We write $x_n = o(n^{-a})$ if $n^a \cdot \sqrt{\|x_n\|^2/n} \to 0$ almost surely. ◀

Definition C.2 (Training routine). A *training routine* is a combination of base learning rate $\eta \geq 0$, perturbation radius $\rho \geq 0$, training sequence $\{(\xi_t, y_t)\}_{t\in\mathbb{N}}$ and a continuously differentiable loss function $\mathcal{L}(f(\xi), y)$ using the SAM update rule with layerwise perturbation scaling (LP). ◀

In addition to the stability conditions from the corresponding SGD result, we demand that the activation perturbations do not blow up. Otherwise the perturbations would strictly dominate both the initialization and the updates which makes the perturbation too strong and is avoided in practice.

Definition C.3 (Stability). We say a bcd-parametrization of an L-hidden layer MLP is *stable* if

1. For every nonzero input $\xi \in \mathbb{R}^{d_{\text{in}}} \setminus \{0\}$,
$$h_0^l, x_0^l = \Theta_\xi(1),\ \forall l \in [L], \quad \text{and} \quad \mathbb{E} f_0(\xi)^2 = O_\xi(1),$$
where the expectation is taken over the random initialization.
2. For any training routine, any time $t \in \mathbb{N}$, $l \in [L]$, $\xi \in \mathbb{R}^{d_{\text{in}}}$, we have
$$h_t^l(\xi) - h_0^l(\xi), x_t^l(\xi) - x_0^l(\xi) = O_*(1), \quad \text{and} \quad f_t(\xi) = O_*(1),$$
where the hidden constant in O_* can depend on the training routine, t, ξ, l and the initial function f_0.
3. For any training routine, any time $t \in \mathbb{N}_0$, $l \in [L]$, $\xi \in \mathbb{R}^{d_{\text{in}}}$, for the perturbed (pre-)activation $\tilde{h}_t^l := h^l(\tilde{W}_t), \tilde{x}_t^l := x^l(\tilde{W}_t)$ and output function $\tilde{f}_t(\tilde{W}_t)$ we have
$$\tilde{h}_t^l(\xi) - h_t^l(\xi), \tilde{x}_t^l(\xi) - x_t^l(\xi) = O_*(1), \quad \text{and} \quad \tilde{f}_t(\xi) = O_*(1),$$
where the hidden constant in O_* can depend on the training routine, t, ξ, l and the initial function f_0.

◀

Definition C.4 (**Nontriviality**). We say a bcd-parametrization is *trivial* if for every training routine, $f_t(\xi) - f_0(\xi) \to 0$ almost surely for $n \to \infty$, for every time $t > 0$ and input $\xi \in \mathbb{R}^{d_{\text{in}}}$. Otherwise the bcd-parametrization is *nontrivial*. ◀

Definition C.5 (**Feature learning**). We say a bcd-parametrization *admits feature learning in the l-th layer* if there exists a training routine, a time $t > 0$ and input ξ such that $x_t^l(\xi) - x_0^l(\xi) = \Omega_*(1)$, where the constant may depend on the training routine, the time t, the input ξ and the initial function f_0 but not on the width n. ◀

Definition C.6 (**Vanishing perturbations**). Let $l \in [L]$. We say that a stable bcd-parametrization *has vanishing perturbations in the l-th layer* if for any training routine, $t \in \mathbb{N}_0$ and $\xi \in \mathbb{R}^{d_{\text{in}}}$, it holds that $\tilde{x}_t^l - x_t^l = o(1)$, and it *has vanishing perturbations in the output* if for any training routine, $t \in \mathbb{N}_0$ and $\xi \in \mathbb{R}^{d_{\text{in}}}$ it holds that $\tilde{\delta} f_t(\xi) := f_{\tilde{W}_t}(\xi) - f_{W_t}(\xi) = o(1)$. ◀

Definition C.7 (**Perturbation nontriviality**). Let $l \in [L]$. We say that a stable bcd-parametrization is *perturbation nontrivial with respect to the l-th layer* if and only if it does not have vanishing perturbations in the l-th layer. A stable bcd-parametrization is *perturbation nontrivial with respect to the output* if it does not have vanishing perturbations in the output. ◀

Definition C.8 (**Effective perturbations**). Let $l \in [L+1]$. We say that a stable bcd-parametrization *effectively perturbs the l-th layer* if there exists a training routine, $t \in \mathbb{N}$ and $\xi \in \mathbb{R}^{d_{\text{in}}}$ such that $\tilde{\delta} W_t^l \tilde{x}_t^{l-1}(\xi) = \Theta(1)$ where $\tilde{\delta} W_t^l$ is defined in (LP) and $\tilde{x}_t^0 = x_t^0 = \xi_t$. ◀

Definition C.9 (σ-**gelu**). Define σ-gelu to be the function $x \mapsto \frac{x}{2}\left(1 + \operatorname{erf}\left(\sigma^{-1}x\right)\right) + \sigma\frac{e^{-\sigma^{-2}x^2}}{2\sqrt{\pi}}$. ◀

In order to apply the Tensor Program Master Theorem, all Nonlin and Moment operations in the NE⊗OR⊤ program, which do not only contain parameters as inputs, are required to be pseudo-Lipschitz in all of their arguments. For training with SGD, this is fulfilled as soon as ϕ' is pseudo-Lipschitz. Both `tanh` as well as σ-`gelu` fulfill this assumption.

Definition C.10 (**Pseudo-Lipschitz**). A function $f : \mathbb{R}^k \to \mathbb{R}$ is called *pseudo-Lipschitz of degree d* if there exists a $C > 0$ such that $|f(x) - f(y)| \leq C\|x - y\|(1 + \sum_{i=1}^k |x_i|^d + |y_i|^d)$. We say f is *pseudo-Lipschitz* if it is so for any degree d. ◀

D Extensive main results

Using the formal definitions from Appendix C, here we provide the full formal statements of all of our main theoretical results together with further details and implications. The proof of all statements is provided in Appendix E. Since SAM evaluates the gradients on perturbed weights, it is not covered by the update rule definition in Yang and Littwin (2023) and an infinite-width analysis requires explicitly deriving the corresponding NE⊗OR⊤ program, scalings and infinite-width limits.

Recall that our definition of bcd-parameterizations extends abc-parameterizations by setting the maximal perturbation scaling to n^{-d} and allowing relative downweighting n^{-d_l} of the global scaling in each layer l. The perturbation scaling does not affect the choice of layerwise initialization variance scalings b_l and the layerwise learning rate scalings c_l. Common bc-parametrizations for SGD are summarized in Table D.1. SAM with SGD as a base optimizer requires the same scalings. Similarly, SAM with Adam as a base optimizer requires the same scalings as Adam (Yang et al., 2022, Table 3). Recall that, for convenience, we require width-independent denominator scaling $\|v_t\| = \Theta(1)$ of the scaled gradient for the perturbation (LP), which imposes the constraints

$$d_1 \geq 1/2 - \min(b_{L+1}, c_{L+1}), \quad d_l \geq 1 - \min(b_{L+1}, c_{L+1}) \text{ for } l \in [2, L], \quad d_{L+1} \geq 1/2. \tag{D.1}$$

All (pre-)activation and function outputs can be thought of as outputs given a fixed input $\xi \in \mathbb{R}^{d_{\text{in}}} \backslash \{0\}$ with $d_{in} \in \mathbb{N}$ fixed, e.g. $f_t := f_{W_t} := f_{W_t}(\xi)$. For the perturbed weights we write $\tilde{W}_t := W_t + \tilde{\delta} W_t$, with $\tilde{\delta} W_t$ defined in (LP) as $\boldsymbol{\varepsilon}_t^l$. Here we write weight perturbations as $\tilde{\delta} W_t^l$ instead of $\boldsymbol{\varepsilon}_t^l$ to show the resemblance to weight updates δW_t^l. Perturbed activations and function outputs at time t are written as $\tilde{x}_t^l(\xi) = x_{\tilde{W}_t}^l(\xi)$ and $\tilde{f}_t(\xi) = f_{\tilde{W}_t^l}(\xi)$. Recall that for all of the results in this section we make the following smoothness assumption on the activation function.

Assumption 1 (**Smooth activation function**). The used activation function is either `tanh` or σ-`gelu` for $\sigma > 0$ sufficiently small. ◀

We define the maximal feature update scale of a *bcd*-parameterization

$$r := \min(b_{L+1}, c_{L+1}, d + d_{L+1}) + \min_{l=1}^{L}(c_l - \mathbb{I}(l \neq 1)). \tag{D.2}$$

as well as the maximal feature perturbation scale of a *bcd*-parameterization

$$\tilde{r} := \min(b_{L+1}, c_{L+1}) + d + \min_{l=1}^{L}(d_l - \mathbb{I}(l \neq 1)). \tag{D.3}$$

Stability requires the constraints (a-c) from SGD and additional perturbation stability constraints (d-e) that include the layerwise perturbation scales $\{d_l\}_{l=1,\dots,L+1}$.

Theorem D.2 (Stability characterization). *A bcd-parametrization is stable if and only if all of the following are true:*

- *(a) (Stability at initialization, $h_0^l, x_0^l = \Theta(1)$ for all l, $f_0 = O(1)$)*
  $b_1 = 0$, $b_l = 1/2$ for $l \in [2, L]$ and $b_{L+1} \geq 1/2$.
- *(b) (Features do not blow up during training, i.e. $\Delta x_t^l = O(1)$ for all l)*
  $r \geq 0$.
- *(c) (Output function does not blow up during training, i.e. $\Delta W_t^{L+1} x_t^L, W_0^{L+1} \Delta x_t^L = O(1)$)*
  $c_{L+1} \geq 1$ and $b_{L+1} + r \geq 1$.
- *(d) (Feature perturbations do not blow up, i.e. $\tilde{\delta} x_t^l = O(1)$ for all l)*
  $\tilde{r} \geq 0$.
- *(e) (Output function perturbations do not blow up during training, i.e. $\tilde{\delta} W_t^{L+1} \tilde{x}_t^L, W_t^{L+1} \tilde{\delta} x_t^L = O(1)$)*
  $d + d_{L+1} \geq 1$ and $b_{L+1} + \tilde{r} \geq 1$.

The nontriviality and feature learning characterizations from SGD remain unaltered. This is because in the definition of r, it holds that $d + d_{L+1} \geq 1$ (from perturbation stability), and $\min(b_{L+1}, c_{L+1}) \leq 1$ already had to hold for nontriviality in SGD, so that stable perturbation scaling does not affect r.

Theorem D.3 (Nontriviality characterization). *A stable bcd-parametrization is nontrivial if and only if $c_{L+1} = 1$ or $\min(b_{L+1}, c_{L+1}) + r = 1$.*

As for nontriviality, the conditions under which a stable, nontrivial parameterization is feature learning in the infinite-width limit are decoupled from the choice of perturbation scalings $\{d_l\}_{l \in [L+1]} \cup \{d\}$. Hence the conditions are the same as for SGD. Below we provide a slightly refined result in terms of the maximal feature update scale r_{l_0} of a *bcd*-parameterization up to layer l_0 (as provided in the Appendix of Yang and Hu (2021)).

Theorem D.4 (Feature learning characterization). *For any $l_0 \in [L]$, the following statements are equivalent:*

- *(a) A stable, nontrivial bcd-parametrization admits feature learning in layer l_0.*
- *(b) A stable, nontrivial bcd-parametrization admits feature learning in layer l for all $l \geq l_0$.*
- *(c) $r_{l_0} := \min(b_{L+1}, c_{L+1}, d + d_{L+1}) + \min_{m=1}^{l_0}(c_m - \mathbb{I}(m \neq 1)) = 0$.*

Consequently, a stable, nontrivial bcd-parametrization admits feature learning (at least in the last layer activations) if and only if $r = 0$.

Remark D.5 (Effective feature learning). As for perturbations, feature learning in later layers can be caused by weight updates in earlier layers that propagate through the network. One could demand effective feature learning in the l-th layer as $\delta W_t^l x_t^{l-1} = \Theta(1)$ and it would occur if and only if $\min(b_{L+1}, c_{L+1}, d + d_{L+1}) + c_l - \mathbb{I}(l \neq 1) = 0$. ◀

As for nontriviality, perturbation nontriviality in the output is attained if the constraints for $\tilde{\delta} W_t^{L+1} \tilde{x}_t^L$ or $W_t^l \tilde{\delta} x_t^L$ are exactly satisfied.

Theorem D.6 (Perturbation nontriviality characterization). *Let $l \in [L]$. A stable bcd-parametrization is perturbation nontrivial with respect to the l-th layer if and only if*

$$\tilde{r}_l := \min(b_{L+1}, c_{L+1}) + d + \min_{m=1}^{l}(d_m - \mathbb{I}(m \neq 1)) = 0.$$

A stable bcd-parametrization is perturbation nontrivial with respect to the output if and only if $d + d_{L+1} = 1$ or $\min(b_{L+1}, c_{L+1}) + \tilde{r} = 1$.

The converse formulation of the perturbation-nontriviality results characterizes the regime of vanishing perturbations.

Corollary D.7 (Vanishing perturbation characterization). *For any $l_0 \in [L]$, the following statements are equivalent:*

(a) A stable bcd-parametrization has vanishing perturbations in layer l_0.
(b) A stable bcd-parametrization has vanishing perturbations in layer l for all $1 \leq l \leq l_0$.
(c) $\tilde{r}_{l_0} := \min(b_{L+1}, c_{L+1}) + d + \min_{m=1}^{l_0}(d_m - \mathbb{I}(m \neq 1)) > 0$.

A stable bcd-parametrization has vanishing perturbations with respect to all layers and the output function if and only if $d_{L+1} > 1/2$ and $\tilde{r} > \max(0, 1 - b_{L+1})$. This case reduces to the results in Yang and Hu (2021).

For perturbation nontriviality it suffices that the perturbation in any of the previous layers is scaled correctly. For effective perturbations, we need the correct scaling in exactly that layer.

Theorem D.8 (Effective perturbation characterization). *For $l \in [L]$, a stable bcd-parametrization effectively performs SAM in the l-th layer if and only if $\min(b_{L+1}, c_{L+1}) + d + d_l - \mathbb{I}(l \neq 1) = 0$.*

A stable bcd-parametrization effectively performs SAM in the last layer if and only if $d + d_{L+1} = 1$.

The above understanding of all update and perturbation scalings allows us to extract the most important consequences of different choices of perturbation scaling on the learning dynamics. Beyond vanishing hidden layer perturbations, the following theorem shows that the joint gradient norm $\|v_t\|$ can be approximated efficiently without an additional backward pass under global perturbation scaling.

Theorem D.9 (Global Perturbation Scaling). *Given any stable bcd-parametrization $\{b_l\}_{l\in[L+1]} \cup \{c_l\}_{l\in[L+1]} \cup \{d_l\}_{l\in[L+1]} \cup \{d\}$. The parametrization performs updates in the original gradient direction if and only if $d_l = C$ for all $l \in [L+1]$ for some $C \in \mathbb{R}$. In this case, the parametrization has vanishing perturbations in all hidden layers $l \in [L]$, and the last layer $l = L+1$ is effectively perturbed if and only if $d = 1/2$. If $b_{L+1} > 1/2$ (as in μP), the gradient norm is dominated by the last layer and simplifies to,*

$$\|v_t\| = \Theta(n^{1/2-C}), \qquad \|v_t\| - \mathcal{L}'(f_t(\xi_t), y_t)\|x_t^L\| = o(n^{1/2-C}).$$

One might suspect that it is desirable to let all layers contribute non-vanishingly to the gradient norm in the denominator of (LP). The following proposition shows that this should be avoided with our definition of *bcd*-parameterizations. Of course, if we add even more hyperparameters by decoupling numerator and denominator scalings, we can set all contributions to $\Theta(1)$, which is what we do in Appendix F.7.

Proposition D.10 (Balancing gradient norm contributions). *Given any stable bcd-parametrization $\{b_l\}_{l\in[L+1]} \cup \{c_l\}_{l\in[L+1]} \cup \{d_l\}_{l\in[L+1]} \cup \{d\}$. If all layers contribute to the gradient norm non-vanishingly in the limit, i.e. $\|v_t^l\| = \Theta(\|v_t\|)$ for all $l \in [L+1], t \in \mathbb{N}_0$, then the parametrization has vanishing perturbations in all hidden layers $l \in [L]$. Such a parametrization effectively performs SAM in the last layer $l = L+1$ if and only if $d = 1/2$.*

The following theorem provides the unique correct perturbation scaling for any stable *bc*-parameterization with $b_{L+1} \geq 1$.

Theorem D.11 (Perturbation Scaling Choice for Effective Perturbations). *Given any stable bcd-parametrization $\{b_l\}_{l\in[L+1]} \cup \{c_l\}_{l\in[L+1]} \cup \{d_l\}_{l\in[L+1]} \cup \{d\}$. If $b_{L+1} < 1$, then there does not exist a stable choice of $\{d_l\}_{l\in[L+1]} \cup \{d\}$ that achieves effective perturbations before the last layer. If $b_{L+1} \geq 1$, then up to the equivalence $d_l' = d_l + C$, $C \in \mathbb{R}$, $\forall l \in [L+1]$, the unique stable choice $\{d_l\}_{l\in[L+1]} \cup \{d\}$ with effective perturbations in all layers $l \in [L+1]$ is given by*

$$d = -1/2, \qquad d_l = \begin{cases} 1/2 - \min(b_{L+1}, c_{L+1}) & l = 1, \\ 3/2 - \min(b_{L+1}, c_{L+1}) & l \in [2, L], \\ 3/2 & l = L+1. \end{cases} \tag{D.4}$$

In this parameterization, the first layer dominates the gradient norm as

$$\|v_t\| = \Theta(1), \qquad \big|\|v_t^1\| - \|v_t\|\big| = \Theta(n^{-1/2}).$$

	Definition	SP	SP (stable)	NTP (stable)	μP
b_l	$\mathcal{N}(0, n^{-2b_l})$	$\begin{cases}0 & l=1,\\ 1/2 & l\geq 2.\end{cases}$	$\begin{cases}0 & l=1,\\ 1/2 & l\geq 2.\end{cases}$	$\begin{cases}0 & l=1,\\ 1/2 & l\geq 2.\end{cases}$	$\begin{cases}0 & l=1,\\ 1/2 & l\in[2,L],\\ 1, & l=L+1.\end{cases}$
c_l	LR ηn^{-c_l}	0	1	$\begin{cases}0 & l=1,\\ 1 & l\geq 2.\end{cases}$	$\begin{cases}-1 & l=1,\\ 0 & l\in[2,L],\\ 1 & l=L+1.\end{cases}$
r	Equation (D.2)	-1	1/2	1/2	0
Stable?			✓	✓	✓
Nontrivial?			✓	✓	✓
Feature learning?					✓

Table D.1: **(bc-parametrizations)** Overview over common implicitly used bc-parametrizations for training MLPs without biases in standard parametrization (SP), standard parametrization with maximal stable nonadaptive LR $c = 1$ (SP (stable)), neural tangent parametrization (NTP) and maximal update parametrization (μP).

	Definition	Naive	Global (stable)	Effective
d	ρn^{-d}	0	$1/2$	$-1/2$
d_l	$n^{-d_l}\nabla_{W^l}\mathcal{L}_t$	$1/2$	$1/2$	$\begin{cases}1/2-c_\nabla & l=1,\\ 3/2-c_\nabla & l\in[2,L],\\ 3/2 & l=L+1.\end{cases}$
$\tilde{r}$	Equation (D.3)	$c_\nabla - 1/2$	c_∇	0
Stable?		✗	✓	✓
Last layer effectively perturbed?		✗	✓	✓
All layers effectively perturbed?		✗	✗	✓

Table D.2: **(Perturbation scalings)** Overview over important choices of the global perturbation scaling ρn^{-d} and the layerwise perturbation scalings n^{-d_l} for training MLPs without biases with SAM: Naive scaling without width dependence (Naive), maximal stable global scaling along the original gradient direction (Global) and the unique scaling that achieves effective perturbations in all layers (Effective). An extensive overview that characterizes all possible choices of perturbation scaling is provided in Appendix F.1. Recall the gradient scaling $c_\nabla := \min(b_{L+1}, c_{L+1})$.

Table D.2 summarizes the consequences of Theorem D.11. Together with Theorem D.11, the following proposition suggests that $b_{L+1} = 1$ is a good choice. However $b_{L+1} > 1$ can also induce effective perturbations, as long as d and d_{L+1} are chosen correctly.

Proposition D.12 (Effects of last-layer initialization b_{L+1} on all perturbations). *If a stable bcd-parametrization with $\min(b_{L+1}, c_{L+1}) \leq 1$ is perturbation nontrivial with respect to any hidden layer $l \in [L]$, it is also perturbation nontrivial with respect to the output.*

Lastly, the following proposition shows that effective perturbations from the first layer propagate through the entire network.

Proposition D.13 (Perturbations propagate through the forward pass). *All stable bcd-parametrizations with $d_1 = -\min(b_{L+1}, c_{L+1}) - d$ effectively perturb the first layer and are perturbation nontrivial in all layers.*

Remark D.14 (Efficiency gains). The above results may be used for efficiency gains. Given any stable bcd-parametrization, we can compute the maximal layer l_0 such that $\tilde{r}_{l_0} > 0$, and in wide networks do not have to compute SAM perturbations before layer $l_0 + 1$; as soon as $b_{L+1} > 1/2$ (as for μP), the gradient norm for the SAM update rule is approximately given by $\|\nabla L_t\| \approx \mathcal{L}'(f_t(\xi_t), y_t)\|x_t^L\|$, which can directly be computed without an additional backward pass. The practical recommendation from our experiments however is to either use μP^2 or to completely abstain from perturbations. ◀

Remark D.15 (SAM without gradient normalization). For the SAM update rule without gradient normalization simply set $d = 0$ and remove the gradient norm constraints (D.1) to arrive at the

adapted NE⊗OR⊤ program and bcd-constraints. Note that standard parametrization gets even more unstable without dividing by $\|\nabla L\| = \Theta(n^{1/2})$, now requiring $d_{L+1} \geq 1$ for stability. Similar to the previous results, this shows that unawareness of bcd-parametrizations requires strongly scaling down ρ for stability, while vasting computation on vanishing perturbations before the last layer. More details can be found in Appendix F.3. ◀

E Proof of main results

In this section we derive the NE⊗OR⊤ program that corresponds to training a MLP without biases with SAM. For simplicity and clarity of the proof, we prove the one-dimensional case $d_{in} = 1$, $d_{out} = 1$, but an extension to arbitrary but fixed d_{in}, d_{out} is straightforward. Recall Assumption 1 that allows us to apply the Tensor Program Master Theorem and explicitly state the infinite-width limit of training MLPs with SAM in Appendix E.2.

E.1 Tensor program formulation

E.1.1 Tensor Program initialization

We initialize the matrices $W_0^2, \dots, W_0^L$ as $(W_0^l)_{\alpha\beta} \sim \mathcal{N}(0, 1/n)$, which absorbs $b_l = 1/2$.

We initialize the input layer matrix $W_0^1 \in \mathbb{R}^{n\times 1}$ and normalized output layer matrix $\hat{W}_0^{L+1} = W_0^{L+1} n^{b_{L+1}} \in \mathbb{R}^{1\times n}$ as $(W_0^1)_\alpha, (\hat{W}_0^{L+1})_\alpha \sim \mathcal{N}(0, 1)$, as initial vectors should have a distribution that is $\Theta(1)$.

In the NE⊗OR⊤formulation, we write all quantities as $\theta_z z$, where θ_z denotes their scaling n^C for some $C \in \mathbb{R}$ and z therefore has a $\Theta(1)$ distribution. The stability, nontriviality and feature learning conditions then stem from requiring either $\theta_z \to 0$ or $\theta_z = 1$ depending on z and its desired scale.

E.1.2 First forward pass

We denote a definition of a Tensor Program (TP) or NE⊗OR⊤computation as :=. Compared to MLPs trained with SGD nothing changes in the first forward pass,

$$h_0^1(\xi) := W_0^1 \xi \quad \text{(NL)}, \quad x_0^l := \phi(h_0^l) \quad \text{(NL)}, \quad h_0^{l+1} := W_0^{l+1} x_0^l. \quad \text{(MatMul)}$$

In the case of MuP, $f_0(\xi) = W_0^{L+1} x_0^L(\xi) \to 0$ defines a scalar in the TP.

Observe the scalings $x_0^1 = \Theta(h_0^1) = \Theta(n^{-b_1}), x_0^l = \Theta(h_0^l) = \Theta(n^{1/2-b_l})$ for $l \in [2, L]$ due to CLT, independence at initialization and $x_0^l = \Theta(h_0^l) = \Theta(1)$ by stability. Hence stability at initialization inductively requires $b_1 = 0$, $b_l = 1/2$ for $l \in [2, L]$ and $b_{L+1} \geq 1/2$.

E.1.3 First backward pass

The chain rule of the derivative remains the same, we just evaluate on different weights compared to standard SGD. We denote the adversarially perturbed weights by $\tilde{W}_t^l$ and the normalized perturbations by $\tilde{\delta} W_t^l$. Before computing the updates we have to compute a full backward pass to determine these perturbed weights for each layer, and then compute a forward pass with these perturbed weights to compute the perturbed preactivations $\tilde{h}_t^l$ that we will need for computing the SAM update. Therefore the NE⊗OR⊤ program for SAM maintains a perturbed copy of all preactivations, activations, last-layer weights and logits just for computing the updates of the actual parameters.

Under MuP, the loss derivative with respect to the function remains $\chi_0 := \mathcal{L}'(f_0(\xi_0), y_0) \to \mathring{\chi}_0 := \mathcal{L}'(0, y_0)$. For the weight perturbation, we need to perform a SGD backward pass,

$$dx_0^L := \hat{W}_0^{L+1}, \quad dh_0^l := dx_0^l \odot \phi'(h_0^l), \quad dx_0^{l-1} := (W_0^l)^T dh_0^l,$$

where $dz := \theta_\nabla^{-1} \nabla_z f$. For SGD (and for SAM, as we will see later) all gradients have scaling $\theta_\nabla := n^{-b_{L+1}}$ in the first step, whereas we overload the notation $\theta_\nabla := n^{-\min(b_{L+1}, c_{L+1})}$ for all later steps. For clarity of presentation assume $b_{L+1} \geq c_{L+1}$ here, the other case follows analogously. For the first step this can be understood from

$$\nabla_{x^L} f_0 = W_0^{L+1} = \Theta(n^{-b_{L+1}}), \qquad \nabla_{h^L} f_0 = \nabla_{x^L} f_0 \odot \phi'(h_0^L) = \Theta(n^{-b_{L+1}}),$$

since $h_0^L = \Theta(1)$ by the stability assumption, and this scale $\Theta(n^{-b_{L+1}})$ propagates through all layers via the chain rule and remains stable in later backward passes. For hidden layer gradients, observe that

$$\begin{aligned}\nabla_{x^{L-1}} f_t &= (W_t^L)^T \nabla_{h^L} f_t = (W_0^L + \Delta W_t^L)^T \nabla_{h^L} f_t \\ &= \Theta\left((W_0^L)^T \nabla_{h^L} f_t - n^{-c_L} \sum_{s=0}^{t-1} ((\nabla_{h^L} f_s)^T \nabla_{h^L} f_t) x_s^{L-1}\right) \\ &= \Theta(n^{1-2b_L}\theta_\nabla - n^{-c_L}\theta_\nabla^2 n) = \Theta(\theta_\nabla),\end{aligned}$$

where first term's scale stems from the products $(W_0^L)^T W_0^L v = \Theta(n^{1-2b_L} v)$ due to Yang (2021), $b_L = 1/2$ for stability at initialization and $b_{L+1} + c_L \geq 1$ for update stability during training ($r \geq 0$). If we allowed the second term to strictly dominate, the gradient scale would explode iteratively in the backward pass.

The gradient norm. Before computing the weight perturbations, we need to compute the gradient norm for the SAM update. The gradient norm at time t in each layer $l \in [2, L]$ is given by the scalar,

$$\theta_\nabla^{-2} \left\| \frac{\partial L_t}{\partial W^l} \right\|^2 = \sum_{i,j=1}^{n} \left(\chi_t (dh_t^l)_i (x_t^{l-1})_j\right)^2 = \chi_t^2 \|dh_t^l (x_t^{l-1})^T\|_F^2 = \chi_t^2 \big((dh_t^l)^T dh_t^l\big)\big((x_t^{l-1})^T x_t^{l-1}\big),$$

where $\chi_t = \mathcal{L}'(f_t(\xi_t), y_t)$ and we used $\partial h^l / \partial W_{ij}^l = (x_j^{l-1}\delta_{ik})_{k=1,\dots,n}$.

Hence the gradient norm of all weights jointly is given by the unnormalized scalar

$$\|\nabla_w L_t\|^2 = \chi_t^2 \left(n\theta_\nabla^2 \frac{(dh_t^1)^T dh_t^1}{n} (\xi_t^T \xi_t) + \sum_{l=2}^{L} n^2 \theta_\nabla^2 \frac{(dh_t^l)^T dh_t^l}{n} \frac{(x_t^{l-1})^T x_t^{l-1}}{n} + n \frac{(x_t^L)^T x_t^L}{n} \right), \tag{E.1}$$

with scaling $\theta_{\|\nabla\|}^2 = \Theta(n^2\theta_\nabla^2 + n) = \Theta(n)$, because stability at initialization requires $b_{L+1} \geq 1/2$ so that $n^2\theta_\nabla^2 \leq n$. Note that the first layer contributes vanishingly to the gradient norm, the hidden layer gradients only if $b_{L+1} = 1/2$ (equivalently $f_0 = \Theta(1)$) and the last-layer activations always in dominating order. So in μP, in the limit, $\|\nabla_w L_t\| = \mathcal{L}'(f_t(\xi_t), y_t)\|x_t^L\|$. This means that the unscaled gradient always aligns with the last-layer activation. For learning in μP, this dominance is corrected by the layerwise learning rates.

The squared norm of the rescaled gradient is given by

$$\begin{aligned}\|v_t\|^2 = \chi_t^2 \Bigg(& n\theta_\nabla^2 n^{-2d_1} \frac{(dh_t^1)^T dh_t^1}{n} (\xi_t^T \xi_t) \\ & + \sum_{l=2}^{L} n^2 \theta_\nabla^2 n^{-2d_l} \frac{(dh_t^l)^T dh_t^l}{n} \frac{(x_t^{l-1})^T x_t^{l-1}}{n} + n n^{-2d_{L+1}} \frac{(x_t^L)^T x_t^L}{n} \Bigg),\end{aligned} \tag{E.2}$$

with scaling $\theta_v^2 = \Theta(n^{1-2d_1}\theta_\nabla^2 + \sum_{l=2}^{L} n^{2-2d_l}\theta_\nabla^2 + n^{1-2d_{L+1}})$. For simplicity, set $\theta_v = 1$. This raises the constraints $n^{1-2d_1}\theta_\nabla^2 \leq 1$, $n^{2-2d_l}\theta_\nabla^2 \leq 1$ for $l \in [2, L]$ and $n^{1-2d_{L+1}} \leq 1$, which can be rewritten as

$$d_1 \geq 1/2 - \min(b_{L+1}, c_{L+1}), \quad d_l \geq 1 - \min(b_{L+1}, c_{L+1}) \text{ for } l \in [2, L], \quad d_{L+1} \geq 1/2,$$

where at least one equality is demanded to hold in order to attain $\theta_v = 1$. If one of the equalities holds, the respective layer contributes to the norm non-vanishingly in the limit.

Thus, applying the square root and dividing by $\theta_v = 1$ the square root of (E.2) defines a normalized TP scalar.

Perturbations. Stability implies that also the perturbed (pre-)activations and output function remain $\Theta(1)$ and $O(1)$ respectively. Otherwise a SAM training step would induce blowup in the updates. We call this weaker property of just the perturbations *perturbation stability*.

Definition E.1 (Perturbation stability)**.** We call a *bcd*-parametrization *perturbation stable* if and only if $\tilde{h}_t^l, \tilde{x}_t^l = \Theta(1)$ for all $l \in [L]$ and $t \in \mathbb{N}$ and $\tilde{\delta} f_t = O(1)$ for all $t \in \mathbb{N}$. ◀

Mathematically we get the normalized weight perturbations for $l \in \{2, \dots, L\}$,

$$\tilde{\delta}W_0^{L+1} := \frac{\rho\ \chi_0\ x_0^L}{\|v_0\|}, \quad \tilde{\delta}W_0^l = \frac{\rho\ \chi_0\ dh_0^l\ (x_0^{l-1})^T}{\|v_0\|}, \quad \tilde{\delta}W_0^1 = \frac{\rho\ \chi_0\ dh_0^1\ \xi_0^T}{\|v_0\|},$$

which scale as $\tilde{\theta}_{L+1} := \tilde{\theta}_{W^{L+1}} := n^{-(d+d_{L+1})}$, $\Theta(n^{(d+d_l)-b_{L+1}})$ and $\Theta(n^{-(d+d_1)-b_{L+1}})$ respectively. But the NE⊗OR⊤ program computation rules do not allow to compute matrices $\tilde{\delta}W_0^l, l \in [L]$, therefore we use the weight updates to directly compute the preactivation and activation changes analogous to the t-th forward pass. For all $t \geq 0$, we write

$$\tilde{h}_t^l = h_t^l + \tilde{\theta}_l \tilde{\delta} h_t^l, \qquad \tilde{x}_t^l = x_t^l + \tilde{\theta}_l \tilde{\delta} x_t^l,$$

with the perturbations for $l \in [2, L]$,

$$\begin{aligned}
\tilde{\delta}h_0^1(\xi) :=\ & +\frac{\rho\chi_0(\xi_0^T\xi)dh_0^1}{\|v_0\|}, \\
\tilde{\delta}x_t^l :=\ & \tilde{\theta}_l^{-1}(\phi(h_t^l + \tilde{\theta}_l\tilde{\delta}h_t^l) - \phi(h_t^l)), \\
\tilde{\theta}_l\tilde{\delta}h_0^l :=\ & \tilde{\theta}_{l-1}W_0^l\tilde{\delta}x_0^{l-1} + (\tilde{W}_0^l - W_0^l)\tilde{x}_0^{l-1} \\
=\ & \tilde{\theta}_{l-1}W_0^l\tilde{\delta}x_0^{l-1} + \rho\tilde{\theta}_{W^l}\frac{\chi_0}{\|v_0\|}\frac{(x_0^{l-1})^T\tilde{x}_0^{l-1}}{n}dh_0^l,
\end{aligned}$$

which defines a NonLin operation with the vectors $W_0^l\tilde{\delta}x_0^{l-1}$ and dh_0^l and everything else treated as scalars, and with first backward pass scalings $\tilde{\theta}_{W^1} := n^{-(d+d_1)}\theta_\nabla$, $\tilde{\theta}_{W^l} := n^{1-(d+d_l)}\theta_\nabla$ and $\tilde{\theta}_l := \max(\tilde{\theta}_{l-1}, \tilde{\theta}_{W^l}) = \max_{m=1}^l \tilde{\theta}_{W^m}$, where we used that $\tilde{x}_0^{l-1} = \Theta(1)$ due to perturbation stability. Note that these scalings may implicitly increase when $t > 0$ since $\theta_\nabla = n^{-b_{L+1}}$ gets replaced by $\theta_\nabla = n^{-\min(b_{L+1}, c_{L+1})}$.

The activation perturbations can then simply be defined via the NonLin operation,

$$\tilde{\delta}x_0^l := \tilde{\theta}_l^{-1}(\phi(h_0^l + \tilde{\theta}_l\tilde{\delta}h_0^l) - \phi(h_0^l)),$$

with the same scaling as $\tilde{\delta}h_0^l$.

The perturbation of the scalar output function can simply be defined via the NonLin operation,

$$\tilde{\delta}f_0 := \tilde{W}_0^{L+1}\tilde{x}_0^L - W_0^{L+1}x_0^L = \tilde{\theta}'_{L+1}\frac{\tilde{\delta}W_0^{L+1}\tilde{x}_0^L}{n} + \tilde{\theta}'_{L\nabla}\frac{\hat{W}_0^{L+1}\tilde{\delta}x_0^L}{n},$$

with $\tilde{\theta}'_{L+1} := n\tilde{\theta}_{W^{L+1}}$ and $\tilde{\theta}'_{L\nabla} := n\theta_\nabla\tilde{\theta}_L$.

SAM Update. Finally, we can compute the SAM updates as follows. In the case $\min(b_{L+1}, c_{L+1}) \leq d + d_{L+1}$ the weight perturbation scale is dominated by the weight scale, so that

$$dx_{SAM,0}^L := \hat{W}_0^{L+1} + \tilde{\theta}_{(L+1)/\nabla}\ \tilde{\delta}W_0^{L+1},$$

with $\tilde{\theta}_{(L+1)/\nabla} := \tilde{\theta}_{L+1}/\theta_\nabla \leq 1$, whereas if $\min(b_{L+1}, c_{L+1}) > d + d_{L+1}$ we write

$$dx_{SAM,0}^L := \tilde{\theta}_{\nabla/(L+1)}\hat{W}_0^{L+1} + \tilde{\delta}W_0^{L+1},$$

with $\theta_{\nabla/(L+1)} := \theta_\nabla/\tilde{\theta}_{L+1} \leq 1$. In any case, the scaling of $dx_{SAM,0}^L$ and all other SAM gradients is $\theta_{SAM} := \max(\theta_\nabla, n^{-(d+d_{L+1})}) = n^{-\min(b_{L+1}, c_{L+1}, d+d_{L+1})}$. The other SAM gradients are given by

$$\begin{aligned}
dh_{SAM,0}^l :=\ & dx_{SAM,0}^l \odot \phi'(\tilde{h}_0^l) \\
dx_{SAM,0}^{l-1} :=\ & (\tilde{W}_0^l)^T dh_{SAM,0}^l = (W_0^l + \tilde{\theta}_{W^l}\tilde{\delta}W_0^l)^T dh_{SAM,0}^l \\
=\ & (W_0^l)^T dh_{SAM,0}^l + \rho\theta_{SAM}\tilde{\theta}_{W^l}\frac{\chi_0}{\|v_0\|}\frac{(dh_0^l)^T dh_{SAM,0}^l}{n}x_0^{l-1}.
\end{aligned}$$

where the last line define a NonLin operation in the vectors $(W_0^l)^T dh_{SAM,t}^l$ and x_0^{l-1} and everything else treated as scalars. Consequently, $\nabla_{h_0^l} f|_{\tilde{W}_0}$ is of the same scale as $\nabla_{x_0^l} f|_{\tilde{W}_0}$ and $\nabla_{x_0^{l-1}} f|_{\tilde{W}_0}$ is of the scale $\max(\theta_{SAM}, \tilde{\theta}_{W^l}\theta_{SAM}) = \theta_{SAM}$ since $\tilde{\theta}_{W^l} \leq 1$ is required for perturbation stability.

Note that for SAM's weight updates the loss derivative is also evaluated on the perturbed weights,

$$\tilde{\chi}_0 := \mathcal{L}'(\tilde{W}_0^{L+1}\tilde{x}_0^L, y_0).$$

Constraints on the output function. Assuming $\tilde{x}_0^L = \Theta(1)$ (perturbation stability), we get $\tilde{\chi}_0 = O(1)$ if and only if $\tilde{\delta}W_0^{L+1} = O(n^{-1})$ if and only if $d + d_{L+1} \geq 1$.

We have $\tilde{\chi}_0 = \Theta(1)$ if and only if $\tilde{f}_0 = \tilde{W}_0^{L+1}\tilde{x}_0^L = \Theta(1)$. This can either be caused by changes in the last-layer weights, by non-vanishing initial function $W_0^{L+1}x_0^L$ (if and only if $b_{L+1} = 1/2$) or by $W_0^{L+1}\tilde{\delta}x_0^L = \Theta(1)$, which holds if and only if $b_{L+1} + \tilde{r}_L = 1$ (analogously, $W_0^{L+1}\tilde{\delta}x_0^L = O(1)$ if and only if $b_{L+1} + \tilde{r}_L \geq 1$). The first case requires $\tilde{\delta}W_0^{L+1} = \Theta(n^{-1})$, since $\tilde{\delta}W_0^{L+1}$ and $\tilde{x}_0^L$ are highly correlated. $\tilde{\delta}W_0^{L+1} = \Theta(n^{-1})$ is fulfilled if and only if $d + d_{L+1} = 1$ (the analogue to $c_{L+1} \geq 1$ for stability and $c_{L+1} = 1$ for nontriviality).

Hence perturbation stability of the output function holds only if $d + d_{L+1} \geq 1$ and $b_{L+1} + \tilde{r}_L \geq 1$. Then, perturbation nontriviality holds if and only if $d + d_{L+1} = 1$ or $b_{L+1} + \tilde{r}_L = 1$.

In the t-th backward pass, $b_{L+1} + \tilde{r}_L \geq 1$ will be replaced by the slightly stronger constraint $b_{L+1} + \tilde{r} \geq 1$.

E.1.4 t-th forward pass

Formally, we sum the updates in each step,

$$\hat{W}_t^{L+1} := \hat{W}_0^{L+1} + \theta_{L+1/\nabla}(\delta W_1^{L+1} + \cdots + \delta W_t^{L+1}),$$

where $\delta W_{t+1}^{L+1} := -\eta\ \tilde{\chi}_t\ (\tilde{x}_t^L)^T$ denotes the normalized change in the weights W^{L+1} (as a row vector) of scaling $\theta_{L+1} = \theta_{W^{L+1}} = n^{-c_{L+1}}$ under perturbation stability and nontriviality so that $\hat{W}_t^{L+1}$ scales as $\theta_\nabla = n^{-\min(b_{L+1}, c_{L+1})}$. δW_{t+1}^{L+1} should not be confused with $\tilde{\delta}W_{t+1}^{L+1}$ which denotes the perturbation of the weights at time $t+1$. For every nontrivial stable parametrization we have $\tilde{\chi}_t = \Theta(1)$ and $\tilde{x}_t^L = \Theta(1)$ which requires $\tilde{\theta}_L \leq 1$. In the case $c_{L+1} < b_{L+1}$, we write $\hat{W}_t^{L+1} := n^{-b_{L+1}+c_{L+1}}\hat{W}_0^{L+1} + (\delta W_1^{L+1} + \cdots + \delta W_t^{L+1})$ with the same scaling $\theta_\nabla = n^{-\min(b_{L+1}, c_{L+1})}$.

For preactivations and activations we also sum the changes from each step,

$$h_t^l := h_0^l + \theta_l(\delta h_1^l + \cdots + \delta h_t^l), \qquad x_t^l := x_0^l + \theta_l(\delta x_1^l + \cdots + \delta x_t^l).$$

Using the fact that

$$W_t^1 - W_{t-1}^1 = -\eta\tilde{\chi}_{t-1}\theta_{W^1} dh_{SAM,t-1}^1 \xi_{t-1}^T,$$

yields the normalized preactivation updates

$$\delta h_t^1(\xi) := -\eta\tilde{\chi}_{t-1} dh_{SAM,t-1}^1 \xi_{t-1}^T \xi \quad \text{(NL)},$$

with scaling $\theta_1 = \theta_{W^1} = n^{-c_1}\theta_{SAM} = n^{-c_1 - \min(b_{L+1}, c_{L+1}, d+d_{L+1})}$ as for SGD under perturbation stability and nontriviality where $\tilde{\chi}_{t-1} = \Theta(1)$.

For $l \in [2, L]$, it holds that

$$W_t^l - W_{t-1}^l = -\eta\tilde{\chi}_{t-1}\theta_{W^l}\frac{1}{n} dh_{SAM,t-1}^l (\tilde{x}_{t-1}^{l-1})^T,$$

with the right scaling $\theta_{W^l} = n^{1-c_l-\min(b_{L+1}, c_{L+1}, d+d_{L+1})}$ as for SGD under perturbation stability $\tilde{x}_{t-1}^{l-1} = \Theta(1)$, so that we get δh_t^l using a telescope sum,

$$\begin{aligned}
\theta_l \delta h_t^l = \quad & W_t^l x_t^{l-1} - W_{t-1}^l x_{t-1}^{l-1} = W_{t-1}^l (x_t^{l-1} - x_{t-1}^{l-1}) + (W_t^l - W_{t-1}^l) x_t^{l-1} \\
= \quad & \theta_{l-1}\left(W_0^l \delta x_t^{l-1} + \sum_{s=1}^{t-1} (W_s^l - W_{s-1}^l)\delta x_t^{l-1} \right) + (W_t^l - W_{t-1}^l) x_t^{l-1} \\
= \quad & \theta_{l-1}\left(W_0^l \delta x_t^{l-1} - \eta\theta_{W^l} \sum_{s=1}^{t-1} \tilde{\chi}_{s-1} \frac{(\tilde{x}_{s-1}^{l-1})^T \delta x_t^{l-1}}{n} dh_{SAM,s-1}^l \right)
\end{aligned}$$

$$-\eta\theta_{W^l}\tilde{\chi}_{t-1}\frac{(\tilde{x}_{t-1}^{l-1})^T x_t^{l-1}}{n}dh_{SAM,t-1}^l,$$

which defines a NonLin operation with the vectors $W_0^l\delta x_t^{l-1}, dh_{SAM,0}^l, dh_{SAM,t-1}^l$ and everything else treated as scalars. The scaling is given by

$$\theta_l = \max(\theta_{l-1}, \theta_{W^l}\theta_{l-1}, \theta_{W^l}) = \max_{m=1}^{l}\theta_{W^m} = n^{-r_l},$$

with

$$r_l := \min(b_{L+1}, c_{L+1}, d + d_{L+1}) + \min_{m=1}^{l}(c_m - \mathbb{I}(m \neq 1)),$$

where $\theta_{W^l} \leq 1$ for all $l \in [L]$ for stability. Note that for $l_1 \leq l_2$, it holds that $\theta_{l_1} \leq \theta_{l_2}$, which explains the sufficiency of $\theta_L = n^{-r_L} = n^{-r}$ for the stability of the activation updates.

Activations with the same scaling θ_l can then simply be defined via the NonLin operation

$$\delta x_t^l := \theta_l^{-1}(\phi(h_{t-1}^l + \theta_l\delta h_t^l) - \phi(h_{t-1}^l)).$$

The updates of the output function are scalars defined as

$$\delta f_t := \theta'_{L+1}\frac{\delta W_t^{L+1}x_t^L}{n} + \theta'_{L\nabla}\frac{\hat{W}_{t-1}^{L+1}\delta x_t^L}{n},$$

where $\theta'_{L+1} = n\theta_{L+1} = n^{1-c_{L+1}}$ and $\theta'_{L\nabla} = n\theta_\nabla\theta_L = n^{1-\min(b_{L+1},c_{L+1})-r_L}$, where we will see why $W_{t-1}^{L+1} = \Theta(n^{-\min(b_{L+1},c_{L+1})})$ in the next paragraph. This leads to the constraints $c_{L+1} \geq 1$ and $b_{L+1} + r \geq 1$ for the stability of the output function, where equality in either constraint leads to nontriviality.

E.1.5 t-th backward pass

Perturbations. Due to linearity and stability, the last layer remains

$$dx_t^L := \hat{W}_t^{L+1},$$

with scaling $\theta_\nabla = n^{-\min(b_{L+1},c_{L+1})}$.

As in the first backward pass, we use the weight updates to directly compute the preactivation and activation perturbations similar to the t-th forward pass but performing SGD instead of SAM in the last step. The SGD backward pass for the perturbation is given by

$$\begin{aligned}
dh_t^l &:= dx_t^l \odot \phi'(h_t^l),\\
dx_t^{l-1} &:= (W_t^l)^T dh_t^l\\
&= \left(W_0^l - \eta\theta_{W^l}\sum_{s=1}^{t}\tilde{\chi}_{s-1}\frac{1}{n}dh_{SAM,s-1}^l(\tilde{x}_{s-1}^{l-1})^T\right)^T dh_t^l\\
&= W_0^l dh_t^l - \eta(n^{1-c_l}\theta_{SAM}\theta_\nabla)\sum_{s=1}^{t}\tilde{\chi}_{s-1}\frac{(dh_{SAM,s-1}^l)^T dh_t^l}{n}\tilde{x}_{s-1}^{l-1},
\end{aligned}$$

with scaling $\max(\theta_\nabla, n^{1-c_l}\theta_{SAM}\theta_\nabla) = \theta_\nabla$, since $n^{1-c_l}\theta_{SAM} \leq 1$ is implied by $r \geq 0$ required for the stability of (pre-)activation updates.

We write $\chi_t = \mathcal{L}'(f_t(\xi_t), y_t)$ for the derivative of the loss with respect to the unperturbed function (which is $\Theta(1)$ under stability and nontriviality), and get

$$\begin{aligned}
\tilde{\delta}h_t^1(\xi) &:= +\frac{\rho\chi_t(\xi_t^T\xi)dh_t^1}{\|v_t\|},\\
\tilde{\theta}_l\tilde{\delta}h_t^l &:= \tilde{\theta}_{l-1}W_t^l\tilde{\delta}x_t^{l-1} + (\tilde{W}_t^l - W_t^l)\tilde{x}_t^{l-1}\\
&= \tilde{\theta}_{l-1}\left(W_0^l\tilde{\delta}x_t^{l-1} + \sum_{s=1}^{t}(W_s^l - W_{s-1}^l)\tilde{\delta}x_t^{l-1}\right) + (\tilde{W}_t^l - W_t^l)\tilde{x}_t^{l-1}
\end{aligned}$$

$$= \quad \tilde{\theta}_{l-1}\left(W_0^l\tilde{\delta}x_t^{l-1} - \eta(n^{1-c_l}\theta_{SAM})\sum_{s=1}^{t}\tilde{\chi}_{s-1}\frac{(\tilde{x}_{s-1}^{l-1})^T\tilde{\delta}x_t^{l-1}}{n}dh_{SAM,s-1}^l\right)$$
$$+\rho\tilde{\theta}_{W^l}\frac{\chi_t}{\|v_t\|}\frac{(x_t^{l-1})^T\tilde{x}_t^{l-1}}{n}dh_t^l,$$

which defines a NonLin operation with the vectors $W_0^l\tilde{\delta}x_t^{l-1}, dh_{SAM,0}^l, \ldots, dh_{SAM,t-1}^l, dh_t^l$, and where we can now define the definitive scalings $\tilde{\theta}_1 := \tilde{\theta}_{W^1} := n^{-(d+d_1)}\theta_\nabla = n^{-(\min(b_{L+1},c_{L+1})+d+d_1)}$, $\tilde{\theta}_{W^l} := n^{1-(d+d_l)}\theta_\nabla = n^{-(\min(b_{L+1},c_{L+1})+d+(d_l-1))}$ and $\tilde{\theta}_l = \max(\tilde{\theta}_{l-1}, n^{1-c_l}\theta_{SAM}\tilde{\theta}_{l-1}, \tilde{\theta}_{W^l}) = \max_{m=1}^l\tilde{\theta}_{W^m} = n^{-\tilde{r}_l}$ with

$$\tilde{r}_l := \min(b_{L+1}, c_{L+1}) + d + \min_{m=1}^{l}(d_m - \mathbb{I}(m \neq 1)),$$

where we used that $n^{1-c_l}\theta_{SAM} \leq 1$ due to $r \geq 0$ for stability and $\tilde{x}_t^{l-1} = \Theta(1)$ due to perturbation stability. Perturbation stability of the hidden layer (pre-)activations $\tilde{\delta}h^l, \tilde{\delta}x^l = O(1)$ for all $l \in [L]$ holds if and only if $\tilde{r} := \tilde{r}_L \geq 0$ since $\tilde{r}_l \geq \tilde{r}_L$ for all $l \leq L$.

The activation perturbations $\tilde{\delta}x_t^l$ and the perturbation of the output function $\tilde{\delta}f_t$ can be defined exactly as in the first backward pass,

$$\tilde{\delta}x_t^l := \quad \tilde{\theta}_l^{-1}(\phi(h_t^l + \tilde{\theta}_l\tilde{\delta}h_t^l) - \phi(h_t^l)),$$
$$\tilde{\delta}f_t := \quad \tilde{W}_t^{L+1}\tilde{x}_t^L - W_t^{L+1}x_t^L = \tilde{\theta}'_{L+1}\frac{\tilde{\delta}W_t^{L+1}\tilde{x}_t^L}{n} + \tilde{\theta}'_{L\nabla}\frac{\hat{W}_t^{L+1}\tilde{\delta}x_t^L}{n},$$

with $\tilde{\delta}W_t^{L+1} := \frac{\rho\,\chi_t\,x_t^L}{\|v_t\|}$ and the same scalings $\tilde{\theta}_l$, $\tilde{\theta}'_{L+1} = n^{1-(d+d_{L+1})}$ and $\tilde{\theta}'_{L\nabla} = n\theta_\nabla\tilde{\theta}_L = n^{1-\min(b_{L+1},c_{L+1})-\tilde{r}}$ since $W_t^{L+1} = W_0^{L+1} + \Delta W_t^{L+1} = \max(n^{-b_{L+1}}, n^{-c_{L+1}})$, which yields the slightly stronger constraint (than in the first backward pass) $\min(b_{L+1}, c_{L+1}) + \tilde{r} \geq 1$ for perturbation stability and either $\tilde{\theta}'_{L+1} = 1$ or $\min(b_{L+1}, c_{L+1}) + \tilde{r} = 1$ for perturbation nontriviality.

SAM Update. For each $l \in \{1, \ldots, L\}$, as in the first backward pass, we get

$$dx_{SAM,t}^L := \hat{W}_t^{L+1} + \tilde{\theta}_{(L+1)/\nabla}\,\tilde{\delta}W_t^{L+1},$$

with scaling $\theta_{SAM} = n^{-\min(b_{L+1},c_{L+1},d_{L+1}+1/2)}$ as well as

$$dh_{SAM,t}^l := dx_{SAM,t}^l \odot \phi'(\tilde{h}_t^l).$$

For $dx_{SAM,t}^l$ we again use a telescope sum over the weight changes,

$$dx_{SAM,t}^{l-1} := \quad (\tilde{W}_t^l)^T dh_{SAM,t}^l = (W_0^l + \theta_{W^l}\sum_{s=1}^{t}\delta W_s^l + \tilde{\theta}_{W^l}\tilde{\delta}W_t^l)^T dh_{SAM,t}^l$$
$$= \quad (W_0^l)^T dh_{SAM,t}^l - \eta(n^{1-c_l}\theta_{SAM})\sum_{s=1}^{t}\tilde{\chi}_{s-1}\frac{(dh_{SAM,s-1}^l)^T dh_{SAM,t}^l}{n}\tilde{x}_{s-1}^{l-1}$$
$$+\rho(n^{1/2-d_l}\theta_\nabla)\frac{\chi_t}{\|v_t\|}\frac{(dh_t^l)^T dh_{SAM,t}^l}{n}x_t^{l-1},$$

which defines a NonLin operation in the vectors $(W_0^l)^T dh_{SAM,t}^l, \tilde{x}_0^{l-1}, \ldots, \tilde{x}_{t-1}^{l-1}, x_t^{l-1}$ and everything else treated as scalars. Note that the scalings remain θ_{SAM}, since $\nabla_{x_t^{l-1}}f|_{\tilde{W}_t} = \Theta(\max(\theta_{SAM}, n^{1-c_l}\theta_{SAM}^2, n^{1/2-d_l}\theta_\nabla\theta_{SAM})) = \Theta(\theta_{SAM})$ under stability, nontriviality, perturbation stability and perturbation nontriviality.

Finally define the loss derivative on the perturbed output function

$$\tilde{\chi}_t := \mathcal{L}'(\tilde{W}_t^{L+1}\tilde{x}_t^L, y_t),$$

and compute the normalized change in W^{L+1},

$$\delta W_{t+1}^{L+1} := -\eta\tilde{\chi}_t\tilde{x}_t^L.$$

E.2 The infinite-width limit

In this section, we apply the Master Theorem's computation rules to derive the marginal distributions Z corresponding to the vectors of the program constructed above. According to the Master Theorem, each such vector z will have roughly iid coordinates distributed like Z^z in the large n limit.

We assume stability holds, so that $\theta \to \mathring{\theta} \in \{0, 1\}$ for all scalars θ in the program.

For the first forward pass, we have

$$Z^{h_0^1(\xi)} = \xi Z^{W_0^1}, \qquad Z^{x_0^l(\xi)} = \phi(Z^{h_0^l(\xi)}), \qquad Z^{h_0^{l+1}(\xi)} = Z^{W_0^{l+1} x_0^l(\xi)}.$$

If $b_{L+1} > 1/2$ then $\mathring{f}_0 = 0$, otherwise if $b_{L+1} = 1/2$ then $\mathring{f}_0$ converges to a nontrivial Gaussian. For the details we refer to Appendix H.4.1 in Yang and Hu (2021), as at initialization their results still hold here.

For the first SGD backward pass, we have

$$Z^{dx_0^L(\xi)} = Z^{\hat{W}_0^{L+1}}, \qquad Z^{dh_0^l(\xi)} = Z^{dx_0^l(\xi)} \phi'(Z^{h_0^l(\xi)}), \qquad Z^{dx_0^{l-1}(\xi)} = Z^{(W_0^l)^T dh_0^l(\xi)},$$

where $\dot{Z}^{dx_0^l(\xi)} = 0$ and $Z^{dx_0^l(\xi)} = \hat{Z}^{dx_0^l(\xi)}$ for all $\xi \in \mathcal{X}$.

For general $t > 0$, we have

$$\begin{aligned}
Z^{dx_t^L(\xi)} &= Z^{\hat{W}_t^{L+1}}, \\
Z^{dh_t^l(\xi)} &= Z^{dx_t^l(\xi)} \phi'(Z^{h_t^l(\xi)}), \\
Z^{dx_t^{l-1}(\xi)} &= Z^{(W_0^l)^T dh_t^l(\xi)} - \eta \mathring{\theta}_{W^l} \sum_{s=1}^{t} \mathring{\tilde{\chi}}_{s-1} \mathbb{E}[Z^{dh_{SAM,s-1}^l} Z^{dh_t^l}] Z^{\tilde{x}_{s-1}^{l-1}},
\end{aligned}$$

where $\mathring{\tilde{\chi}}_s = \mathcal{L}'(\mathring{\tilde{f}}_s(\xi_s), y_s)$ for $s < t$, and $Z^{(W_0^l)^T dh_t^l(\xi)}$ is a $\Theta(1)$ random variable distributed as

$$Z^{(W_0^l)^T dh_t^l(\xi)} = \hat{Z}^{(W_0^l)^T dh_t^l(\xi)} + \sum_{v \in \mathcal{V}:\, W_0^l v \in \mathcal{V}} Z^v \, \mathbb{E} \frac{\partial Z^{dh_t^l(\xi)}}{\partial \hat{Z}^{W_0^l v}}.$$

For all $t \geq 0$, the limit of the gradient norm is given by

$$\|\mathring{v}\| = \mathring{\chi}_t \left(\mathring{\theta}^2_{\|v^1\|} \mathbb{E}[Z^{(dh_t^1)^2}](\xi_t^T \xi_t) + \sum_{l=2}^{L} \mathring{\theta}^2_{\|v^l\|} \mathbb{E}[Z^{(dh_t^l)^2}] \mathbb{E}[Z^{(x_t^{l-1})^2}] + \mathring{\theta}^2_{\|v^{L+1}\|} \frac{(x_t^L)^T x_t^L}{n} \right)^{1/2}, \tag{E.3}$$

where $\mathring{\chi}_t = \mathcal{L}'(\mathring{f}_t(\xi_t), y_t)$, $\theta^2_{\|v^1\|} := n^{1-2d_1} \theta^2_\nabla$, $\theta^2_{\|v^l\|} := n^{2-2d_l} \theta^2_\nabla$ for $l \in [2, L]$ and $\theta^2_{\|v^{L+1}\|} := n^{1-2d_{L+1}}$, and where $\mathring{\theta}^2_{\|v^{L+1}\|} = 1$ if and only if $d_{L+1} = 1/2$ and $\mathring{\theta}^2_{\|v^{L+1}\|} = 0$ if and only if $d_{L+1} > 1/2$, while $\mathring{\theta}^2_{\|v^l\|} = 1$ if and only if $2d_l = 1 + \mathbb{I}(l > 1) - 2\min(b_{L+1}, c_{L+1})$ and $\mathring{\theta}^2_{\|v^l\|} = 0$ if and only if $2d_l > 1 + \mathbb{I}(l > 1) - 2\min(b_{L+1}, c_{L+1})$.

For the last-layer weight perturbations (for $\theta_\nabla \geq \tilde{\theta}_{L+1}$, else $Z^{\hat{\tilde{W}}_t^{L+1}} = Z^{\tilde{\delta} W_t^{L+1}}$) we have

$$Z^{\hat{\tilde{W}}_t^{L+1}} = Z^{\hat{W}_t^{L+1}} + \mathring{\tilde{\theta}}_{(L+1)/\nabla} Z^{\tilde{\delta} W_t^{L+1}}, \qquad Z^{\tilde{\delta} W_t^{L+1}} = \frac{\rho \, \mathring{\chi}_t}{\|\mathring{v}\|} Z^{x_t^L}.$$

Note that $\mathring{\chi}_t$ cancels itself out and we purely get a perturbation in distribution $Z^{x_t^L}$ scaled to have standard deviation ρ.

For all $t \geq 0$ and $l \in [1, L]$, we have

$$Z^{\tilde{h}_t^l} = Z^{h_t^l} + \mathring{\tilde{\theta}}_l Z^{\tilde{\delta} h_t^l}, \qquad Z^{\tilde{x}_t^l} = Z^{x_t^l} + \mathring{\tilde{\theta}}_l Z^{\tilde{\delta} x_t^l},$$

where for $l = 1$,

$$Z^{\tilde{\delta} h_t^1(\xi)} = + \frac{\rho \mathring{\chi}_t (\xi_t^T \xi)}{\|\mathring{v}\|} Z^{dh_t^1}.$$

If $\mathring{\tilde{\theta}}_l = 0$, then

$$Z^{\tilde{\delta}x_t^l} = \phi'(Z^{h_t^l}) Z^{\tilde{\delta}h_t^l},$$

otherwise $\mathring{\tilde{\theta}}_l = 1$ and

$$Z^{\tilde{\delta}x_t^l} = \phi(Z^{\tilde{h}_t^l}) - \phi(Z^{h_t^l}).$$

For $l \geq 2$, we have

$$\begin{aligned} Z^{\tilde{\delta}h_t^l} = \quad & \mathring{\tilde{\theta}}_{(l-1)/l}\, Z^{W_0^l \tilde{\delta}x_t^{l-1}} - \eta \mathring{\theta}_{W^l(\tilde{l}-1)/\tilde{l}} \sum_{s=1}^{t} \mathring{\tilde{\chi}}_{s-1} \mathbb{E}[Z^{\tilde{x}_{s-1}^{l-1}} Z^{\tilde{\delta}x_t^{l-1}}] Z^{dh_{SAM,s-1}^l} \\ & + \rho \mathring{\tilde{\theta}}_{W^l/l} \frac{\mathring{\chi}_t}{\|\mathring{v}\|} \mathbb{E}[Z^{x_t^{l-1}} Z^{\tilde{x}_t^{l-1}}] Z^{dh_t^l}, \end{aligned}$$

where $\tilde{\theta}_{(l-1)/l} = \frac{\tilde{\theta}_{l-1}}{\tilde{\theta}_l}$, $\theta_{W^l(\tilde{l}-1)/\tilde{l}} = \frac{\theta_{W^l}\tilde{\theta}_{l-1}}{\tilde{\theta}_l}$ and $\tilde{\theta}_{W^l/l} = \frac{\tilde{\theta}_{W^l}}{\tilde{\theta}_l}$, and $Z^{W_0^l \tilde{\delta}x_t^{l-1}}$ has the decomposition

$$Z^{W_0^l \tilde{\delta}x_t^{l-1}} = \hat{Z}^{W_0^l \tilde{\delta}x_t^{l-1}} + \sum_{v \in \mathcal{V}:\ (W_0^l)^T v \in \mathcal{V}} Z^v\, \mathbb{E} \frac{\partial Z^{\tilde{\delta}x_t^{l-1}}}{\partial \hat{Z}^{(W_0^l)^T v}}.$$

The perturbed output function has the limit $\mathring{\tilde{f}}_t := \mathring{f}_t + \mathring{\tilde{\delta}} f_t$ with

$$\mathring{\tilde{\delta}} f_t := \mathring{\tilde{\theta}}'_{L+1} \mathbb{E}[Z^{\tilde{\delta}W_t^{L+1}} Z^{\tilde{x}_t^L}] + \mathring{\tilde{\theta}}'_{L\nabla} \mathbb{E}[Z^{\hat{W}_t^{L+1}} Z^{\tilde{\delta}x_t^L}],$$

so that we can define $\mathring{\tilde{\chi}}_t = \mathcal{L}'(\mathring{\tilde{f}}_t(\xi_t), y_t)$ or equivalently $\mathring{\tilde{\chi}}_t = \mathcal{L}'(\mathring{\tilde{\theta}}_{L+1} \mathring{\tilde{\theta}}_L \mathbb{E}[Z^{\hat{\tilde{W}}_t^{L+1}} Z^{\tilde{x}_t^L}], y_t)$.
For the SAM gradients we have

$$\begin{aligned} Z^{dx_{SAM,t}^L} = \quad & Z^{\hat{W}_t^{L+1}} + \mathring{\tilde{\theta}}_{(L+1)/\nabla}\, Z^{\tilde{\delta}W_t^{L+1}}, \\ Z^{dh_{SAM,t}^l} = \quad & Z^{dx_{SAM,t}^l} \cdot \phi'(Z^{\tilde{h}_t^l}) \\ Z^{dx_{SAM,t}^{l-1}} = \quad & Z^{(W_0^l)^T dh_{SAM,t}^l} - \eta \mathring{\theta}_{W^l} \sum_{s=1}^{t} \mathring{\tilde{\chi}}_{s-1} \mathbb{E}[Z^{dh_{SAM,s-1}^l} Z^{dh_{SAM,t}^l}] Z^{\tilde{x}_{s-1}^{l-1}} \\ & + \rho \mathring{\tilde{\theta}}_{W^l} \frac{\mathring{\chi}_t}{\|\mathring{v}\|} \mathbb{E}[Z^{dh_t^l} Z^{dh_{SAM,t}^l}] Z^{x_t^{l-1}}, \end{aligned}$$

where $Z^{(W_0^l)^T dh_{SAM,t}^l}$ is given by

$$Z^{(W_0^l)^T dh_{SAM,t}^l} = \hat{Z}^{(W_0^l)^T dh_{SAM,t}^l} + \sum_{v \in \mathcal{V}:\ W_0^l v \in \mathcal{V}} Z^v\, \mathbb{E} \frac{\partial Z^{dh_{SAM,t}^l}}{\partial \hat{Z}^{W_0^l v}}.$$

Now SAM's (pre-)activation updates are given by

$$Z^{h_t^l} = Z^{h_0^l} + \mathring{\theta}_l (Z^{\delta h_1^l} + \cdots + Z^{\delta h_t^l}), \qquad Z^{x_t^l} = Z^{x_0^l} + \mathring{\theta}_l (Z^{\delta x_1^l} + \cdots + Z^{\delta x_t^l}),$$

with, for $l \in [2, L]$,

$$\begin{aligned} Z^{\delta h_t^1(\xi)} = \quad & -\eta \mathring{\tilde{\chi}}_{t-1} (\xi_{t-1}^T \xi) Z^{dh_{SAM,t-1}^1}, \\ Z^{\delta h_t^l} = \quad & \mathring{\theta}_{(l-1)/l} \left(Z^{W_0^l \delta x_t^{l-1}} - \eta \mathring{\theta}_{W^l} \sum_{s=1}^{t-1} \mathring{\tilde{\chi}}_{s-1} \mathbb{E}[Z^{\tilde{x}_{s-1}^{l-1}} Z^{\delta x_t^{l-1}}] Z^{dh_{SAM,s-1}^l} \right) \\ & - \eta \mathring{\theta}_{W^l/l} \mathring{\tilde{\chi}}_{t-1} \mathbb{E}[Z^{\tilde{x}_{t-1}^{l-1}} Z^{x_t^{l-1}}] Z^{dh_{SAM,t-1}^l}, \end{aligned}$$

where $\theta_{(l-1)/l} := \theta_{l-1}/\theta_l$, $\theta_{W^l/l} := \theta_{W^l}/\theta_l$ and $Z^{W_0^l \delta x_t^{l-1}}$ has the decomposition

$$Z^{W_0^l \delta x_t^{l-1}} = \hat{Z}^{W_0^l \delta x_t^{l-1}} + \sum_{v \in \mathcal{V}:\ (W_0^l)^T v \in \mathcal{V}} Z^v\, \mathbb{E} \frac{\partial Z^{\delta x_t^{l-1}}}{\partial \hat{Z}^{(W_0^l)^T v}}.$$

If $\mathring{\theta}_l = 0$, then

$$Z^{\delta x_t^l} = \phi'(Z^{h_{t-1}^l}) Z^{\delta h_t^l},$$

otherwise $\mathring{\theta}_l = 1$ and

$$Z^{\delta x_t^l} = \phi(Z^{h_t^l}) - \phi(Z^{h_{t-1}^l}).$$

The last-layer SAM weight update is given by

$$Z^{\hat{W}_t^{L+1}} = Z^{\hat{W}_0^{L+1}} + \mathring{\theta}_{L+1/\nabla}(Z^{\delta W_1^{L+1}} + \cdots + Z^{\delta W_t^{L+1}}),$$

with $Z^{\delta W_t^{L+1}} = -\eta \mathring{\tilde{\chi}}_{t-1} Z^{\tilde{x}_{t-1}^L}$.

For $t > 0$, the SAM function update is given by

$$\mathring{f}_t = \mathring{f}_0 + \mathring{\delta} f_1 + \cdots + \mathring{\delta} f_t,$$

with $\mathring{\delta} f_t = \mathring{\theta}'_{L+1} \mathbb{E}[Z^{\delta W_t^{L+1}} Z^{x_t^L}] + \mathring{\theta}'_{L\nabla} \mathbb{E}[Z^{\hat{W}_{t-1}^{L+1}} Z^{\delta x_t^L}]$.

E.3 Concluding the proof of all main results

After writing out the NE⊗OR⊤program and its limit, as well as tracking all scalings, the main results stated in Appendix D all follow from the Tensor Program Master Theorem and from the characterization results in Yang and Hu (2021) in the following way.

Formally Yang and Hu (2021) show feature learning for SGD with small enough learning rate $\eta > 0$ by proving $\partial_\eta^2 \mathbb{E}(Z^{x_1^L(\xi_0)})^2 \neq 0$ at $\eta = 0$, and they show that learning does not occur in the kernel regime by showing $\partial_\eta^3 \mathring{f}_1 \neq 0$, hence $\mathring{f}_1 - \mathring{f}_0$ is not linear in η.

Both $\mathbb{E}(Z^{x_1^L(\xi_0)})^2$ and $\mathring{f}_1$ are defined via NE⊗OR⊤computations and can be written as a composition of additions, multiplications, the expectation operator, applications of ϕ and ϕ', overall applications of infinitely differentiable, pseudo-Lipschitz functions to (Gaussian) random variables, η and ρ. Consequently $\mathbb{E}(Z^{x_1^L(\xi_0)})^2$ and $\mathring{f}_1$ are infinitely often differentiable as a function of both η and ρ, where differentiating the expectation operator is covered in Yang and Hu (2021, Lemma H.39). Since Yang and Hu (2021) cover the case $\rho = 0$, their proofs immediately show the correctness of the derived scalings for SAM as long as $\eta > 0$ and $\rho > 0$ are chosen small enough. Both the gradient evaluation for the perturbation as well as the gradient evaluation for the updates stay arbitrarily close to those of SGD if $\rho > 0$ is chosen small enough. The conditions for stability, nontriviality, feature learning, perturbation nontriviality and effective perturbations now follow from considering the respective scaling.

E.3.1 Proof of Theorem D.2

A *bcd*-parameterization is stable if and only if all scalings in the Tensor Program have the limit $\mathring{\theta} \in \{0, 1\}$, where $\mathring{\theta} = 1$ is required for activations at initialization (for which nothing changes compared to SGD). Potential cancellations are taken care of for sufficiently small $\eta > 0$ and $\rho > 0$ by the argument above. Now collecting all constraints that are already stated in the Tensor Program formulation at the respective step concludes the proof.

E.3.2 Proof of Theorem D.3

A stable *bcd*-parameterization is nontrivial if and only if $\mathring{f}_t = \Theta(1)$ if and only if $\mathring{\theta}'_{L+1} = 1$ or $\mathring{\theta}'_{L\nabla} = 1$.

E.3.3 Proof of Theorem D.4

A stable *bcd*-parametrization is feature learning in layer l if and only if the feature update scaling $\mathring{\theta}_l = 1$ where

$$\theta_l = n^{-r_l}, \quad r_l := \min(b_{L+1}, c_{L+1}, d_{L+1} + 1/2) + \min_{m=1}^{l}(c_m - \mathbb{I}(m \neq 1)).$$

Hence a stable *bcd*-parametrization is feature learning in layer l if and only if $r_l = 0$.

Since for all $l_1 \leq l_2$, it holds that $r_{l_1} \geq r_{l_2} \geq 0$, we get the equivalence for any $l_0 \in [L]$: A stable *bcd*-parametrization is feature learning in layer l_0 if and only if it is feature learning in layer l for all $l \geq l_0$ if and only if $r_{l_0} = 0$.

E.3.4 Proof of Theorem D.6

Given a stable *bcd*-parametrization, perturbation triviality is fulfilled if and only if $\mathring{\tilde{\theta}}'_{L+1} = 0$ and $\mathring{\tilde{\theta}}'_{L\nabla} = 0$, where $\tilde{\theta}'_{L+1} = n^{1/2-d_{L+1}}$ and $\tilde{\theta}'_{L\nabla} = n\theta_\nabla \tilde{\theta}_L = n^{1-\min(b_{L+1},c_{L+1})-\tilde{r}}$, hence if and only if $d_{L+1} > 1/2$ and $\min(b_{L+1}, c_{L+1}) + \tilde{r} > 1$.

In that case, $\mathring{\tilde{f}}_t = \mathring{f}_t$, but $\mathring{f}_t$ may still be affected by non-vanishing SAM perturbations in δW_t^{L+1} and δx_t^L. Only when all SAM perturbations vanish are we effectively only using SGD. By definition, the perturbation scale in the l-th layer vanishes if and only if $\mathring{\tilde{\theta}}_l = 0$, where $\tilde{\theta}_l = n^{-\tilde{r}_l}$ with $\tilde{r}_l = \min(b_{L+1}, c_{L+1}) + 1/2 + \min_{m=1}^{l}(d_m - \mathbb{I}(m \neq 1))$, hence if and only if $\tilde{r}_l > 0$. Since $\tilde{r}_l \geq \tilde{r}_L = \tilde{r}$ for all $l \leq L$, we get $\mathring{\tilde{\theta}}_l = 0$ for all $l \in [L]$ if and only if $\tilde{r} > 0$. Similarly, for any reference layer $l_0 \in [L]$, we get $\mathring{\tilde{\theta}}_l = 0$ for all $l \leq l_0$ if and only if $\tilde{r}_{l_0} > 0$. In words, for any $l_0 \in [L]$, we have vanishing perturbations in layer l_0 if and only if we have vanishing perturbations until layer l_0 if and only if $\tilde{r}_{l_0} > 0$.

Altogether, a stable *bcd*-parametrization has vanishing perturbations if and only if $\tilde{r} > 0$, $d_{L+1} > 1/2$ and $\min(b_{L+1}, c_{L+1}) + \tilde{r} > 1$. This case reduces to the results in Yang and Hu (2021) in the limit. Since stability requires $c_{L+1} \geq 1$ and $\tilde{r} \geq 0$, we can rewrite the equivalence conditions as $d_{L+1} \geq 1/2$ and $\tilde{r} > \max(0, 1 - b_{L+1})$.

E.3.5 Proof of Theorem D.8

Recall $\tilde{\theta}_{W^1} := n^{-(d+d_1)}\theta_\nabla$, $\tilde{\theta}_{W^l} := n^{1-(d+d_l)}\theta_\nabla$ and, for the last layer $\tilde{\theta}_{W^{L+1}} := n^{-(d+d_{L+1})}$.

As opposed to perturbation nontriviality, we are not only interested in $\tilde{\theta}_l = \max(\tilde{\theta}_{l-1}, \tilde{\theta}_{W^l}) = \max_{m=1}^{l} \tilde{\theta}_{W^m} \to 1$, but in a non-vanishing contribution of the perturbations in layer l, i.e. $\mathring{\tilde{\theta}}_{W^l} = 1$ or, for the last layer, $\mathring{\tilde{\theta}}_{L+1} = 1$.

E.3.6 Proof of Theorem D.9

The limit of the gradient norm is defined as a NE$\otimes$OR$\top$ program scalar (E.3). Note that for $b_{L+1} > 1/2$, the last-layer scaling strictly dominates all other scalings leading to the simplified gradient norm formula.

Now consider an arbitrary stable choice of layerwise initialization variances $\{b_l\}_{l\in[L+1]}$ and learning rates $\{c_l\}_{l\in[L+1]}$. To fulfill the gradient norm constraints (D.1), we have to choose $d_l = C = 1/2$ for all $l \in [L+1]$, because stability requires $\min(b_{L+1}, c_{L+1}) \geq 1/2$. Now stability of the output function perturbations requires $d \geq 1/2$, where $d > 1/2$ yields vanishing perturbations and $d = 1/2$ yields effective last-layer SAM through the term $\tilde{\delta} W_t^{L+1} \tilde{x}_t^L$. After choosing $d \geq 1/2$, we get $\tilde{r} \geq \min(b_{L+1}, c_{L+1}) \geq 1/2 > 0$ which implies vanishing perturbations in all hidden layers.

E.3.7 Proof of Proposition D.10

To achieve non-vanishing gradient norm contribution of the last layer in (D.1), we need to choose $d_{L+1} = 1/2$, which requires $d \geq 1/2$ for stability of the output function perturbations. Achieving non-vanishing gradient norm contributions of all layers requires $d_1 = 1/2 - \min(b_{L+1}, c_{L+1})$ and $d_l = 1 - \min(b_{L+1}, c_{L+1})$ for $l \in [2, L]$, which results in $\tilde{r} = d \geq 1/2 > 0$ which implies vanishing perturbations in all hidden layers.

E.3.8 Proof of Theorem D.11

Given a stable *bcd*-parametrization, we know $d + d_{L+1} \geq 1$, so that the feature learning constraint r is not affected by any stable choice of $d \cup \{d_l\}_{l\in[L+1]}$. The maximal stable choice of layerwise initialization variances $\{b_l\}_{l\in[L+1]}$ and learning rates $\{c_l\}_{l\in[L+1]}$ that constitute μP is therefore unaffected by the perturbation scalings $d \cup \{d_l\}_{l\in[L+1]}$.

Stability of the output function perturbations requires $b_{L+1} + \tilde{r} \geq 1$. Hence if $b_{L+1} < 1$, then $\tilde{r} \geq 1 - b_{L+1} > 0$, which implies vanishing perturbations in all hidden layers.

From now on consider $b_{L+1} \geq 1$. Recall $c_\nabla := \min(b_{L+1}, c_{L+1})$. In μP, $c_\nabla = 1$, but effective perturbations in all layers can be achieved more generally for $c_\nabla \geq 1$. Choosing $d_1 = 1/2 - c_\nabla$ saturates the gradient norm constraint (D.1). To reach effective perturbations already in the first layer $\tilde{r}_1 = c_\nabla + d + d_1 = 0$, we need $d = -1/2$. For perturbation stability and last-layer effective perturbations, we need $d + d_{L+1} = 1$ which requires $d_{L+1} = 3/2$. Achieving perturbation stability and effective perturbations in all hidden layers requires $\tilde{\theta}_{W^l} = 1$ which is equivalent to $c_\nabla + d + d_l - \mathbb{I}(l \neq 1) = 0$. For $l \in [2, L]$, we therefore need $d_l = 3/2 - c_\nabla$. This choice of $\{d_l\}_{l \in [L+1]}$ achieves effective perturbations in all layers.

To show uniqueness we iterate through all possibilities of saturating the norm bound constraint (D.1). We have considered the cases $d_{L+1} = 1/2$ in (b) leading to vanishing perturbations in all hidden layers and $d_1 = 1/2 - c_\nabla$ in (c) with only one choice for effective perturbations in all layers. Lastly consider $d_l = 1 - c_\nabla$ for $l \in [2, L]$ for non-vanishing gradient contribution of the hidden layers. Note that all hidden layers play the same role in all relevant constraints. Effective perturbations in any hidden layer $l \in [2, L]$ requires $\tilde{\theta}_{W^l} = 1$ for which we need $d = 0$. But then, as $d_1 \geq 1/2 - c_\nabla$, it holds that $\tilde{r}_1 \geq 1/2$ implying vanishing perturbations in the first layer. This shows the uniqueness of (1).

For the gradient norm statements, note that the gradient norm $\|v_t\|$ can be written as a NE$\otimes$OR$\top$computation rule (E.2) where the layer scalings in this parameterization are $\Theta(1)$ for the input layer, $\Theta(n^{-1/2})$ for hidden layers and $\Theta(n^{-1})$ for the output layer. Now the Tensor Program master theorem immediately implies the result.

E.3.9 Proof of Proposition D.12

Perturbation nontriviality with respect to any hidden layer is equivalent to $\tilde{r} = 0$. Since $\min(b_{L+1}, c_{L+1}) \leq 1$, we get $\min(b_{L+1}, c_{L+1}) + \tilde{r} \leq 1$. Since stability requires $\min(b_{L+1}, c_{L+1}) + \tilde{r} \geq 1$, we get $\min(b_{L+1}, c_{L+1}) + \tilde{r} = 1$, which implies perturbation nontriviality with respect to the output.

E.3.10 Proof of Proposition D.13

The constraint is the same constraint as in Theorem D.8, which implies effective perturbations in the first layer. Now $\tilde{r}_l \leq \tilde{r}_1 = 0$ implies perturbation nontriviality in all hidden layers due to Theorem D.6.

E.4 Analytic expression of the features after first SAM update

Below we state the analytic expression of the first SAM update, but leave a closer analysis of its fine-grained dynamics in comparison to SGD to future work. Before looking into the effective perturbation regime, we restate Lemma H.37 in Yang and Hu (2021) with a more detailed proof.

First, we define $\ell \in [L]$ as the unique index that satisfies $\theta_L = \cdots = \theta_\ell = 1 > \theta_{\ell-1} \geq \cdots \geq \theta_1$. In words, ℓ is the first layer in which feature learning occurs. Analogously, we define $\tilde{\ell} \in [L]$ as the unique index that satisfies $1 = \frac{\tilde{\theta}_L}{\theta_L} = \cdots = \frac{\tilde{\theta}_{\tilde{\ell}}}{\theta_L} > \frac{\tilde{\theta}_{\tilde{\ell}-1}}{\theta_L} \geq \cdots \geq \frac{\tilde{\theta}_1}{\theta_L}$.

Lemma E.2 (Features after first SGD step). *Defining $Z_t^l := Z^{h_t^l}$, $\gamma^l(\eta) = \mathbb{E}\phi(Z_0^l)\phi(Z_1^l)$ for $l \geq 1$, $\gamma^0 = \xi_0^T \xi$ and $\gamma_{11}^l(\eta) = \mathbb{E}\phi'(Z_0^l)\phi'(Z_1^l)$, we have*

$$Z_1^{\ell-1} = Z_0^{\ell-1}, \ldots, Z_1^1 = Z_0^1,$$

and, for all $l \geq \ell$,

$$Z_1^l = Z_0^l + \mathbb{I}_{l>\ell} \hat{Z}^{W_0^l \delta x_1^{l-1}} + \eta \beta^l Z^{dx_0^l} \phi'(Z_0^l),$$

where β^l is defined recursively by

$$\beta^l = \beta^l(\eta) = -\mathring{\chi}_0 \gamma^{l-1}(\eta) + \beta^{l-1}(\eta)\gamma_{11}^{l-1}(\eta),$$

with $\beta^{\ell-1} = 0$. Note that $\beta^l(0) < 0$ for all $l \geq \ell$.

Proof. By the defining infinite-width equations, assuming $\mathring{\theta}_{W^l/l} = 1$ (so minimal stable choice of c_l),

$$Z_1^l = Z_0^l + \mathring{\theta}_{(\ell-1)/\ell} Z^{W_0^l \delta x_1^{l-1}} - \eta \mathring{\chi}_0 \gamma^{l-1} Z^{dx_0^l} \phi'(Z_0^l).$$

At $l = \ell$, we get $\mathring{\theta}_{(\ell-1)/\ell} = 0$, whereas for $l > \ell$ we get $\mathring{\theta}_{(l-1)/l} = 1$, which results in $\mathring{\theta}_{(\ell-1)/\ell} = \mathbb{I}_{l>\ell}$.

Now, for $l > \ell$, the second term decomposes into $\hat{Z}^{W_0^l \delta x_1^{l-1}}$ and

$$\dot{Z}^{W_0^l \delta x_1^{l-1}} = Z^{dh_0^l} \mathbb{E} \frac{\partial Z^{\delta x_1^{l-1}}}{\partial \hat{Z}^{(W_0^l)^T dh_0^l}}.$$

Since by induction hypothesis,

$$Z^{\delta x_1^{l-1}} = \phi(Z_1^{l-1}) - \phi(Z_0^{l-1}) = \phi\left(Z_0^{l-1} + \mathbb{I}_{l>\ell} \hat{Z}^{W_0^l \delta x_1^{l-1}} + \eta \beta^{l-1} Z^{dx_0^{l-1}} \phi'(Z_0^{l-1})\right) - \phi(Z_0^{l-1}),$$

where $Z^{dx_0^{l-1}} = Z^{(W_0^l)^T dh_0^l}$ is the only dependence on $\hat{Z}^{(W_0^l)^T dh_0^l}$, we get

$$\frac{\partial Z^{\delta x_1^{l-1}}}{\partial \hat{Z}^{(W_0^l)^T dh_0^l}} = \phi'(Z_1^{l-1}) \eta \beta^{l-1} \phi'(Z_0^{l-1}).$$

Plugging the derivative back into the defining equation and noticing that $Z^{dh_0^l} = Z^{dx_0^l} \phi'(Z_0^l)$ concludes the proof. □

An analogous analysis for the perturbation at initialization shows.

Lemma E.3 (**Feature perturbation at initialization**). *The perturbation trivial layers fulfill*

$$Z^{\tilde{h}_0^{\tilde{\ell}-1}} = Z^{h_0^{\tilde{\ell}-1}}, \dots, Z^{\tilde{h}_0^1} = Z^{h_0^1},$$

and, for all $l \geq \tilde{\ell}$,

$$Z^{\tilde{h}_0^l} = Z^{h_0^l} + \mathbb{I}_{l>\tilde{\ell}} \hat{Z}^{W_0^l \tilde{\delta} x_0^{l-1}} + \rho \tilde{\beta}^l Z^{dx_0^l} \phi'(Z^{h_0^l}),$$

where $\tilde{\beta}^l$ independent of η is defined recursively by

$$\tilde{\beta}^l = \tilde{\beta}^l(\rho) = \frac{\mathring{\chi}_0}{\|\mathring{\nabla} L_0\|} \mathbb{E}[\phi(Z^{h_0^{l-1}}) \phi(Z^{\tilde{h}_0^{l-1}})] + \tilde{\beta}^{l-1} \mathbb{E}[\phi'(Z^{h_0^{l-1}}) \phi'(Z^{\tilde{h}_0^{l-1}})]$$

with $\tilde{\beta}^{\tilde{\ell}-1} = 0$. Note that $\tilde{\beta}^l(0) > 0$ for all $l \geq \tilde{\ell}$.

Remark E.4. If $\tilde{\ell} = 1$, in the definition of $\tilde{\beta}^l$ replace $\mathbb{E}[\phi(Z^{h_0^{l-1}}) \phi(Z^{\tilde{h}_0^{l-1}})]$ by $\xi_0^T \xi$. ◄

Now we are ready to state the closed form expression for the first SAM update.

Lemma E.5 (**Features after first SAM update**). *Defining $Z_t^l := Z^{h_t^l}$ and $\tilde{Z}_t^l := Z^{\tilde{h}_t^l}$, we have*

$$Z_1^{\ell-1} = Z_0^{\ell-1}, \dots, Z_1^1 = Z_0^1,$$

and, for all $l \geq \ell$,

$$Z_1^l = Z_0^l + \mathbb{I}_{l>\ell} \hat{Z}^{W_0^l \delta x_1^{l-1}} + \eta \beta^l Z^{dx_{SAM,0}^l} \phi'(\tilde{Z}_0^l) + \eta \gamma^l Z^{dh_0^l},$$

where β^l is defined recursively by

$$\beta^l = \beta^l(\eta) = -\mathring{\chi}_0 \mathbb{E}[\phi(\tilde{Z}_0^{l-1}) \phi(Z_1^{l-1})] + \beta^{l-1}(\eta) \mathbb{E}[\phi'(Z_1^{l-1}) \phi'(\tilde{Z}_0^{l-1})],$$

with $\beta^{\ell-1} = 0$, and $\gamma^l = \gamma^l(\eta)$ is recursively defined by

$$\gamma^l := \beta^{l-1} \rho \tilde{\beta}^{l-1} \mathbb{E}[\phi'(Z_1^{l-1}) \phi'(Z_0^{l-1}) \phi''(\tilde{Z}_0^{l-1}) Z^{dx_{SAM,0}^{l-1}}] + \gamma^{l-1} \mathbb{E}[\phi'(Z_0^{l-1}) \phi'(Z_1^{l-1})],$$

with $\gamma^{\ell-1} = \gamma^\ell = 0$.

Remark E.6. If $\ell = 1$, in the definition of β^l replace $\mathbb{E}[\phi(\tilde{Z}_0^{l-1}) \phi(Z_1^{l-1})]$ by $(\xi_{t-1}^T \xi)$. ◄

Proof. By the defining infinite-width equations, for $l \geq \ell$, assuming $\mathring{\theta}_{W^l/l} = 1$ (so minimal stable choice of c_l),

$$Z_1^l = Z_0^l + \mathring{\theta}_{(l-1)/l} Z^{W_0^l \delta x_1^{l-1}} - \eta \mathring{\chi}_0 \mathbb{E}[\phi(\tilde{Z}_0^{l-1})\phi(Z_1^{l-1})] Z^{dx_{SAM,0}^l \phi'(\tilde{Z}_0^l)}. \tag{E.4}$$

At $l = \ell$, we get $\mathring{\theta}_{(\ell-1)/\ell} = 0$ and $\mathring{\theta}_{W^\ell/\ell} = 1$, whereas for $l > \ell$ we get $\mathring{\theta}_{(l-1)/l} = 1$ and $\mathring{\theta}_{W^l/l} = 1$ (under minimal stable choice of c_l), which results in $\mathring{\theta}_{(l-1)/l} = \mathbb{I}_{l>\ell}$. Now, for $l > \ell$, the second term decomposes into $\hat{Z}^{W_0^l \delta x_1^{l-1}}$ and $\dot{Z}^{W_0^l \delta x_1^{l-1}}$. For the rest of the proof it remains to analyse $\dot{Z}^{W_0^l \delta x_1^{l-1}}$. Since by induction hypothesis,

$$\begin{aligned} Z^{\delta x_1^{l-1}} &= \phi(Z_1^{l-1}) - \phi(Z_0^{l-1}) \\ &= \phi\left(Z_0^{l-1} + \mathbb{I}_{l>\ell} \hat{Z}^{W_0^l \delta x_1^{l-1}} + \eta \beta^{l-1} Z^{dx_{SAM,0}^{l-1} \phi'(\tilde{Z}_0^{l-1})} + \eta \gamma^{l-1} Z^{dh_0^{l-1}}\right) - \phi(Z_0^{l-1}), \end{aligned}$$

where $Z^{dx_{SAM,0}^{l-1}} = Z^{(W_0^l)^T dh_{SAM,0}^l} + \rho \mathring{\tilde{\theta}}_{W^l} \frac{\mathring{\chi}_0}{\|\nabla \tilde{L}_0\|} \mathbb{E}[Z^{dh_0^l} Z^{dh_{SAM,0}^l}] Z^{x_0^{l-1}}$ with the second term independent of $(W_0^l)^T$ and by Lemma E.3 we know $\tilde{Z}_0^{l-1} = Z_0^{l-1} + \mathbb{I}_{l-1>\tilde{\ell}} \hat{Z}^{W_0^{l-1} \tilde{\delta} x_0^{l-2}} + \rho \tilde{\beta}^{l-1} Z^{dx_0^{l-1}} \phi'(Z_0^{l-1})$, where only the last term with $Z^{dx_0^{l-1}} = Z^{(W_0^l)^T dh_0^l}$ influences $\dot{Z}^{W_0^l \delta x_1^{l-1}}$, we get

$$\dot{Z}^{W_0^l \delta x_1^{l-1}} = Z^{dh_0^l} \mathbb{E} \frac{\partial Z^{\delta x_1^{l-1}}}{\partial \hat{Z}^{(W_0^l)^T dh_0^l}} + Z^{dh_{SAM,0}^l} \mathbb{E} \frac{\partial Z^{\delta x_1^{l-1}}}{\partial \hat{Z}^{(W_0^l)^T dh_{SAM,0}^l}}, \tag{E.5}$$

with

$$\frac{\partial Z^{\delta x_1^{l-1}}}{\partial \hat{Z}^{(W_0^l)^T dh_{SAM,0}^l}} = \phi'(Z_1^{l-1}) \eta \beta^{l-1} \phi'(\tilde{Z}_0^{l-1}),$$

and, using $Z^{dh_0^{l-1}} = Z^{dx_0^{l-1}} \phi'(Z_0^{l-1}) = Z^{(W_0^l)^T dh_0^l} \phi'(Z_0^{l-1})$, yields

$$\begin{aligned} \frac{\partial Z^{\delta x_1^{l-1}}}{\partial \hat{Z}^{(W_0^l)^T dh_0^l}} &= \phi'(Z_1^{l-1}) \left(\eta \beta^{l-1} Z^{dx_{SAM,0}^{l-1}} \phi''(\tilde{Z}_0^{l-1}) \frac{\partial \tilde{Z}_0^{l-1}}{\partial \hat{Z}^{(W_0^l)^T dh_0^l}} + \eta \gamma^{l-1} \phi'(Z_0^{l-1}) \right) \\ &= \phi'(Z_1^{l-1}) \left(\eta \beta^{l-1} Z^{dx_{SAM,0}^{l-1}} \phi''(\tilde{Z}_0^{l-1}) \rho \tilde{\beta}^{l-1} \phi'(Z_0^{l-1}) + \eta \gamma^{l-1} \phi'(Z_0^{l-1}) \right). \end{aligned}$$

Plugging Eq. (E.5) back into the defining equation (E.4) and noticing that $Z^{dh_{SAM,0}^l} = Z^{dx_{SAM,0}^l \phi'(\tilde{Z}_0^l)}$ as well as $Z^{dh_0^l} = Z^{dx_0^l \phi'(Z_0^l)}$ concludes the proof. □

F Generalizations and further perturbation scaling considerations

F.1 Overview over choices of d_l and d

Since for some combinations of architectures and datasets it turns out that performing SAM on a subset of layers performs better than effective perturbations in all layers (Müller et al., 2024), we would like to know how to choose d and d_l to adjust which layers should be effectively perturbed and which should have vanishing weight perturbations. In practice, simply set all perturbations that should vanish to 0 by design, and use the global scaling d and relative scalings d_l from μP^2 for the perturbed layers. This section is instead interested in a complete characterization of all possible choices of $\{d_l\}_{l \in [L+1]}$ and d. The heuristic derivation only requires the gradient norm constraints (D.1) and the perturbation stability constraints that require $\tilde{\delta} W^1 = O(1)$ and $\tilde{\delta} W^l = O(n^{-1})$ for $l > 1$ given by

$$d_l \geq \begin{cases} -c_\nabla - d & \text{if } l \text{ is input-like,} \\ 1 - c_\nabla - d & \text{if } l \text{ is hidden-like,} \\ 1 - d & \text{if } l \text{ is output-like,} \end{cases} \tag{F.1}$$

where a layer is effectively perturbed if and only if equality holds in the respective perturbation stability inequality. This heuristic claim yields the characterization of all phases of the choices of

	Effective perturbations possible			Gradient norm may be dominated by		
	input-like	hidden-like	output-like	input-like	hidden-like	output-like
$d = -1/2$	✓	✓	✓	✓	✗	✗
$d \in (-1/2, 0)$	✗	✓	✓	✓	✗	✗
$d = 0$	✗	✓	✓	✓	✓	✗
$d \in (0, 1/2)$	✗	✗	✓	✓	✓	✗
$d = 1/2$	✗	✗	✓	✓	✓	✓
$d > 1/2$	✗	✗	✗	✓	✓	✓

Table F.1: **(Characterization of perturbation scalings)** Overview over the regimes of all possible choices of d and d_l. A layer is effectively perturbed if and only d_l satisfies (F.1). At least one layer must satisfy equality in its gradient norm constraint (D.1). This table summarizes which layers can exhibit effective perturbations, and which may dominate the gradient norm, given a choice of d. The choice $d < -1/2$ results in perturbation blowup $\tilde{r} < 0$. At the critical $d = -1/2$ (respectively, $d = 0$; $d = 1/2$) a input-like (respectively hidden-like; output-like) layer is effectively perturbed if and only if it dominates the gradient norm. Consequently $d = -1/2$ implies effective perturbations in at least one input-like layer.

perturbation scalings d and d_l in Table F.1 and allows us to formulate a simple rule of how to choose d and d_l given the information which layers should be effectively perturbed, and which should have vanishing weight perturbations.

Choice of perturbation scaling from list of layers to effectively perturb. We denote the set of all layers by $\mathcal{L}$, whereas the subset of layers, which we want to effectively perturb, is denoted by $\mathcal{L}_{SAM} \subseteq \mathcal{L}$.

1. If there exists an input-like layer $l \in \mathcal{L}_{SAM}$, set $d = -1/2$. Input-like layers are effectively perturbed if and only if $d_l = 1/2 - c_\nabla$. Hidden-like (respectively, output-like) layers are effectively perturbed if and only if $d_l = 3/2 - c_\nabla$ (respectively, $d_l = 3/2$). For all layers that have vanishing weight perturbations, do not perturb these weights or choose $d_l > 1/2 - c_\nabla$ for input-like, $d_l > 3/2 - c_\nabla$ for hidden-like and $d_l > 3/2$ for output-like layers.
2. If all input-like layers should have vanishing weight perturbations but there exists a hidden-like layer $l \in \mathcal{L}_{SAM}$, set $d = 0$. Hidden-like layers are effectively perturbed if and only if $d_l = 1 - c_\nabla$. Output-like layers are effectively perturbed if and only if $d_{L+1} = 1$. For all layers that have vanishing weight perturbations, do not perturb these weights, or set $d_l > c_\nabla$ for input-like, $d_l > 1 - c_\nabla$ for hidden-like and $d_l > 1$ for output-like layers (as required by the perturbation stability and gradient norm constraints).
3. If both all input-like and all hidden-like layers have vanishing weight perturbations, but there exists some output-like layer $l \in \mathcal{L}_{SAM}$, then set $d = 1/2$. Output-like layers are effectively perturbed if and only if $d_l = 1/2$. For all layers that have vanishing weight perturbations, do not perturb these weights or set $d_l \geq 1/2 - c_\nabla$ for input-like, $d_l \geq 1 - c_\nabla$ for hidden-like and $d_l > 1/2$ for output-like layers (as required by the perturbation stability and gradient norm constraints).
4. If $\mathcal{L}_{SAM} = \emptyset$, then set $d > 1/2$ or simply perform SGD.

Example F.1 (**First-layer-only effective perturbations**)**.** Instead of simply using the rule set above, we derive the necessary choice of perturbation scaling from the scaling equalities and the norm constraints (D.1). To achieve first-layer effective perturbations, but trivial weight perturbations in all other layers, we need $\tilde{\theta}_{W^1} = 1$ and $\mathring{\tilde{\theta}}_{W^l} = 0$, for which we will choose $\tilde{\theta}_{W^l} = n^{-1}$. This requires setting

$$d_1 = -(c_\nabla + d), \qquad d_l = 2 - c_\nabla - d, \qquad d_{L+1} = 2 - d,$$

where one of the constraints (D.1) has to be fulfilled. Plugging the above d_l-choices into (D.1) results in the constraints $d \leq -1/2$, $d \leq 1$, $d \leq 3/2$, hence choose $d = -1/2$ so that only the first layer contributes non-vanishingly to the gradient norm. Note that $\tilde{r} = 0$ and output perturbation nontriviality holds if and only if $\min(b_{L+1}, c_{L+1}) = 1$ (as in μP). We apply this perturbation scaling in Appendix H.2 to show that propagating perturbations from early layers are not enough to inherit SAM's inductive bias that leads to improved generalization performance. ◀

F.2 Other ways to introduce layerwise perturbation scaling

Before presenting alternative ways how layerwise perturbation scaling could be accomplished, let us collect desirable properties that a definition should fulfill:

- Layerwise perturbation scaling should enable stable, effective perturbations in every layer.
- The perturbation step should require at most one additional forward and backward pass in each update step.
- The adapted optimization algorithm should recover the original (SAM) algorithm when not using layerwise perturbation scaling.

We start with the simplest case where the perturbations are normalized in each layer separately.

Remark F.2 (SAM with layerwise gradient normalization). For (SAM) with layerwise gradient normalization of the perturbations

$$\boldsymbol{\varepsilon}^l = \rho_l \cdot \nabla_{W^l}\mathcal{L}/\|\nabla_{W^l}\mathcal{L}\|, \tag{LN}$$

where $\|\cdot\|$ may denote the Frobenius or the spectral norm (equivalent under limited perturbation batch size), the spectral scaling condition $(*)$ immediately yields the correct layerwise perturbation scaling $\rho_l \overset{!}{=} \Theta(\sqrt{\texttt{fan\_out}/\texttt{fan\_in}})$. ◀

However, in practice, perturbations are usually globally normalized across layers, according to the GitHub repositories provided by Foret et al. (2021); Samuel (2022); Kwon et al. (2021); Andriushchenko and Flammarion (2022); Müller et al. (2024). Preliminary ablations in Appendix H.5 suggest that layer-coupled SAM with global normalization slightly outperforms SAM with layerwise gradient normalization. As our goal in this paper is to study (SAM) as applied in practice, we consider SAM with joint gradient normalization.

A first alternative to Definition 4 could scale perturbations after the joint gradient normalization. Opposed to Definition 4, for this variant the perturbation norm, i.e. the radius of the adversarial ascent ball, is not guaranteed to be ρn^{-d}, but $\rho(\sum_{l\in[L+1]}\rho_l^2)^{1/2}$. The correct perturbation scaling for this version more immediately follows from the condition that perturbations scale like updates.

Remark F.3 (Layerwise perturbation scaling after joint gradient normalization). For (SAM) with joint gradient normalization of the perturbations

$$\boldsymbol{\varepsilon}^l = \rho_l \cdot \frac{\nabla_{W^l}\mathcal{L}}{\|\nabla_{\mathbf{W}}\mathcal{L}\|},$$

the correct perturbation scaling in μP is given by $\rho_l \overset{!}{=} \Theta(n^{1/2}\cdot \texttt{fan\_out}/\texttt{fan\_in})$.

To understand this scaling rule, note that for $b_{L+1} > 1/2$ (such as in μP), the last layer always dominates the gradient norm (see Eq. (E.2) for the TP argument), resulting in the scaling

$$\|\nabla_{\mathbf{W}}\mathcal{L}\|_F \approx \mathcal{L}'(f_t(\xi_t), y_t)\|x^L\| = \Theta(n^{1/2}).$$

Thus, compared to SAM without gradient normalization (Appendix F.3), $\|\nabla_{\mathbf{W}}\mathcal{L}\|_F$ always contributes the scaling $n^{1/2}$. Noting that perturbations should scale like updates, and updates receive the layerwise learning rates $\eta_l \overset{!}{=} \Theta(\texttt{fan\_out}/\texttt{fan\_in})$ concludes the derivation. ◀

In Definition 4, we accept the additional layer-coupling complications that the layerwise gradient scaling before the joint gradient normalization entails in order to analytically control the perturbation radius to ρn^{-d}. To simplify the analysis as much as possible, we will first ensure width-independence of the normalization, so that the layerwise perturbation scaling is not affected by the normalization term. Then, layerwise perturbations should be scaled like updates.

Another alternative to layerwise perturbation scaling as in Definition 4 is motivated by the observation, that in μP^2 with Definition 4, only the first layer dominates the joint gradient norm (Theorem D.11). To let all layers contribute width-independently to the joint gradient norm, we can introduce even more hyperparameters (with limited benefit) by decoupling the numerator and denominator scalings in the perturbation. Opposed to Definition 4, the perturbation norm is again not analytically set with the choice of ρ, but may be smaller. Empirically, we do not observe performance differences due to denominator contribution scaling (Appendix H.4). This is the perturbation scaling we implement for ViTs (see Algorithm 1 for details).

Remark F.4 (**SAM with decoupled perturbation numerator and denominator scaling**). For (SAM) with perturbations

$$\boldsymbol{\varepsilon}^l = \rho n^{-d_l} \frac{\nabla_{W^l}\mathcal{L}}{\|v\|} \quad \text{with} \quad \|v\|^2 = \sum_{l=1}^{L+1} n^{-2\tilde{d}_l} \|\nabla_{W^l}\mathcal{L}\|^2, \tag{DP}$$

with layerwise perturbation radii $\rho \cdot n^{-d_l}$ and separate gradient norm scaling $n^{-\tilde{d}_l}$. ◀

In all alternatives, nontrivial layerwise perturbation scaling is necessary for effective perturbations in every layer, which necessarily changes the direction away from the original gradient direction. Such a layerwise gradient rescaling can also be achieved by adapting the architecture with width-dependent weight multipliers. The multipliers (a-μP^2) achieve effective perturbations without layerwise perturbation scaling such that all layers contribute non-vanishingly to the joint gradient norm. They rescale the gradients equivalently to (DP) when scaling all denominator terms to be width-independent. See Appendix F.6 for all details about weight multipliers.

Adapting the TP-based analysis. Our TP-based analysis covers all of the above perturbation scaling alternatives with minor adjustments. We just have to replace the normalized TP scalar (E.2). If we want to express $\|\nabla_{\mathbf{W}}\mathcal{L}\|_F$, we just drop all perturbation scaling terms n^{-d_l}. For the examples of (LN) and (DP), we replace (E.2) in each layer separately by the normalized TP scalars,

$$\|\nabla_{W^1}\mathcal{L}_t\| := \chi_t \left(\frac{(dh_t^1)^T dh_t^1}{n}(\xi_t^T \xi_t)\right)^{1/2},$$

with scaling $\theta_{\|\nabla_1\|} = n^{1/2}\theta_\nabla$ for the first layer, where θ_∇ is overloaded to denote $\theta_\nabla = n^{-b_{L+1}}$ in the first step and $\theta_\nabla = n^{-\min(b_{L+1}, c_{L+1})}$ in all later steps (in μP, $\theta_\nabla = n^{-1}$ always),

$$\|\nabla_{W^l}\mathcal{L}_t\| := \chi_t \left(\frac{(dh_t^l)^T dh_t^l}{n}\frac{(x_t^{l-1})^T x_t^{l-1}}{n}\right)^{1/2},$$

with scaling $\theta_{\|\nabla_{L+1}\|} = n\theta_\nabla$ for all hidden layers $l \in [2, L]$, and

$$\|\nabla_{W^{L+1}}\mathcal{L}_t\| := \chi_t \left(\frac{(x_t^L)^T x_t^L}{n}\right)^{1/2}.$$

with scaling $\sqrt{n}$ for the output layer, with respective normalized limits

$$\mathring{\chi}_t(\mathbb{E}[Z^{(dh_t^1)^2}](\xi_t^T\xi_t))^{1/2}, \quad \mathring{\chi}_t(\mathbb{E}[Z^{(dh_t^l)^2}]\mathbb{E}[Z^{(x_t^{l-1})^2}])^{1/2}, \quad \mathring{\chi}_t(\mathbb{E}[Z^{(x_t^L)^2}])^{1/2},$$

where $\mathring{\chi}_t = \mathcal{L}'(\mathring{f}_t(\xi_t), y_t)$.

The adapted scalings can then be tracked as before to derive the maximal stable layerwise perturbation scaling. Consider for example input layers in (LN). In μP, we know $\|\nabla_{W^1}\mathcal{L}_t\| = \Theta(n^{-1/2})$ and $\nabla_{W^1}\mathcal{L}_t = \Theta(n^{-1})$ entrywise. Effective perturbations are achieved with $\boldsymbol{\varepsilon}^1 = \Theta(1)$, so choose $\rho_l = n^{1/2}$ as expected from ($*$). Proceed similarly for all layers and perturbation scaling variants.

F.3 Extension to SAM without gradient normalization

Andriushchenko and Flammarion (2022) and Andriushchenko et al. (2023a) consider the SAM update without normalizing the gradient in the adversarial ascent. The corresponding update rule is given by

$$W_t = W_t - \eta\nabla_w\mathcal{L}(f(\xi_t; W_t + \rho v_t, y_t)), y_t), \qquad v_t = \nabla_w\mathcal{L}(f(\xi_t; W_t).$$

The NE⊗OR⊤ program for this update rule with arbitrary $v_t^l = n^{-c_l}\nabla_w\mathcal{L}(f(\xi_t; W_t)$ is also easily adapted from the above derivation. Just note that the gradient norm appears in an equation if and only if the perturbation radius ρn^{-d} appears. Without dividing by $\|v_t\|$, the parameter d becomes superfluous. Simply set $d = 0$ and remove the gradient norm constraints (D.1) to arrive at the NE⊗OR⊤ program and bcd-constraints for the update rule without gradient normalization.

Perturbation scaling d_l plays a similar role as learning rate scaling c_l as both scale similar gradients. We get effective perturbations in the l-th layer from the equation $d_l + \min(b_{L+1}, c_{L+1}) = c_l +$

$\min(b_{L+1}, c_{L+1}, d_{L+1})$ in μP, which yields $d_l = c_l$ for all $l \in [L]$ (since $d_{L+1} = 1$ for stability). **In particular, in μP, the correct layerwise perturbation scaling of unnormalized gradients is given by the rule $\frac{\text{fan out}}{\text{fan in}}$ or the squared weight (update) spectral norm $\|W^l\|_*^2$** (Yang et al., 2023a), which could be efficiently approximately tracked with a running power iteration.

Note that Dai et al. (2024) argue that the normalizing the gradients for the perturbation is crucial (in standard parametrization) due to a stabilizing effect and an enhanced drift along manifolds of minima. Monzio Compagnoni et al. (2023) find that unnormalized SAM gets stuck around saddles while SAM slowly escapes through additional Hessian-induced noise. This suggests that the additional effort of analysing the original SAM update rule with gradient normalization is necessary for practically useful theory. From this paper's point of view, the gradient normalization may be adding stability via the $n^{-1/2}$ contribution which allows to scale down ρ less aggressively in practice.

F.4 Extension to Adaptive SAM

Adaptive SAM (ASAM) (Kwon et al., 2021) is motivated by a sharpness definition that is invariant to parameter rescaling operators that leave the output function invariant, and can provide a further improvement over SAM of 0.5% to 1%, depending on the considered vision dataset and model (Müller et al., 2024). Here we consider the two examples of elementwise rescaling operators (with $p = 2$) and layerwise rescaling operators (with $p = 2$), which are the best performing SAM variant in most settings in Müller et al. (2024).

Proposition F.5. *Neither elementwise ASAM, which performs* (SAM) *but using the perturbation rule* (F.4)*, nor layerwise ASAM, which performs* (SAM) *but using the perturbation rule* (F.6)*, can be written as a* NE⊗OR⊤*program.*

Proof sketch. Elementwise ASAM requires an elementwise multiplication of matrices, and layerwise ASAM requires calculating the Frobenius norm of a matrix. A NonLin operation only takes vectors as arguments, so NE⊗OR⊤calculations with a matrix require its multiplication with a vector. But then a single coordinate of the resulting vector contains a mixture of an entire row of that matrix. Since we are only allowed to define random vectors and matrices, and the NE⊗OR⊤master theorem states that coordinates of NE⊗OR⊤vectors behave iid-like, this mixture cannot be disentangled by choosing a structured vector. Hence, already at initialization, the square of individual entries/the Frobenius norm of a random matrix cannot be exactly recovered by a function of matrix-vector products with NE⊗OR⊤vectors. □

Although ASAM is not formally covered by our theory, we still expect that the ASAM perturbations are correlated with the gradient and therefore with the incoming activations, so that heuristically we can still expect LLN-like behaviour and apply our scaling condition. If the perturbation rules still behave LLN-like, then Table 1 summarizes which layers are effectively perturbed under global scaling and provides the unique maximal perturbation scalings for all considered SAM variants. The correct perturbation scaling in μP for other perturbation rules that behave LLN-like can always be derived following the same steps:

1. In μP, it always holds that
$$W^l = \begin{cases} \Theta(1) & l = 1, \\ \Theta(n^{-1/2}) & l \in [2, L], \\ \Theta(n^{-1}) & l = L+1, \end{cases} \quad \text{and} \quad \nabla_{W^l}\mathcal{L} = \begin{cases} \Theta(\theta_\nabla) = \Theta(n^{-1}) & l \leq L, \\ \Theta(1) & l = L+1. \end{cases} \tag{F.2}$$
2. Assuming the normalization term in the denominator is scaled to $\Theta(1)$, track the layerwise scalings of the numerator. Maximal stable perturbations are always achieved with
$$\tilde{\delta}W_t^l = \begin{cases} \Theta(1) & l = 1, \\ \Theta(n^{-1}) & l > 1. \end{cases} \tag{F.3}$$
This yields constraints for achieving maximal stable perturbations in each layer.
3. Now replace the norm constraints (D.1) by tracking the scalings of each layer's contribution to the update rule's total normalization term.
4. To ensure normalization term scaling $\Theta(1)$, iterate through the layers l:

(a) choose d_l to satisfy its norm constraint,
(b) choose d to induce maximal stable perturbations in that layer,
(c) choose all other $d_{l'}$, $l' \neq l$, minimal to both satisfy its norm constraint as well as perturbation stability $\tilde{\delta}W_t^l = \begin{cases} O(1) & l = 1, \\ O(n^{-1}) & l > 1. \end{cases}$

5. From the above configurations, choose the unique one that yields maximal stable perturbations in all layers.[3]

F.4.1 Elementwise ASAM

If we want to be invariant to elementwise rescaling operators $T_w^l(x) = |W^l| \odot x$ where $x, W^l \in \mathbb{R}^{m\times n}$ and $\odot$ denotes elementwise multiplication, the resulting ASAM perturbation rule (where we introduce (layer-wise) perturbation scalings $\{d\} \cup \{d_l\}_{l\in[L+1]}$) replaces (LP) and is given by

$$\tilde{\delta}W_t^l := \rho n^{-d} \frac{n^{-d_l}|W^l| \odot |W^l| \odot \nabla_{W^l}\mathcal{L}(f(\xi_t; W_t), y_t)}{\|\nabla_{ASAM}^{elem}\|}, \tag{F.4}$$

with normalization

$$\|\nabla_{ASAM}^{elem}\| := \sum_{l=1}^{L+1} n^{-d_l} \left\| |W^l| \odot \nabla_{W^l}\mathcal{L}(f(\xi_t; W_t), y_t) \right\|_F,$$

where the absolute values $|W^l|$ are computed and multiplied elementwise. To find the correct perturbation scalings, we track the typical elementwise scaling of each quantity as before.

Elementwise ASAM in μP. In μP, the layerwise weights and gradients scale as (F.2). For $\|\nabla_{ASAM}^{elem}\| = O(1)$, we therefore replace the constraints (D.1) by the constraints

$$d_l \geq 1/2 - c_\nabla, \text{ for } l \in [L], \qquad d_{L+1} \geq -1/2, \tag{F.5}$$

where we can choose $\{d_l\}_{l\in[L+1]}$ to achieve equality in at least one constraint to achieve $\|\nabla_{ASAM}^{elem}\| = \Theta(1)$.

The layerwise perturbations scale as $\tilde{\delta}W_t^l = n^{-d} \begin{cases} \Theta(n^{-d_1}\theta_\nabla) & l = 1, \\ \Theta(n^{-1-d_l}\theta_\nabla) & l \in [2, L], \\ \Theta(n^{-d_{L+1}}n^{-2}) & l = L+1. \end{cases}$

Stable and nontrivial perturbations in each layer are achieved under condition (F.3), which induces the constraints for optimal layerwise perturbation scaling

$$d + d_l = -c_\nabla, \text{ for } l \in [L], \qquad d + d_{L+1} = -1.$$

Irrespective which of the above norm constraints (F.5) we satisfy, we need $d = -1/2$ to achieve optimal layerwise perturbation scaling. Hence $d = d_{L+1} = -1/2$ and $d_l = 1/2 - c_\nabla$ for $l \in [L]$ is the unique choice of $\{d\} \cup \{d_l\}_{l\in[L+1]}$ modulo norm scaling equivalence that achieves $\Theta(1)$ perturbation scaling in all layers. With this choice all layers contribute non-vanishingly to the gradient norm. In μP $c_\nabla = 1$, so that $d_l = -1/2$ for all $l \in [L+1]$, so that ASAM does not require layerwise rescaling of the gradients, but upscaling of the perturbation by $n^{1/2}$ to achieve nontrivial perturbations in any layer. This may explain why ASAM often outperforms SAM in large models: By only requiring global scaling, ASAM achieves maximal stable perturbations in all layers if the perturbation radius is tuned globally at every width.

If instead of a global gradient norm $\|\nabla_{ASAM}^{elem}\|$, one would want to normalize in each layer separately with $\|\nabla_{ASAM}^{elem,l}\| := n^{-d_l}\||W^l| \odot \nabla_{W^l}\mathcal{L}(f(\xi_t; W_t), y_t)\|_F$, the layerwise perturbation scalings become $\tilde{\delta}W_t^l = n^{-d} \begin{cases} \Theta(n^{-1/2}) & l = 1, \\ \Theta(n^{-3/2}) & l > 1. \end{cases}$ Again, to achieve maximal stable perturbations in all layers we need $d = -1/2$ and no layerwise adaptation of the gradient norm.

[3] There must exist such a choice of $\{d_l\}_{l\in[L+1]}$ and d, because $\{d_l\}_{l\in[L+1]}$ allow to set any relative scalings between layers and d determines the global scaling, which overall allows to set all possible layerwise scalings. Any deviation from the choice that achieves effective perturbations in all layers either results in blowup or a vanishing effect of the weight perturbation in some layer. This shows uniqueness.

F.4.2 Layerwise ASAM

ASAM with layerwise rescaling as in Müller et al. (2024) employs the layerwise transformations $T_w^l(x) = \|W^l\|_F \cdot x$. This ASAM perturbation rule replaces (LP) and is given by

$$\tilde{\delta} W_t^l := \rho n^{-d} \frac{n^{-d_l} \|W^l\|_F^2 \nabla_{W^l} \mathcal{L}(f(\xi_t; W_t), y_t)}{\|\nabla_{ASAM}^{layer}\|}, \tag{F.6}$$

with normalization

$$\|\nabla_{ASAM}^{layer}\| := \sum_{l=1}^{L+1} n^{-d_l} \|W^l\|_F \|\nabla_{W^l} \mathcal{L}(f(\xi_t; W_t), y_t)\|_F.$$

Layerwise ASAM in μP. In μP, we have $\|W^l\|_F = \begin{cases} \Theta(n^{1/2}) & l = 1, \\ \Theta(n^{1/2}) & l \in [2, L], \\ \Theta(n^{-1/2}) & l = L+1. \end{cases}$

Hence, the norm constraints (D.1) are now replaced by

$$d_1 \geq 1 - c_\nabla, \qquad d_l \geq 3/2 - c_\nabla \quad \text{for } l \in [2, L], \qquad d_{L+1} \geq 0.$$

The scale of the perturbation numerator now scales as $\tilde{\delta} W_t^l = n^{-d} \begin{cases} \Theta(n^{-d_1} n \theta_\nabla) & l = 1, \\ \Theta(n^{-d_l} n \theta_\nabla) & l \in [2, L], \\ \Theta(n^{-d_{L+1}} n^{-1}) & l = L+1. \end{cases}$

In μP, achieving maximal stable perturbations (F.3) is therefore equivalent to satisfying the constraints

$$d + d_1 = 0, \qquad d + d_l = 1 \quad \text{for } l \in [2, L], \qquad d + d_{L+1} = 0.$$

Now we can simultaneously satisfy the first- and last-layer norm constraints with $d_1 = 0$ and $d_{L+1} = 0$, while achieving effective perturbations in all layers with $d = 0$ and $d_l = 1$. Satisfying the norm constraint in the hidden layers with $d_l = 1/2$ would imply vanishing perturbations in the first and last layer (by requiring $d \geq 1/2$).

F.5 Representing general architectures and adaptive optimizers as Tensor Programs

Here, we lay out explicitly how to write some of the building blocks in ResNets and ViTs in a Tensor Program and provide further scaling considerations. According to Yang and Hu (2021), it is straightforward to generalize scaling conditions that induce feature learning in MLPs to these other common neural network building blocks. Since perturbations should always scale like updates, the conditions for stable feature learning and those for stable effective perturbations are analogous.

One potential complication in the case of SAM would be a contribution to the joint gradient normalization $\|v_t\|$ that differs from the classical input, hidden or output layer contribution. But we will see that these contributions do not differ for any of the considered layer types.

Layernorm. The Layernorm operation is defined as

$$h_t^{l+1} = \gamma_t^l \frac{x_t^l - \nu_t^l}{\sigma_t^l + \varepsilon} + \beta_t^l,$$

where $\varepsilon > 0$ is a small positive constant, γ_t^l, β_t^l are learnable parameters and $\nu_t^l = \frac{1}{n} \sum_{i=1}^n (x_t^l)_i$ is an Avg operation as in Yang and Littwin (2023, Def. 2.6.1) and $\sigma_t^l = \sqrt{\frac{1}{n} \sum_{i=1}^n (x_t^l - \nu_t^l)^2}$ is a composition of Nonlin, Avg and Nonlin. The parameters γ_t^l, β_t^l can be seen as input weights to the input 1. They should be initialized as $\gamma_0^l = 1$ and $\beta_0^l = 0$. In the forward pass, the layernorm preserves stability $h_t^{l+1} = \Theta(1)$ when $\gamma_t^l + \beta_t^l = \Theta(1)$ except for the Lebesgue nullset of learning rates for which they exactly cancel each other out. Recall the notation $dz = \theta_z^{-1} \partial f / \partial z$, where $\theta_z = n^C$ for some $C \in \mathbb{R}$ denotes the width-dependent scaling. The derivatives are

$$d\beta_t^l = dh_t^{l+1}, \qquad d\gamma_t^l = dh_t^{l+1} \frac{x_t^l - \nu_t^l}{\sigma_t^l + \varepsilon}.$$

These gradients coincide both in shape and scaling with the scaling we expect for an input layer, resulting in the same gradient spectral/Frobenius norm scaling. Continuing the backward pass, using $\frac{\partial \sigma_t^l}{\partial x_t^l} = \frac{x_t^l - \nu_t^l}{n\sigma_t^l}$, we get

$$\begin{aligned} dx_t^l = & \quad dh_t^{l+1}\gamma_t^l \left(\frac{1}{\sigma_t^l + \varepsilon}(I - \frac{1}{n}) - \frac{x_t^l - \nu_t^l}{(\sigma_t^l + \varepsilon)^2} \frac{\partial \sigma_t^l}{\partial x_t^l} \right) \\ = & \quad dh_t^{l+1}\gamma_t^l \left(\frac{1}{\sigma_t^l + \varepsilon}(I - \frac{1}{n}11^T) - \frac{x_t^l - \nu_t^l}{(\sigma_t^l + \varepsilon)^2} \frac{(x_t^l - \nu_t^l)^T}{n\sigma_t^l} \right), \end{aligned}$$

which preserves the order as long as $\gamma_t^l = \Theta(1)$, since $x_t^l = \Theta(1)$, we know $\nu_t^l, \sigma_t^l = \Theta(1)$.

Note that Layernorm removes the necessity to avoid blowup in the activations x_t^l in the forward pass (ignoring potential numerical issues), and always rescales to $\Theta(\max(\gamma_t^l, \beta_t^l))$. However, in the backward pass, a scaling $x_t^l = \Theta(n^c)$, with $c > 0$, results in $dx_t^l = \Theta(n^{-c} dh_t^{l+1}\gamma_t^l)$, hence vanishing gradients. The gradients would only stabilize if $\phi'(h_t^l) = \Theta(h_t^l)$, but no popular activation function has a scale equivariant derivative. Yang (2019) shows how to write Batchnorm and Average Pooling as a Tensor Program.

Convolutions. Convolutional layers can be seen as a collection of dense weight matrices where width corresponds to the number of channels (Yang, 2019). With kernel positions ker, input channels $[n^l]$ and output channels $[n^{l+1}]$, the weights of a stride-1 convolution are given by $\{W_{i\alpha\beta}^l\}_{i \in ker, \alpha \in [n^{l+1}], \beta \in [n^l]}$, so that for each $i \in ker$, $W_i^l \in \mathbb{R}^{n^{l+1} \times n^l}$ is a dense matrix. With $\{x_{i\alpha}^l\}_{i \in pos^l, \alpha \in [n^l]}$, the convolution operation is given by

$$(W^l * x)_{i\alpha} = \sum_{\beta, j : j + i \in pos^l} W_{j\alpha\beta}^l x_{i+j,\beta}^l,$$

which performs MatMul and Avg and where ker, pos^l are assumed to be of fixed size. For ker of fixed size, convolutional weights scale like hidden layer weight matrices, also in Frobenius norm contributing to $\|v_t\|$.

Residual connections. A residual connection propagates the current activation forward, skipping an arbitrarily complex nonlinear block $f_t^l : \mathbb{R}^{n_l} \to \mathbb{R}^{n_{l+1}}$ in between, where f_t^l can depend on time-dependent parameters like a weight matrix. The forward pass can be written as

$$x_t^l = x_t^{l-1} + f_t^l(x_t^{l-1}).$$

If $x_t^l = \Theta(1)$ for all layers l holds in the model without residual connections, it also holds in the model with residual connections. At fixed depth, $f_t^l = o(1)$ should be avoided, as it would hold that $x_t^{l+1} = x_t^l$ in the infinite-width limit and the layer would be superfluous. The derivative of the activations becomes

$$dx_t^{l-1} = dx_t^l + dx_t^l \frac{\partial f_t^l}{\partial x_t^{l-1}},$$

where the second term stays the same as without the residual connection. For the example of f_t^l being a fully connected layer we get $dx_t^{l-1} = dx_t^l + (W_t^l)^T \left(dx_t^l \odot \phi'(W_t^l x_t^{l-1}) \right)$. In this example, the derivative with respect to the weights becomes

$$\frac{\partial f_t}{\partial W_t^l} = dx_t^l \frac{\partial x_t^l}{\partial W_t^l} = dx_t^l \frac{\partial f_t^l}{\partial W_t^l} = (dx_t^l \odot \phi'(W_t^l x_t^{l-1}))(x_t^{l-1})^T,$$

where the residual connection does not alter the functional dependence on dx_t^l and x_t^l compared to a MLP, but implicitly influences the weight gradient since dx_t^l and x_t^l are altered. As for the forward pass, the gradient scaling dx_t^l gets stabilized in the backward pass so that $\frac{\partial f_t^l}{\partial x_t^{l-1}}$ is now allowed to be vanishing with width. Again, we are not aware of an architecture in which that would be desirable. Since a residual connection does not introduce learnable parameters, it interferes in $\|v_t\|$ only implicitly through the stabilized gradients in earlier layers, which can contribute non-vanishingly to $\|v_t\|$ even if later layers are wrongly scaled and their scaling is not adapted.

Adam as a base optimizer. When using Adam or similar adaptive optimizers as a base optimizer, the learning rate should scale as $\Theta(1)$ for input-like layers and biases, and $\Theta(n^{-1})$ for hidden and output

layers (Yang et al., 2022). Yang et al. (2023b) provide proofs for arbitrary optimizers that perform generalized, nonlinear outer products. In the example of Adam, the update rule can be written as

$$\phi(u_\alpha^1, \dots, u_\alpha^k, v_\beta^1, \dots, v_\beta^k) = \sum_i \gamma_i u_\alpha^i v_\beta^k / \left(\sum_i \omega_i (u_\alpha^i v_\beta^i)^2 \right)^{1/2},$$

where γ_i, ω_i are the weights that stem from the moving averages. By using a learning rate of n^{-1} and using the fact that both u and v have approximately iid coordinates of order $\Theta(1)$, the law of large numbers yields $\Theta(1)$ updates of the form

$$\frac{1}{n} \sum_{\beta=1}^{n} \phi(u_\alpha^1, \dots, u_\alpha^k, v_\beta^1, \dots, v_\beta^k) x_\beta = \mathbb{E}\phi(u_\alpha^1, \dots, u_\alpha^k, Z^{v^1}, \dots, Z^{v^k}) Z^x.$$

Any other learning rate scaling would either result in blowup or vanishing updates.

Adaptive optimizers have not been used for the ascent/perturbation step. In the descent/update step, nothing changes compared to unperturbed optimization as long as we ensure stable perturbations.

F.6 Influence of width-dependent weight multipliers on bcd-parameterizations

Our definition of bcd-parameterizations is convenient because it purely adapts the learning algorithm but not the architecture. **We can also adapt the architecture by using layerwise width-dependent weight multipliers to effectively perturb all layers without any perturbation scaling.** The reason is that layerwise weight multipliers scale the layerwise gradients. Here, we study how the introduction of weight multipliers affects bcd-parameterizations.

In this section, we consider L-hidden layer MLPs with weight multipliers $\{a_l\}_{l\in[L+1]}$, width $n \in \mathbb{N}$, inputs $\xi \in \mathbb{R}^{d_{\text{in}}}$, and with outputs $f(\xi) := n^{-a_{L+1}} W^{L+1} x^L(\xi)$ where the activations $x^L(\xi)$ are defined via the iteration

$$h^1(\xi) := n^{-a_1} W^1 \xi, \qquad x^l(\xi) := \phi(h^l(\xi)), \qquad h^{l+1}(\xi) := n^{-a_{l+1}} W^{l+1} x^l(\xi).$$

We define $abcd$-parameterizations in the same way as bcd-parameterizations, but instead of MLPs we use MLPs with weight multipliers $\{a_l\}_{l\in[L+1]}$.

Definition F.6 ($abcd$**-parametrization).** An *$abcd$-parametrization* $\{a_l\}_{l\in[L+1]} \cup \{b_l\}_{l\in[L+1]} \cup \{c_l\}_{l\in[L+1]} \cup \{d_l\}_{l\in[L+1]} \cup \{d\}$ defines the training of an MLP with weight multipliers $\{a_l\}_{l\in[L+1]}$ with SAM in the following way:

(a) Initialize weights iid as $W_{ij}^l \sim \mathcal{N}(0, n^{-2b_l})$.
(b) Train the weights using the SAM update rule with layerwise learning rates,

$$W_{t+1}^l = W_t^l - \eta n^{-c_l} \nabla_{W^l} \mathcal{L}\left(f\left(\xi_t; W_t + \boldsymbol{\varepsilon}_t\right), y_t\right),$$

with the scaled perturbation $\boldsymbol{\varepsilon}_t$ via layerwise perturbation radii,

$$\boldsymbol{\varepsilon}_t := \rho n^{-d} \frac{v_t}{\|v_t\|}, \quad \text{with} \quad v_t = (v_t^1, \dots, v_t^{L+1}), \quad v_t^l := n^{-d_l} \cdot \nabla_{W^l} \mathcal{L}(f(\xi_t; W_t), y_t), \text{ (LP)}$$

W.l.o.g. we set $\|v_t\| = \Theta(1)$, which prevents nontrivial width-dependence from the denominator. This imposes the constraints:

$$d_1 + a_1 \geq 1/2 - c_\nabla, \qquad d_l + a_l \geq 1 - c_\nabla, \qquad d_{L+1} + a_{L+1} \geq 1/2,$$

with at least one equality required to hold, where $l \in [2, L]$, and where $\nabla_{x^L} f = n^{-a_{L+1}} W^{L+1} = \Theta(n^{-c_\nabla})$ with $c_\nabla = \min(b_{L+1} + a_{L+1}, c_{L+1} + 2a_{L+1})$. The normalization $v_t/\|v_t\|$ removes one degree of freedom from $\{d_l\}_{l\in[L+1]}$ via the equivalence $\{d_l'\}_{l\in[L+1]} \cong \{d_l\}_{l\in[L+1]}$ iff there exists a $C \in \mathbb{R}$ such that $d_l' = d_l + C$ for all $l \in [L+1]$. ◀

F.6.1 $abcd$-equivalence classes

Update scalings behave as in SGD. The weight multiplier n^{-a_l} scales the gradient $\nabla_{W^l} f$ by n^{-a_l}. In the following forward pass, another multiplication of the weight updates with n^{-a_l} leads to the activation update scaling n^{-2a_l}. This can be counteracted by adapting the learning rate scaling. For

abc-parameterizations and SGD training, this induces the layerwise equivalence between parameterizations with (a_l, b_l, c_l) or with $(a_l + \theta_l, b_l - \theta_l, c_l - 2\theta_l)$. The extension of all of our results to Adam as a base optimizer is straightforward, since learning rate scalings and perturbation scalings are decoupled. For Adam, c_l should be adapted to $c_l - \theta_l$.

Again, perturbations with joint gradient normalization complicate matters compared to SGD and Adam. Keeping the gradient norm scalings invariant under $a_l \mapsto a_l + \theta$ would require $d_l \mapsto d_l - \theta$, but keeping the activation perturbation scaling invariant would require $d_l \mapsto d_l - 2\theta$ as for updates. Consequently, an exact equivalence between $abcd$-parameterizations at finite width requires θ to be the same for all layers and the conflicting gradient norm in the denominator and perturbation scaling in the numerator to be accounted for by adapting the global perturbation scaling $d \mapsto d - \theta$ (together with $d_l \mapsto d_l - \theta$). In other words, (SAM) with layer-coupling gradient normalization (LP) does not have layerwise analytical equivalence classes at finite width. Below, we provide two alternative perturbation rules that resolve these complications and recover layerwise equivalence classes. The following lemma formally states the layer-coupled equivalence relation for the perturbation rule (LP). All proofs are provided at the end of this section.

Lemma F.7 ($abcd$**-equivalence classes).** *Let $f_t(\xi)$ denote the output of a MLP in a stable $abcd$-parameterization with weight multipliers $\{a_l\}_{l\in[L+1]}$ after t steps of training with the SAM update rule with layerwise perturbation scaling (LP) using a fixed sequence of batches and evaluated on input ξ. Then for any $\theta \in \mathbb{R}$ and any $C \in \mathbb{R}$, $f_t(\xi)$ stays fixed for all t and ξ if, for all $l \in [L+1]$,*

$$(a_l, b_l, c_l, d_l, d) \text{ is reparameterized to } (a_l + \theta, b_l - \theta, c_l - 2\theta, d_l - \theta + C, d - \theta).$$

Remark F.8 (Infinite-width equivalences). In the infinite-width limit, $abcd$-parameterizations remain equivalent under $(a_l + \theta_l, b_l - \theta_l, c_l - 2\theta_l, d_l - 2\theta_l, d)$ layerwise as long as the set of layers that contribute to the gradient norm non-vanishingly remains invariant. The gradient norm constraints for $\|v^l\| = O(1)$ become

$$d_1 + a_1 \geq 1/2 - c_\nabla, \qquad d_l + a_l \geq 1 - c_\nabla, \qquad d_{L+1} + a_{L+1} \geq 1/2,$$

where $\nabla_{x^L} f = n^{-a_{L+1}} W^{L+1} = \Theta(n^{-c_\nabla})$ with $c_\nabla = \min(b_{L+1} + a_{L+1}, c_{L+1} + 2a_{L+1})$ remains invariant under equivalence transformations. ◀

Remark F.9 (SAM with layerwise gradient normalization). As the layer coupling is induced by the joint gradient normalization in the perturbations, layerwise gradient normalization simplifies the analysis. For (SAM) with layerwise gradient normalization (LN) of the perturbations global perturbation scaling d is superfluous, and there exist layerwise equivalence classes: For any $\{\theta_l\}_{l\in[L+1]} \subset \mathbb{R}$,

$$(a_l, b_l, c_l, d_l) \text{ is equivalent to } (a_l + \theta_l, b_l - \theta_l, c_l - 2\theta_l, d_l - \theta_l).$$

To understand this equivalence, observe that any layerwise gradient scaling is cancelled out by the normalization $\nabla_{W^l}\mathcal{L}/\|\nabla_{W^l}\mathcal{L}\|$. Only the n^{-a_l} factor from subsequent forward passes has to be counteracted. ◀

Remark F.10 (SAM with decoupled perturbation numerator and denominator scaling). A perturbation rule with joint gradient normalization and layerwise equivalence classes can be achieved by introducing even more hyperparameters and decoupling numerator and denominator scalings of each layer. For (SAM) with perturbations (DP) with layerwise perturbation radii $\rho \cdot n^{-d_l}$ and separate gradient norm scaling $n^{-\tilde{d}_l}$, global perturbation scaling d is superfluous, and there exist layerwise equivalence classes: For any $\{\theta_l\}_{l\in[L+1]} \subset \mathbb{R}$,

$$(a_l, b_l, c_l, d_l, \tilde{d}_l) \text{ is equivalent to } (a_l + \theta_l, b_l - \theta_l, c_l - 2\theta_l, d_l - 2\theta_l, \tilde{d}_l - \theta_l).$$

This perturbation rule also allows us to recover an analytical equivalence between trivial weight multipliers $a_l = 0$ for all l, and any other weight multipliers. ◀

F.6.2 μP^2 under non-trivial weight multipliers.

Our goal here is to find the weight multipliers that simplify the necessary perturbation scaling for effective perturbations in all layers as much as possible. The non-existence of layerwise equivalence classes in $abcd$-parameterizations from (LP) is not an issue if we are interested in effective perturbation properties and recovering μP^2 for arbitrary weight multipliers $\{a_l\}_{l\in[L+1]}$, as the equivalence breaks due to varying gradient norm contributions, which are inconsequential for achieving effective perturbations.

As we aim to reproduce μP^2, we restrict ourselves to the μP equivalence class of abc-parameterizations. We do not allow layerwise perturbation scaling and are interested in the maximal stable choice of global perturbation scaling ρn^{-d} to at least achieve non-vanishing perturbations in some layers. The following lemma shows even more: **The choice**

$$a_l = -1/2 \cdot \mathbb{I}(l = 1) + 1/2 \cdot \mathbb{I}(l = L + 1)$$

achieves effective perturbations in all layers with the naive (SAM) update rule with naive perturbation scaling $\rho \cdot n^0$, and all layers contribute non-vanishingly to the joint gradient norm. Hence this seems to be a natural choice of weight multipliers for SAM. However, it is in conflict with unit scaling considerations (Blake et al., 2024). Effectively, naive learning rate and perturbation scaling with these multipliers is equivalent to (DP) where all denominator terms are scaled to be width independent, as implemented by Algorithm 1, which resembles our implementation for ViTs. Our ablations in Appendix H.4 suggest that gradient norm contributions have a negligible effect on generalization performance.

Lemma F.11 (Naive perturbation scaling can effectively perturb all layers). *Consider an $abcd$-parameterization where $\{(a_l, b_l, c_l)\}_{l\in[L+1]}$ are chosen from the μP equivalence class, and where there is some $C \in \mathbb{R}$ such that $d_l = C$ for all $l \in [L + 1]$. This reduces to training a MLP with weight multipliers with (SAM) with global perturbation scaling ρn^{-d} for some $d \in \mathbb{R}$. Effective perturbations in all layers are achieved and all layers contribute non-vanishingly to the gradient norm if and only if*

$$a_1 = -d - 1/2, \qquad a_l = -d \quad \text{for } l \in [2, L], \qquad a_{L+1} = -d + 1/2.$$

Achieving μP^2 with the current implementation of the `mup`-package requires both an adaptation of the architecture and of the learning algorithm, as the following lemma shows. Hence the package is not particularly suited for SAM learning in μP^2 when the goal is simple perturbation scaling.

Lemma F.12 (Effective perturbations with the `mup`-package). *Consider an $abcd$-parameterization where $\{(a_l, b_l, c_l)\}_{l\in[L+1]}$ are chosen from the μP equivalence class, and with the weight multipliers $a_{L+1} = \mathbb{I}(l = L + 1)$ as in the `mup`-package.*

- *(a) **(`mup`-package global scaling effectively perturbs hidden layers)** Under global scaling $d_l = C$, $C \in \mathbb{R}$, for all $l \in [L + 1]$, maximal stable perturbations are achieved with $d = 0$. In this parameterization, hidden layers are effectively perturbed, but input and output layers are not effectively perturbed.*
- *(b) **(μP^2 with the `mup`-package)** Effective perturbations in all layers are achieved with the choice $d = d_1 = d_{L+1} = -1/2$ and $d_l = 1/2$ for $l \in [2, L]$.*

The following lemma covers the general case how to achieve μP^2 given arbitrary weight multipliers.

Lemma F.13 (μP^2 with arbitrary weight multipliers). *Consider an $abcd$-parameterization where $\{(a_l, b_l, c_l)\}_{l\in[L+1]}$ are chosen from the μP equivalence class. Then effective perturbations in all layers are achieved with the choice $d = \min_{l\in[L+1]}(-a_l - 1/2\mathbb{I}(l = 1) + 1/2\mathbb{I}(l = L + 1))$, and*

$$d_1 = -1 - d - 2a_1, \qquad d_l = -d - 2a_l, \text{ for } l \in [2, L], \qquad d_{L+1} = 1 - d - 2a_{L+1}.$$

The following lemma shows that weight multipliers that achieve μP^2 with naive perturbation scaling under perturbations with layerwise normalization (LN) are exactly the same as the ones for (LP).

Lemma F.14 ((LN) with naive perturbation scaling can effectively perturb all layers). *Consider (SAM) with layerwise normalization (LN). Assume $\{(a_l, b_l, c_l)\}_{l\in[L+1]}$ are chosen from the μP equivalence class, and assume there is some $C \in \mathbb{R}$ such that $d_l = C$ for all $l \in [L + 1]$. Then all layers are effectively perturbed if the multipliers are chosen as*

$$a_1 = -1/2 - C, \qquad a_l = -C, \qquad a_{L+1} = 1/2 - C.$$

Proof of Lemma F.14. As derived in Appendix F.7, under $a_l = 0$ for all $l \in [L + 1]$, all layers are effectively perturbed if and only if $d_l = -1/2 \cdot \mathbb{I}(l = 1) + 1/2 \cdot \mathbb{I}(l = L + 1)$. Now we can exploit the layerwise equivalence relation to enforce $d_l = C$ in each layer by adapting all a_l. □

Proof of Lemma F.11. In general, in the abc-equivalence class of μP, the l-th layer's gradient norm is scaled by n^{-a_l}. This induces the generalized gradient norm constraints for $\|\nabla_W L\| = \Theta(1)$,

$$d_1 + a_1 \geq -1/2, \qquad d_l + a_l \geq 0, \qquad d_{L+1} + a_{L+1} \geq 1/2.$$

Effective perturbations are achieved when $\rho n^{-d-d_l-a_l}\nabla_{W^l}L = \Theta(n^{-\mathbb{I}(l>1)})$, which induces the perturbation stability constraints

$$d + d_1 + 2a_1 \geq -1, \qquad d + d_l + 2a_l \geq 0, \qquad d + d_{L+1} + 2a_{L+1} \geq 1,$$

with effective perturbations whenever the equality of the respective layer holds.

Under global scaling, the gradient norm constraints become, for some $C \in \mathbb{R}$,

$$C + a_1 \geq -1/2, \qquad C + a_l \geq 0, \qquad C + a_{L+1} \geq 1/2,$$

and the conditions for effective perturbations become

$$d + C + 2a_1 \geq -1, \qquad d + C + 2a_l \geq 0, \qquad d + C + 2a_{L+1} \geq 1.$$

As $d + C$ is a common term in all layers, we get the relations $a_l = a_1 + 1/2$, $a_{L+1} = a_1 + 1$, so that all gradient norm constraints are simultaneously satisfied with $C = -a_l$ and effective perturbations are achieved in all layers with $d = -a_l$. □

Proof of Lemma F.12. Under the choice $a_l = \mathbb{I}(l = L + 1)$, the gradient norm constraints become

$$d_1 \geq -1/2, \qquad d_l \geq 0, \qquad d_{L+1} \geq -1/2,$$

and the conditions for effective perturbations become

$$d + d_1 \geq -1, \qquad d + d_l \geq 0, \qquad d + d_{L+1} \geq -1.$$

Proof of (a):

Satisfying the gradient norm constraints with global scaling requires $d_l = 0$ for all $l \in [L + 1]$, then the minimal stable choice of d is $d = 0$ which only effectively perturbs hidden layers.

Proof of (b):

The choice $d = -1/2$ and $d_1 = -1/2$ saturates the gradient norm constraint and achieves effective perturbations in the input layer. Then the choice $d_l = 1/2$ and $d_{L+1} = -1/2$ satisfies the gradient norm constraints and achieves effective perturbations in all layers. □

Proof of Lemma F.7. To understand the influence of weight multipliers on updates and perturbations, first note that under an equivalence transformation of all $abcd$-parameters w.l.o.g from $a_l = 0$ for all $l \in [L + 1]$, the scalings of h^l, x^l and of $n^{-a_l}W^l$ remain invariant. This implies that the scalings of $\nabla_{x^L} f = n^{-a_{L+1}}W^{L+1}$, $\nabla_{h^l} f = \nabla_{x^l} f \odot \phi'(h^l)$ and $\nabla_{x^l} f$ for all $l \in [L]$ also remain invariant. Hence the weight gradients, $\nabla_{W^{L+1}} f = n^{-a_{L+1}}x^L$ and $\nabla_{W^l} f = n^{-a_l}\nabla_{h^l} f \cdot (x^{l-1})^\top$ are scaled by n^{-a_l} in each layer.

In the following forward pass, we get

$$h^l = n^{-a_l}(W^l + \Delta W^l)x^{l-1} = n^{-a_l}(W^l - \eta n^{-c_l}\nabla_{W^l}\mathcal{L})x^{l-1},$$

so that activation/output updates and perturbations of layer l are scaled by n^{-2a_l}.

Again, a complication compared to SGD or Adam arises through the gradient normalization of SAM's weight perturbation. If the gradients are simply normalized layerwise $\varepsilon^l = \rho \cdot n^{-d_l} \cdot \nabla_{W^l}\mathcal{L}/\|\nabla_{W^l}\mathcal{L}\|$, the n^{-a_l}-term from the backward pass cancels out, and only in the forward pass we get a scaling n^{-a_l}. Hence an exact layerwise equivalence still exists for SAM with layerwise gradient normalization:

$$(a_l, b_l, c_l, d_l) \text{ is equivalent to } (a_l + \theta_l, b_l - \theta_l, c_l - 2\theta_l, d_l - \theta_l).$$

Under joint gradient normalization (SAM), as we consider in our definition of bcd-parameterizations, keeping the gradient norm scalings invariant under $a_l \mapsto a_l + \theta$ would require $d_l \mapsto d_l - \theta$, but keeping the perturbation scaling invariant would require $d_l \mapsto d_l - 2\theta$ as for updates. Consequently, due to the layer coupling of joint gradient normalization $\|\nabla_{\mathbf{W}}\mathcal{L}\|$, an exact equivalence between $abcd$-parameterizations at finite width requires θ to be the same for all layers and the conflicting gradient norm in the denominator and perturbation scaling in the numerator to be accounted for by $d_l \mapsto d_l - \theta$ and $d \mapsto d - \theta$.

In the infinite-width limit, $abcd$-parameterizations remain equivalent under $(a_l + \theta_l, b_l - \theta_l, c_l - 2\theta_l, d_l - 2\theta_l, d)$ layerwise as long as the set of layers that contributes to the gradient norm non-vanishingly remains invariant. The gradient norm constraints for $\|v^l\| = O(1)$ become

$$d_1 + a_1 \geq 1/2 - c_\nabla, \qquad d_l + a_l \geq 1 - c_\nabla, \qquad d_{L+1} + a_{L+1} \geq 1/2,$$

where $\nabla_{x^L} f = n^{-a_{L+1}} W^{L+1} = \Theta(n^{-c_\nabla})$ with $c_\nabla = \min(b_{L+1} + a_{L+1}, c_{L+1} + 2a_{L+1})$ remains invariant under equivalence transformations. □

Proof of Lemma F.13. First, the choices of d_l ensure that the constraints for effective perturbations from the proof of Lemma F.11 are saturated in each layer. It is left to show, that these choices satisfy the $\|\nabla_W L\| = \Theta(1)$-constraints. For input layers, since $-d \geq a_1 + 1/2$, it holds that $d_1 + a_1 \geq -1/2$. For hidden layers, since $-d \geq a_l$, it holds that $d_l + a_l \geq 0$. For output layers, since $-d \geq a_{L+1} - 1/2$, it holds that $d_{L+1} + a_{L+1} \geq 1/2$. Observe that the minimizer in the definition of d saturates its gradient norm constraint. □

F.7 The spectral perspective on μP^2

While Tensor Programs allow to track the transformations of vectors like activations, Yang et al. (2023a) provide an equivalent formulation in terms of weight matrix spectral norms. They find that the spectral norm measures the effect of a weight update on the activations, under certain non-cancellation assumptions and limited batch size. For all MLP layers, they show that μP is equivalent to achieving the condition

$$\|W_t^l\|_* = \Theta\left(\sqrt{\frac{\texttt{fan\_out}}{\texttt{fan\_in}}}\right) \quad \text{and} \quad \|\Delta W_t^l\|_* = \Theta\left(\sqrt{\frac{\texttt{fan\_out}}{\texttt{fan\_in}}}\right),$$

at all times t, where $W_t^l : \mathbb{R}^{\texttt{fan\_in}} \to \mathbb{R}^{\texttt{fan\_out}}$. This condition is achieved with initialization σ_l, SGD learning rate η_l and Adam learning rate η_l^{Adam} chosen as,

$$\sigma_l = \Theta\left(\frac{1}{\sqrt{\texttt{fan\_in}}} \min\left\{1, \sqrt{\frac{\texttt{fan\_out}}{\texttt{fan\_in}}}\right\}\right), \quad \eta_l = \Theta\left(\frac{\texttt{fan\_out}}{\texttt{fan\_in}}\right), \quad \eta_l^{Adam} = \Theta\left(\frac{1}{\texttt{fan\_in}}\right).$$

This generalizes μP to varying widths inside the network. For varying widths, we adopt the notation $W^l : \mathbb{R}^{n_{l-1}} \to \mathbb{R}^{n_l}$ with $n_0 = d_{in}$ and $n_{L+1} = d_{out}$, whereas `fan_in` and `fan_out` always adapt to the weight matrix under consideration.

To understand why the spectral norm is desirable, note that $\Delta W^l = \eta_l \nabla_{h^l} \mathcal{L} (x^{l-1})^\top$ is low rank and aligned with the incoming activations. For batch size 1, we even have rank-1 updates with $\|\Delta W^l\|_* = \eta_l \|\nabla_{h^l} \mathcal{L}\|_2 \|x^{l-1}\|_2$, aligned with the incoming activations x^{l-1}, hence $\|\Delta W^l x^{l-1}\|_2 = \|\Delta W^l\|_* \|x^{l-1}\|_2$. This allows to achieve $\|\Delta x^l\|_2 = \Theta(\sqrt{n_l})$ irrespective of the layer type with $\|\Delta W_t^l\|_* = \Theta(\sqrt{n_l / n_{l-1}})$.

Our simple condition that perturbations should scale like updates, which is rigorously justified by our Tensor Program based proof in Appendix E, now allows to derive the correct perturbation scalings using the spectral weight perspective.

Layerwise perturbations. As a simple starting point, consider a variant of (SAM) that does not globally normalize the gradient of all layers jointly, but uses layerwise normalization (LN), resulting in the layerwise perturbation rule,

$$\boldsymbol{\varepsilon}^l = \rho_l \cdot \nabla_{W^l} \mathcal{L} / \|\nabla_{W^l} \mathcal{L}\|,$$

where $\|\cdot\|$ may denote either the spectral or the Frobenius norm (equivalent under limited perturbation batch size). Without the global normalization, the scalings of all layers are not coupled, and the spectral condition $\|\boldsymbol{\varepsilon}^l\|_* = \Theta(\sqrt{\texttt{fan\_out}/\texttt{fan\_in}})$ immediately requires choosing

$$\rho_l = \rho \cdot \sqrt{\texttt{fan\_out}/\texttt{fan\_in}}$$

for effective perturbations in layer l with width-independent hyperparameter $\rho \geq 0$.

Perturbations with global gradient normalization. Perturbations that are globally normalized across layers have usually been implemented practice according to the GitHub repositories provided

by Foret et al. (2021); Samuel (2022); Kwon et al. (2021); Andriushchenko and Flammarion (2022); Müller et al. (2024). Since we are interested in analysing (SAM) as it is applied in practice, we study variants with joint gradient normalization in more detail. Preliminary ablations in Appendix H.5 suggest that layer-coupled SAM with global normalization slightly outperforms SAM with layerwise gradient normalization. To simplify the analysis as much as possible, we will first ensure width-independence of the normalization, so that the layerwise perturbation scaling is not affected by the normalization term. Then, layerwise perturbations should again be scaled like updates.

Separate denominator scalings. If we allow to scale each denominator term separately from the corresponding numerator term (DP), the perturbation radius in each layer for the numerator can be scaled like updates, $\rho_l = \Theta\left(\frac{\texttt{fan\_out}}{\texttt{fan\_in}}\right)$.

Now, to ensure $\Theta(1)$ in the denominator, each input and hidden-like gradient norm $\|\nabla_{W^l}\mathcal{L}\|_F$, $l \in [L]$, achieves width-independence if it scaled by $\sqrt{\texttt{fan\_out}/\texttt{fan\_in}}$. The same rule applies to biases when understanding them as weights $\mathbb{R} \to \mathbb{R}^{n_l}$ to the input 1. These scalings are derived in the next paragraph. The last-layer gradient norm $\|\nabla_{W^{L+1}}\mathcal{L}\|_F$ should be scaled as $(n_L n_{L+1})^{-1/2}$, and $\|\nabla_{b^{L+1}}\mathcal{L}\|_F$ as $n_{L+1}^{-1/2}$.

If we care about the correct width-independent constants, observe that the learning rate scaling $\eta_{L+1} = \Theta(\texttt{fan\_out}/\texttt{fan\_in})$ induces $\|W^{L+1}\| = \|\Delta W^{L+1}\| = \Theta(\sqrt{n_{L+1}^3/n_L})$. If we wanted to achieve $\Delta W^{L+1} = \Theta(\sqrt{n_{L+1}/n_L})$ we would need $\eta_{L+1} = \Theta(1/\texttt{fan\_in})$. As $n_{L+1} = d_{out}$ is width-independent, $\sqrt{\texttt{fan\_out}/\texttt{fan\_in}}$ would result in the same width-dependent scaling for the last layer, but ignoring large constants can introduce a significant width-independent spectral distortion. For example in ImageNet1K, n_{L+1} is large. By tuning input, hidden and output multipliers such constant distortions may be corrected. The multiplier used in the `mup`-package does not correct this distortion. Using a base width at which SP is recovered may also cement such spectral distortions, if no multipliers are tuned.

Derivation of gradient norm $\|\nabla_{W^l}\mathcal{L}\|_F$ scalings. In this paragraph, $\|\cdot\|$ may denote the Frobenius or spectral norm. As all matrices are of limited rank, both norms scale equivalently. As a first step, $\|\nabla_{h^l}\mathcal{L}\| = \Theta(\frac{1}{\sqrt{n_l}})$ can be reconstructed from

$$\Theta(\sqrt{n_l/n_{l-1}}) = \|\Delta W^l\|_* = \eta_l \|\nabla_{h^l}\mathcal{L}\| \|x^{l-1}\|_2 = n_l/n_{l-1} \cdot \sqrt{n_{l-1}} \|\nabla_{h^l}\mathcal{L}\|.$$

Now, for input and hidden layers, $\|\nabla_{W^l}\mathcal{L}\| = \|\nabla_{h^l}\mathcal{L}\| \|x^{l-1}\|_2 = \Theta(\sqrt{n_{l-1}/n_l})$. Multiplying by the inverse yields width-independent scaling. The output layer gradient $\nabla_{W^{L+1}}\mathcal{L} \in \mathbb{R}^{n_{L+1} \times n_L}$ is given by $(\nabla_{W^{L+1}}\mathcal{L})_{ij} = x_j^L = \Theta(1)$, so that $\|\nabla_{W^{L+1}}\mathcal{L}\| = \Theta(\sqrt{n_L n_{L+1}})$. Biases before the last layer follow the scheme $\|\nabla_{b^l}\mathcal{L}\| = \|\nabla_{h^l}\mathcal{L}\| = \Theta(\sqrt{1/n_l}) = \Theta(\sqrt{\texttt{fan\_in}/\texttt{fan\_out}})$. The last layer bias $\|\nabla_{b^{L+1}}\mathcal{L}\| = \sqrt{n_{L+1}}$ scales width-independently as it should, but needs to be scaled by a different constant $1/\sqrt{\texttt{fan\_out}}$ than earlier layers.

Extensions to ASAM. As ASAM cannot be written as a NE⊗OR⊤program, its scaling can only be derived heuristically. As provided in Table 1 and derived in Appendix F.4, elementwise ASAM scales all layer types correctly in relation to each other, and it suffices to rescale the global perturbation radius by $\sqrt{n_L}$, assuming all width dimensions scale proportionally. For SAM-ON, we only perturb input-like layers such as normalization layers. As the conditions for correct scaling remain the same, the above scalings for input layers in SAM also apply to SAM-ON.

For layerwise ASAM, first note that $\|W_t^l\|_F = \Theta(\|W_0^l\|_F) = \Theta(\sqrt{n_l})$ for input and hidden layers $l \in [L]$. As the numerator contains $\|W_t^l\|_F^2$, it requires the layerwise perturbation scaling $\frac{1}{\texttt{fan\_in}}$. In the denominator, width independence is achieved with the multiplier $\sqrt{\frac{1}{\texttt{fan\_in}}}$, since $\|W^l\|_F \|\nabla_{W^l}\mathcal{L}\|_* = \sqrt{n_l}\sqrt{\frac{n_{l-1}}{n_l}} = \sqrt{n_{l-1}}$. Again, the output layer requires a special treatment. Due to its small initialization, it holds that $\|W^{L+1}\|_F^2 = \|\Delta W^{L+1}\|_F^2 = \Theta(\frac{n_{L+1}^3}{n_L})$. For perturbations that fulfill the spectral condition $\rho_{L+1}\|W^{L+1}\|_F^2 \|\nabla_{W^{l+1}}\mathcal{L}\|_* = \Theta(\sqrt{\frac{n_{L+1}}{n_L}})$, we need to choose $\rho_{L+1} = \rho \cdot \frac{1}{n_{L+1}^3}$ (width-independent, but very small). The last-layer denominator term scales as $\|W^{L+1}\|_F \|\nabla_{W^{l+1}}\mathcal{L}\|_* = \Theta(\sqrt{\frac{n_{L+1}^3}{n_L}} \cdot \sqrt{n_L n_{L+1}}) = \Theta(n_{L+1}^2)$, which is width independent, but

can be a large constant, as for ImageNet1K. The output bias numerator exactly conforms with the correct scaling $\|\nabla_{b^{L+1}}\mathcal{L}\|_F^2 = n_{L+1} = \texttt{fan\_out}/\texttt{fan\_in} = \Theta(1)$.

Note that weight decay may break statements like $\|W_t^l\|_F = \Theta(\|W_0^l\|_F)$ over long training. Everett et al. (2024) have recently observed more generally that scalings may evolve differently over long training than predicted by pure infinite-width TP theory, because alignments evolve dynamically between CLT- and LLN-like behaviour.

Using the `mup`-package. The `mup`-package introduces the output layer weight multiplier n_L^{-1} so that input and output layer learning rates may be scaled by the same width-dependent factor. Hence, only the last-layer scalings change. The scalings of $n_L^{-1}W^{L+1}$ and $n_L^{-1}\Delta W^{L+1}$ remain the unique ones that achieve μP, but $\nabla_{W^{L+1}}\mathcal{L}$ is scaled by n_L^{-1}. This requires adapting the last-layer learning rate η_{L+1} to scale like input layers. For SAM, the last-layer perturbation radius can now be scaled like input layers. That is, assuming proportionally growing width n, in the numerator $\rho_{L+1} = \rho_1 = \rho \cdot n$ and $\rho_l = \rho$ for $l \in [2, L]$, and the gradient norm contributions should be scaled by $\sqrt{n}$ for input and output layers, and by 1 for hidden layers. The Tensor Program perspective on weight multipliers can be found in Appendix F.6. The correct width-independent constants are achieved with the last-layer numerator scaling $\rho_{L+1} = \rho \cdot n_L$ and the last-layer denominator scaling $\sqrt{n_L/n_{L+1}}$, since $\nabla_{W^{L+1}}\mathcal{L} = \Theta(\sqrt{n_{L+1}/n_L})$ and for the numerator we get an additional n_L^{-1} in the forward pass.

For SAM-ON nothing changes, as only input-like layers are perturbed. For elementwise ASAM, ignoring width-independent constants, nothing changes as the weight multiplier n_L^{-1} increases the weight scaling W^{L+1} and decreases the gradient scaling $\nabla_{W^{L+1}}\mathcal{L}$ by the same amount. The additional n_L^{-1}-factor in the numerator is cancelled out by the additional W^{L+1}-factor. For the correct width-independent constants with decoupled numerator and denominator scaling, we would scale the denominator by $\sqrt{n_L/n_{L+1}^3}$ with or without weight multiplier, and scale the numerator by $\rho_{L+1} = \rho \cdot n_L/n_{L+1}^2$ with or without weight multiplier. For the example of layerwise ASAM, we still get for the denominator $\|W^{L+1}\|_F\|\nabla_{W^{l+1}}\mathcal{L}\|_* = \Theta(n_{L+1}^2)$, again because the weights W^{L+1} are scaled up by n_L and the gradient is scaled down by the same amount. In the numerator, the upscaling of the weights also cancels out the downscaling of the gradient and additional n_L^{-1} in the subsequent forward pass, leading to an unchanged $\rho_{L+1} = \rho \cdot n_{L+1}^{-3}$, which is width-independent but potentially leads to numerical issues.

Code for μP^2 with separate denominator scalings. Algorithm 1 provides a PyTorch code example that implements the above μP^2 scalings for SAM, scaling the gradient norm contributions of all layers to $\Theta(1)$ (equivalent to (a-μP^2) together with naive perturbation and learning rate scaling). We adapt the popular SAM implementation Samuel (2022) using the `mup`-package. This code resembles our implementation for the ViT experiments. In the `mup`-package, 'vector-like´ parameters scale as $n \times$ constant or constant $\times\, n$ and include input and output weights. The last-layer multiplier n_L^{-1} is chosen so that input and output layers can be scaled by the same width-dependent factor. On the other hand, 'matrix-like´ parameters scale as $n \times n$ and include hidden weights. The implementation uses a base width at which μP^2 and SP are equivalent; all width-dependent scalings then scale with width-multipliers `width`/`base_width`. This allows to immediately transfer well-performing settings from SP to μP^2.

Let us recapitulate how the μP^2 scaling in the following code arises. The crucial variables to track are `factor`, `group["rho"]` and `group["gradnorm_scaling"]`. For limited batch size, the spectral and Frobenius norm of gradients scale equivalently, and we get, for all $l \in [L]$,

$$\|\nabla_{W^l}\mathcal{L}\|_F = \Theta(\|\nabla_{W^l}\mathcal{L}\|_*) = \Theta\left(\sqrt{\frac{\texttt{fan\_in}}{\texttt{fan\_out}}}\right).$$

We want to scale each weight's contribution in the denominator to be width-independent, hence need the factor $\sqrt{\texttt{factor}}$ with $\texttt{factor} = \texttt{fan\_out}/\texttt{fan\_in}$. For the numerator, the spectral condition ($*$) demands $\|\rho_l \cdot \nabla_{W^l}\mathcal{L}\|_* \overset{!}{=} \Theta(\sqrt{\frac{\texttt{fan\_out}}{\texttt{fan\_in}}})$, so that we need to scale the weight's perturbation radius to $\rho_l = \rho \cdot \texttt{factor}$. Since the `mup`-package sets the last-layer weight multiplier such that input and output layers can be scaled in the same way, the implementation is short. For optimal numerical properties however, this choice of multipliers is sub-optimal (Blake et al., 2024).

```
import math, torch
from mup import MuAdamW

# specify parameterization
parameterization = 'mupp' # 'sp-naive', 'mup-naive'
# for 'mup-global' use 'mup-naive' and scale rho accordingly

# specify model and hyperparameters
model, lr, rho, weight_decay, last_layer_weight_name = ...



# adapt SAM to allow gradient norm scaling of each weight tensor
class SAM(torch.optim.Optimizer):
  ...

  def grad_norm(self):
    grads = []
    for i, group in enumerate(self.param_groups):
      for p in group["params"]:
        grads.append((group["gradnorm_scaling"] * p.grad).norm(p=2))
    norm = torch.stack(grads).norm(p=2)
    return norm

  @torch.no_grad()
  def first_step(self): # perturbation step before the weight update
    grad_norm = self.grad_norm()
    for group in self.param_groups:
      scale = group["rho"] / (grad_norm + 1e-12)
      for p in group["params"]:
        if p.grad is None: continue
        self.state[p]["old_p"] = p.data.clone()
        e_w = p.grad * scale.to(p)

        p.add_(e_w)  # climb to the local maximum "w + e(w)"



# set width-dependent rho and gradient norm scaling for each weight
param_groups = []
for name, p in model.named_parameters():
  if p.infshape.ninf() == 0 or 'naive' in parameterization:
    factor = 1
  elif p.infshape.ninf() == 1:
    # vector-like
    for d in p.infshape:
      if d.base_dim is not None:
        factor = d.dim / d.base_dim #width
        break
  elif p.infshape.ninf() == 2:
    # matrix-like
    factor = (p.infshape[0].dim/p.infshape[1].dim) * (p.infshape[1].base_dim/
   p.infshape[0].base_dim) # fan_out/fan_in
  else:
    raise NotImplementedError

  group = {
    "params": [p],
    "lr": lr,
    "rho": rho * factor,
    "gradnorm_scaling": math.sqrt(factor),
  }
  param_groups.append(group)


```

```
optimizer = SAM(param_groups,
  base_optimizer=MuAdamW if parameterization=='mup' else torch.optim.
    AdamW, weight_decay=weight_decay)
```

Algorithm 1: Pytorch implementation of μP^2 for SAM using the `mup`-package. Key changes from the original implementation that correct the layerwise perturbation scaling are highlighted with gray boxes. This code decouples the scalings of numerator and denominator terms following (DP), and scales the gradient norm contributions of all layers by `group["gradnorm_scaling"]` in the denominator to be width-independent. The numerator terms `group["rho"]` of all weight tensors are scaled to achieve effective perturbations. This scaling is equivalent to (a-μP^2) together with naive perturbation and learning rate scaling.

G Experimental details

If not mentioned otherwise, experiments use the settings specified in this section.

Implementation details. For MLPs, we exactly implement our Definition 4 of bcd-parameterizations to precisely validate our theoretical results. For ResNets and ViTs, the width varies inside the network, so that we implement the spectral scaling rules derived in Appendix F.7. Like the `mup`-package, we introduce a base width at which SP and μP are equivalent, allowing to immediately transfer setups that perform well in SP. We use the `mup`-package only for ViTs, and our implementation of μP^2 resembles the pseudocode provided in Algorithm 1. For ResNets, we use no width-dependent last-layer multiplier. At initialization, μP differs from SP only through a smaller last layer initialization. For MLPs we exactly implement the bcd-parameterization with $b_{L+1} = 1$, but use the large width-independent input layer initialization variance 2 instead of the width-independent $2/d_{in}$ in μP, which can be seen as a tuned initialization variance multiplier. For ResNets and Vits, we initialize the last layer to 0 in μP, which corresponds to $b_{L+1} \to \infty$ and which recovers the limit behaviour $f_0 \to 0$ already at finite width. We are working on making Python code to reproduce all of our experiments publicly available.

MLPs. We train 3-layer MLPs without biases with ReLU activation function for 20 epochs with constant learning rate, using SGD as base optimizer as specified in Definition 4, but allow for SGD batchsize larger than 1, defaulting to batch size 64. We evaluate the test accuracy after every epoch and use the snapshot across training with the best accuracy. This is necessary as the test accuracy is not monotonically increasing across training, while the training accuracy is. For ResNets we do not observe such harmful overfitting. For the standard parametrization, we use He initialization (He et al., 2015) and don't tune multipliers to mimic standard training procedures. For μP, we resort to the optimal multipliers from Yang et al. (2022). We then find the optimal learning rate and perturbation radius for each bcd-parametrization and SAM variant separately.

ResNets. For ResNet18 experiments, we augment the CIFAR10 data with random crops and random horizontal flips, set labelsmoothing to 0.1 and use a cosine learning rate schedule. ResNets in μP have base width 0.5, gradient norm scaling according to Definition 4 and their last layer is initialized to 0. For SP, we again adopt the standard hyperparameters from Müller et al. (2024) by using a momentum of 0.9, weight decay 0.0005, an output multiplier of 1.0, and individually tuned learning rate and perturbation radius for each SAM variant. For μP, at base width multiplier 0.5 compared to the original width, for each SAM variant, we perform a random grid search over the hyperparameters learning rate, perturbation radius, output multiplier $[2^{-8}, 2^{-7}, \dots, 2^{8}]$, weight decay $[0, 10^{-5}, 10^{-4}, 5 \cdot 10^{-4}, 10^{-3}, 10^{-2}]$ and momentum $[0, 0.1, 0.4, 0.7, 0.9]$. Learning rate and perturbation radius grids were either set to $[2^{-10}, 2^{-9}, \dots, 2^{1}]$ or centered around recommendations from the literature. The optimal hyperparameter configurations found from at least 150 runs for each SAM variant are summarized in Table G.1. Learning rates and perturbation radii were further tuned with the experiments from Appendix H.3.3.

ViTs. We train ViT-S/16 with 6 layers and 12 attention heads on ImageNet1K (Deng et al., 2009) and a ViT-S/4 with 12 layers and 12 attention heads on CIFAR100 (Krizhevsky et al., 2009) (see Appendix H.6), again adopting the hyperparameter settings from Müller et al. (2024). This means we use AdamW as a base optimizer with warmup and a cosine learning rate decay. For CIFAR100, we use random crops, random horizontal flips and AutoAugment as data augmentations. For Imagenet we use the original preprocessing from Huggingface `vit-base-patch16-224` (Wu et al., 2020). For μP, we tune multipliers at a basewidth 384, initialize the last layer and query weights to 0. By

using the μP package, the relative perturbation scalings change as explained in Appendix F.7 and Appendix F.6. Global and naive perturbation scaling in μP now coincide. Here, instead of the original perturbation scaling Definition 4, we scale the gradient norm contributions of all layers in the denominator to $\Theta(1)$. The hyperparameter choices for ViTs on CIFAR100 and ImageNet are summarized in Table G.2. For μP, the learning rate, perturbation radius, input multiplier, output multiplier and weight decay were tuned using 3 independent runs of Nevergrad `NGOpt` with budget 56 on ImageNet. The same multipliers are used on CIFAR100.

Figures. Whenever multiple runs with independent random seeds are used for training, confidence bands cover the interval from the empirical 2.5%- to the empirical 97.5%-quantile. The line then denotes the average of all runs. When confidence bands are given, but the number of independent runs is not specified, the number of runs defaults to 4.

Computational resources. We ran all of our experiments on Amazon EC2 G5 instances each containing up to 8 NVIDIA A10G GPUs. On a single GPU, our μP^2-SAM training script for MLPs of width 4096 on CIFAR10 takes 502 seconds to run in total (25 seconds per epoch), where data handling takes most of the time. The training times for ResNets and ViTs are presented in Table G.3.

Hyperparam.	SAM		SAM-ON		ResNet18	Elem. ASAM		Layer ASAM	
	SP	μP^2	SP	μP^2	SGD	SP	μP^2	SP	μP^2
Training epochs					200				
Batch size					64				
LR η	0.05	2^{-4}	0.05	2^{-4}		0.05	2^{-4}	0.1	2^{-4}
LR decay					Cosine				
Weight decay					0.0005				
Momentum					0.9				
Labelsmoothing					0.1				
Pert. radius ρ	0.1	2^{-4}	0.5	$5 \cdot 2^{-4}$		2	$10 \cdot 2^{-4}$	0.02	2^{-6}
Output multiplier	1	0.125	1	0.125		1	0.125	1	0.125

Table G.1: **(ResNet-18 hyperparameters for CIFAR10)** Hyperparameters for SP are taken from Müller et al. (2024). Learning rate and perturbation radius are tuned using the experiments in Appendix H.3.3. ResNets in μP have base width 0.5, gradient norm scaling according to Definition 4 and their last layer is initialized to 0.

Hyperparam.	SAM on ImageNet1K			SAM on CIFAR100	
	SP	μP^2	shared	SP	μP^2
Training epochs	100			300	
Batch size			128		
LR η	0.001	0.00226		0.0005	
LR warmup epochs	10			30	
LR decay			Cosine		
Weight decay	0.1	0.0872		0.05	
Labelsmoothing			0.1		
Pert. radius ρ	1	1.1939		0.25	0.25
Input multiplier	1	1.7309		1	1.7309
Output multiplier	1	4.0946		1	4.0946
Layers	6			12	
Attention heads			12		
Patch size	16			4	

Table G.2: **(Vision Transformer hyperparameters)** Hyperparameters for SP are taken from Müller et al. (2024) using AdamW as a base optimizer. ViTs in μP have base width 384, last layer and query weights are initialized to 0 and gradient norm contributions of all layers are scaled to $\Theta(1)$.

H Supplemental experiments

This section provides more extensive empirical evaluations to validate the claims of the main paper. By naive perturbation scaling (naive) we denote parameterizations that do not adapt any perturbation scalings ($d = d_l = 0$ for all l). Global perturbation scaling (global) denotes the maximal stable scaling n^{-d} of the global perturbation radius that achieves effective perturbations in some layers without layerwise perturbation scaling ($d_l = 0$ for all l).

H.1 SAM is approximately LL-SAM in μP with global perturbation scaling

Figure H.1 compares SAM in μP under global perturbation scaling (μP-global) with SAM under global perturbation scaling where only the last-layer weights are perturbed (LL-SAM) by showing more neural network statistics that are related to SAM's inductive bias and to learning in general. From top-left to bottom right, the statistics are: Frobenius norm of the layerwise weight perturbation (which is closely related to spectral norm as perturbations are low rank); Frobenius norm of the layerwise weight perturbation normalized by the weight spectral norm to upper bound the influence of the perturbations on the output; spectral norm of the weight updates across training scaled by the spectral condition $n^{1/2}$, 1 and $n^{-1/2}$ for input, hidden and output layers respectively; norm of the activation updates for each layer normalized by the square root of the layer's output dimension to measure coordinatewise update scaling; layerwise effective feature ranks measured as in Andriushchenko et al. (2023a) by the minimal amount of singular values to make up 99% of the variance of the activations in a given layer; gradient norm, Hessian spectral norm and Hessian trace of loss with respect to weights; training accuracy, test accuracy after optimally stopping.

Observe that, especially for large widths, global perturbation scaling effectively only perturbs the last layer, as predicted by Theorem 11. Last-layer SAM is more similar to μP-global SAM than SGD on all of the tracked statistics, in particular at large widths. Only perturbing the last layer still affects the gradients in earlier layers so that weight updates and activations change in all layers. We find that SAM in μP with global scaling does not consistently improve generalization performance over SGD, whereas μP^2 does improve over SGD for all widths (Figure H.3). Last-layer perturbation norms coincide by design with the global perturbation radius $n^{-d}\rho$ and their effect on the activations stays $\Theta(1)$ with increasing width as measured in relation to weight spectral norm. Formally the last-layer perturbation norm converges due to

$$\|\tilde{W}^{L+1} - W^{L+1}\|_F = n^{-d}\rho \|\frac{\chi_t x_t^L}{\|v_t\|}\|_F \to n^{-d}\rho \|\frac{x_t^L}{\|x_t^L\|}\|_F = n^{-d}\rho \to 0,$$

where the loss derivative χ_t always cancels out due to the normalization and the global gradient norm $\|v_t\|$ is dominated by the last-layer gradient norm due to the global scaling (Theorem 11). Normalizing the weight perturbations by the weight spectral norm measures the influence of the perturbations on the activations. Note that this influence is also vanishing. Feature ranks stay close to initialization, since random initialization has high rank and training does low effective rank updates. Here we do not observe that SAM reduces the feature rank compared to SGD. The Hessian spectral norm and trace are quite noisy. The last-layer Hessian spectral norm explodes with width in μP, because last-layer learning rate is scaled as n^{-1}, hence the edge of stability explodes. ResNets in μP are more stable, their Hessian spectral norm even shrinks with width (not shown).

Contrast the results for μP-global with the results for μP^2 in Figure H.2 for a comparison with SGD in μP. The Hessian spectral norm is reduced by SAM as you would expect. Additionally μP^2 shows low variability in performance and all other statistics. SAM in μP^2 does not reduce the feature rank compared to SGD in μP. This suggests that the conclusions drawn by Andriushchenko et al. (2023a) do not apply to MLPs in μP.

	ResNet-18 on CIFAR10				ViT on CIFAR100			ViT on ImageNet1K		
Width multiplier	0.5	1	2	4	0.5	1	2	0.5	1	2
Seconds per epoch	109	161	327	803	209	327	777	2550	4151	9802

Table G.3: **(Training time per epoch)** Training time (in seconds) per epoch of the entire data loading and training pipeline of SAM in μP^2 on a single NVIDIA A10G GPU.

H.2 Propagating perturbations from the first layer does not inherit SAM's benefits

Here we apply a parametrization that only effectively perturbs the first layer weights (derived in Example F.1). Figure H.2 shows that effective first-layer SAM loses both μP^2 SAM's improvement in test accuracy as well as SAM's inductive bias towards smaller gradient norm and Hessian norm, i.e. lower sharpness in MLPs. This performance deterioration occurs although the perturbation of first-layer SAM has an effect of the same order of magnitude as μP^2 on weight and activation updates in all layers. This shows that mere propagation of weight perturbations from earlier layers cannot replace effective weight perturbations in each layer in order to benefit from SAM. It is crucial to correctly adjust the layerwise perturbation scaling, and to distinguish between effective perturbations and perturbation nontriviality in each layer.

SAM in μP^2, on the other hand, achieves the correct perturbation and update scaling, has lower final gradient and Hessian spectral norm, improves test accuracy over SGD and has overall lower variance between training runs.

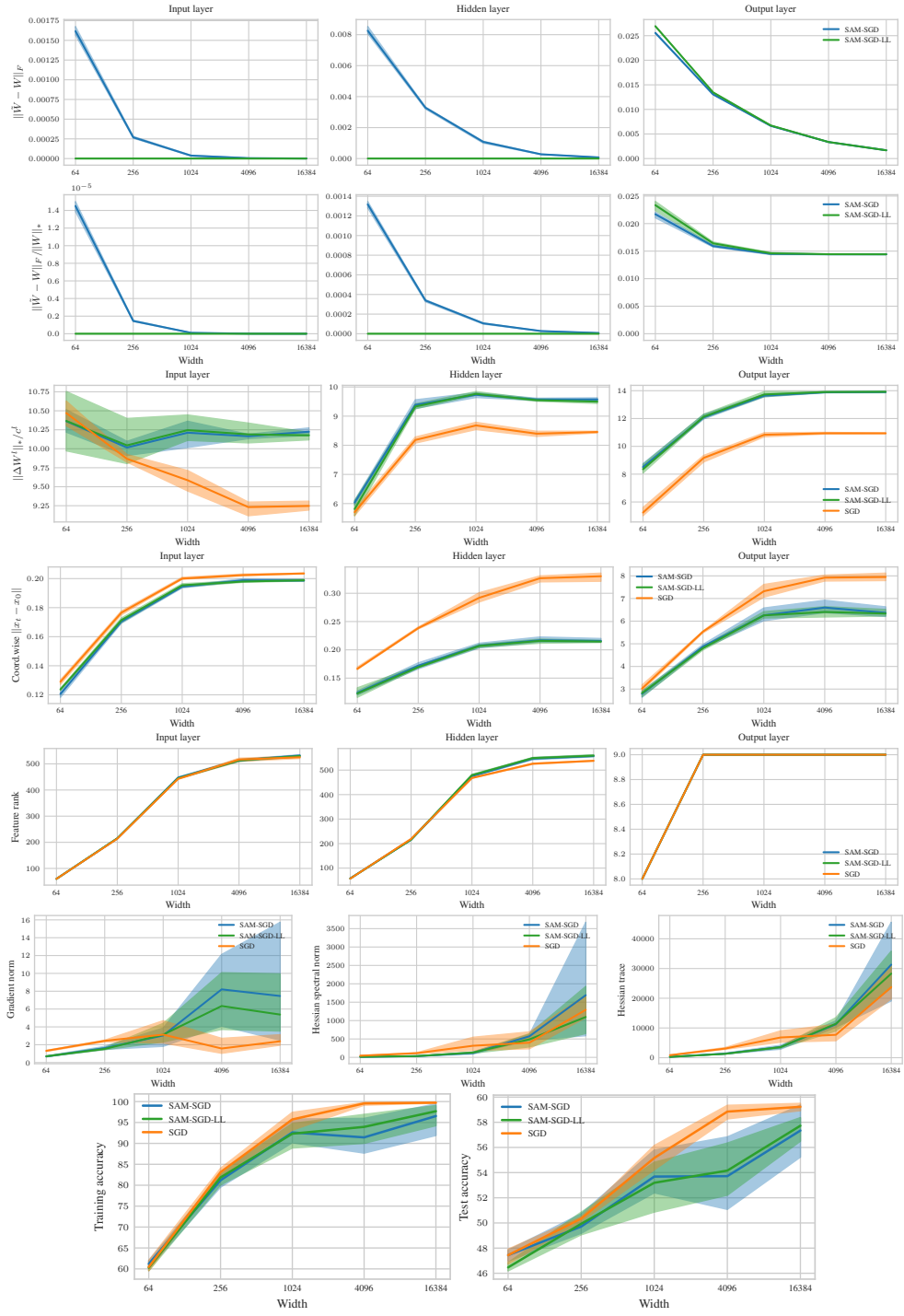


Figure H.1: Several neural network statistics for SAM (blue), LL-SAM (green) and SGD as a baseline (orange) across width after training a 3-layer MLP in μP-global for 20 epochs with the optimal learning rate 0.3432 and perturbation radius 0.2154. The statistics are explained in the text of Appendix H.1.

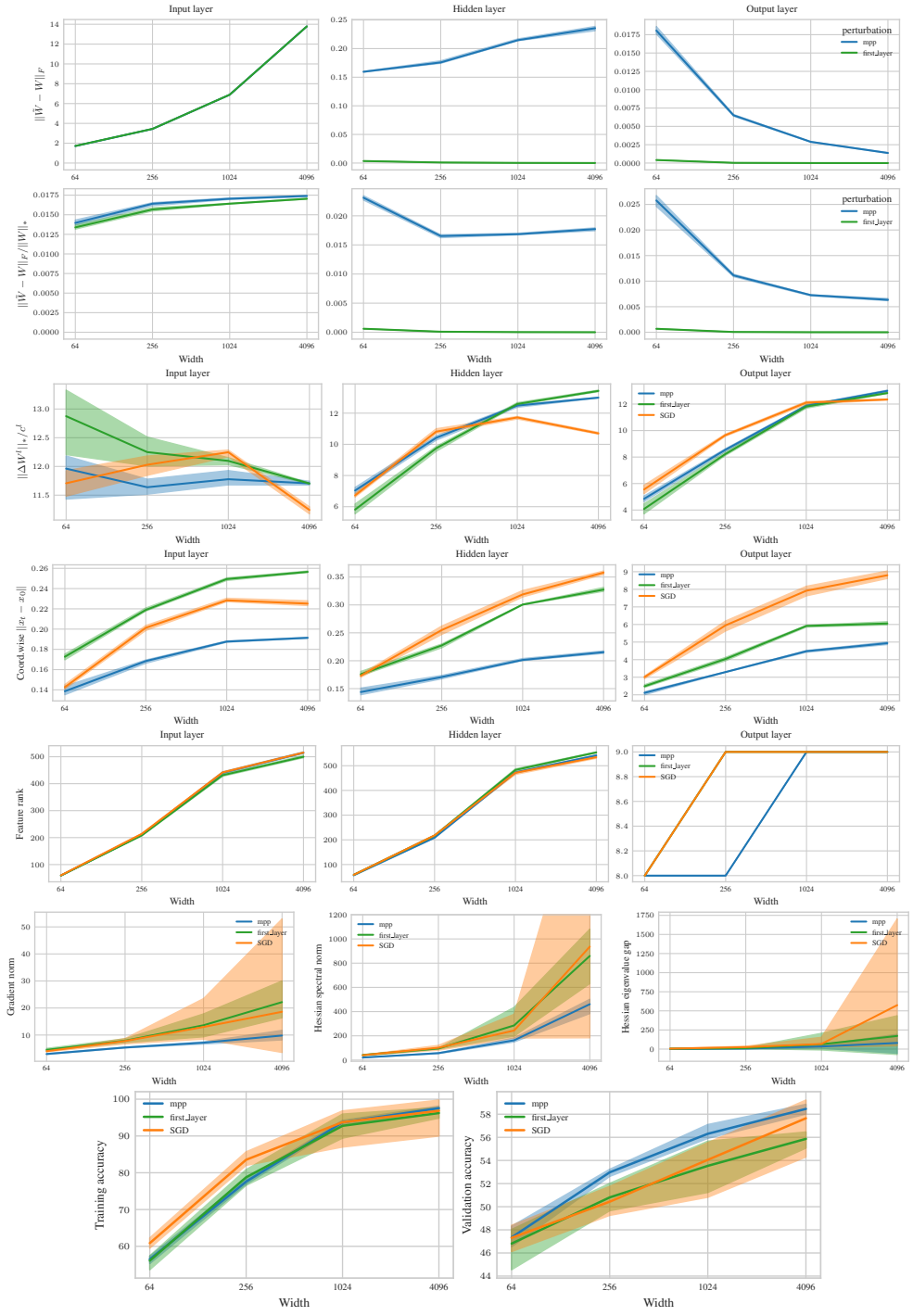


Figure H.2: Same neural network statistics as in Figure H.1 but SAM-SGD in μP^2 (blue) versus MUP with perturbations scaled to only effectively perturb the first layer weights (green) with SGD in μP as a baseline. The first-layer perturbation parameterization performs worse than μP^2 and results in gradient norm and Hessian norm similar to that of SGD, larger than those of SAM. While the spectral norm of the weights converges to a similar quantity as for μP^2, the effect of the weight changes on the hidden activation updates behaves more like SGD. Feature ranks all look similar.

H.3 Hyperparameter transfer

In this section, we provide supplemental evidence that, μP^2 is the unique perturbation scaling that robustly achieves hyperparameter transfer in μP both for the optimal learning rate and the optimal perturbation radius across neural architectures and datasets. But we also show that both MLPs and ResNets in SP can sometimes achieve hyperparameter transfer after long training.

H.3.1 MLPs in μP

Figure H.3 shows that in μP^2 the optimal hyperparameters in terms of test accuracy transfer in both learning rate and perturbation radius at sufficient width, and test accuracy monotonically improves with model scale. In addition, SAM in μP^2 outperforms SAM in μP with global perturbation scaling at all widths.

While other works focus on hyperparameter transfer in training loss, we are ultimately interested in transfer with respect to test accuracy. Especially under harmful overfitting, the test accuracy is affected by nontrivial interactions between the learning rate and the perturbation radius. While the joint optimum is slightly shifting towards larger learning rate and perturbation radius for small widths, it remains remarkably stable for sufficient width ≥ 1024. Note that slight shifts in the optimal learning rate due to finite width biases have also been observed in earlier works (Yang et al., 2022).

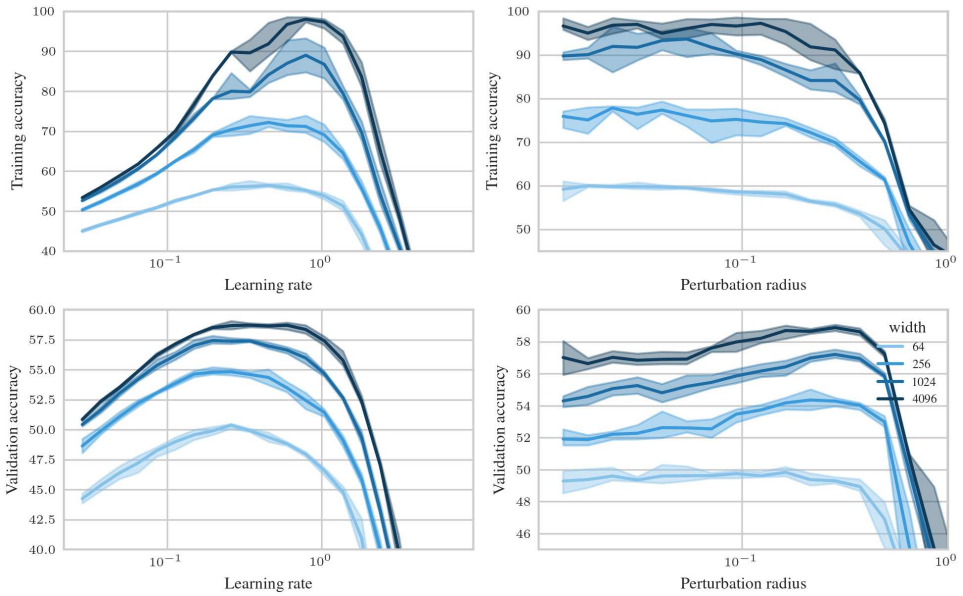


Figure H.3: Training accuracy (top) and test accuracy (bottom) after optimally stopping 20 epoch SAM training as a function of learning rate (left) with perturbation radius $\rho = 0.2154$, and as a function of perturbation radius (right) with learning rate $\eta = 0.4529$ in μP^2. The optimal learning rate transfers. The smaller the perturbation radius the better the training accuracy. For sufficiently wide MLPs, the validation-optimal perturbation radius transfers as well and SAM reduces harmful overfitting.

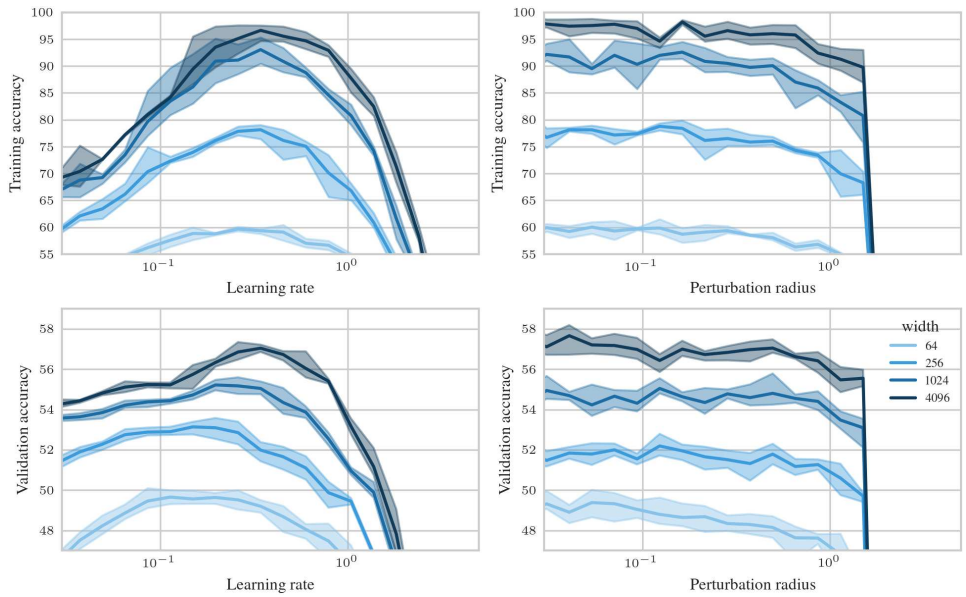


Figure H.4: Training accuracy (top) and test accuracy (bottom) after optimally stopping 20 epoch SAM training as a function of learning rate (left) and perturbation radius (right) in μP-global with the same base learning rate and perturbation radius as in Figure H.9. For global perturbation scaling, we do not observe a benefit of SAM over SGD.

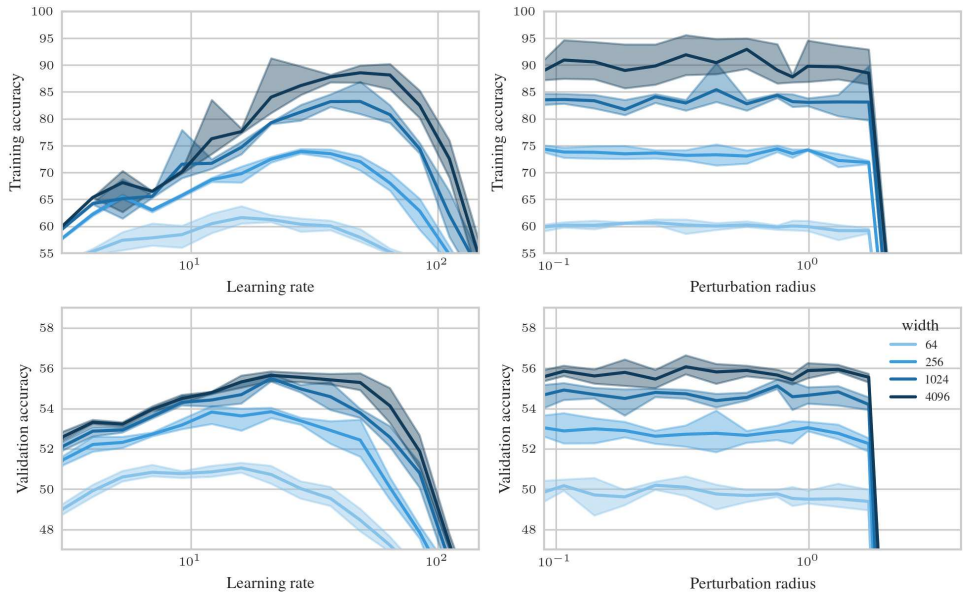


Figure H.5: Same as Figure H.4 but with input multiplier 0.0305 and small output multiplier 0.0098. Note that networks with width at most 256 perform better in terms of test accuracy than with the other multiplier choice in Figure H.4, but the multipliers here have worse width scaling properties. To the best of our knowledge, the issue that optimally tuned hyperparameters on small models may scale worse than slightly suboptimal hyperparameters has not been stated before. This raises the question when and how can we use small models to predict the optimal hyperparameters of large models.

Figure H.4 shows that global perturbation scaling does transfer the same perturbation instability threshold, whereas in μP-naive every fixed perturbation radius becomes unstable at sufficient width (Figure H.7). But in μP-global we do not observe a benefit of SAM over SGD. While the optimal learning rate with respect to the training accuracy transfers, the optimal learning rate with respect to the validation error is smaller for MLPs of moderate widths due to harmful overfitting. How to control for non-monotonic dependence of the test error on the training error is an important question for future work. Figure H.5 also shows μP-global but with a different choice of input and output multipliers. With these multipliers, networks with width at most 256 perform better in terms of test accuracy than with the other multiplier choice in Figure H.4, but these multipliers have worse width scaling properties. To the best of our knowledge, the issue that optimally tuned hyperparameters on small models may scale worse than slightly suboptimal hyperparameters has not been stated before. This raises the question when and how can we use small models to predict the optimal choice of all hyperparameters jointly in large models.

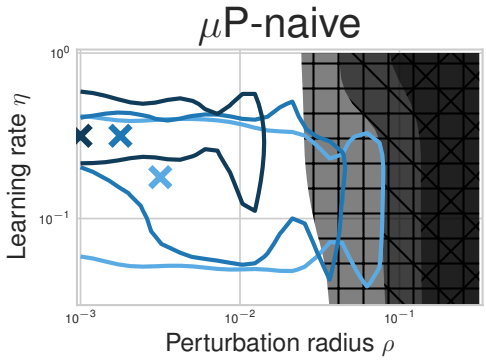


Figure H.6: Same as Figure 1 but for μP with naive width-independent perturbation scaling ρ. The regime of stable perturbation radii shrinks with increasing width as predicted by Proposition 1.

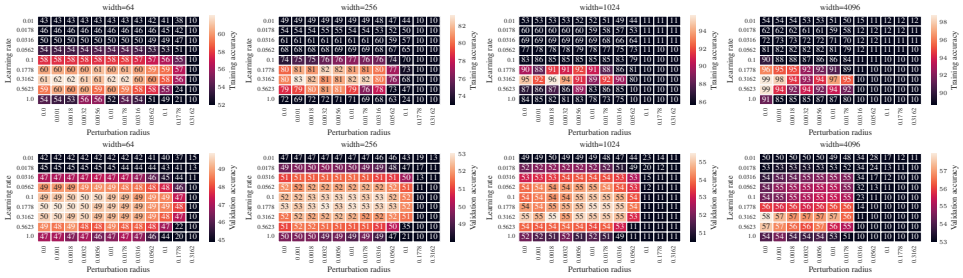


Figure H.7: Mean (over 3 runs) of training accuracy (top) and of test accuracy (bottom) after optimally stopping 20 epoch SAM training of a MLP in μP-naive as a function of learning rate and perturbation radius. The optimal hyperparameters do not transfer. Every fixed perturbation radius becomes unstable in sufficiently wide networks.

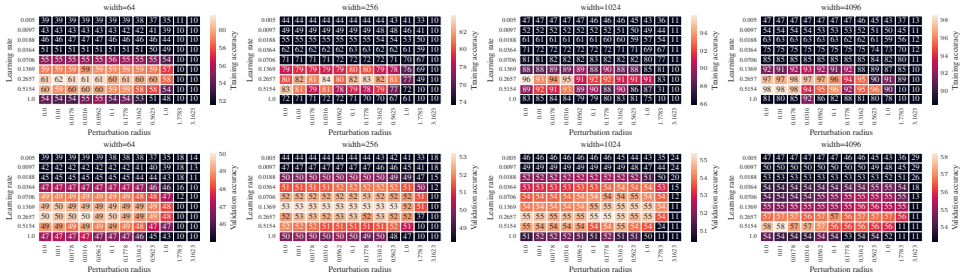


Figure H.8: Mean (over 3 runs) of training accuracy (top) and of test accuracy (bottom) after optimally stopping 20 epoch SAM training of a MLP in μP-global as a function of learning rate and perturbation radius. The global scaling of the perturbation radius by $n^{-1/2}$ compared to μP-naive (Figure H.7) makes the stable regime invariant to width. But the suboptimal layerwise perturbation scaling that only perturbs the last layer does not consistently improve over SGD ($\rho = 0$).

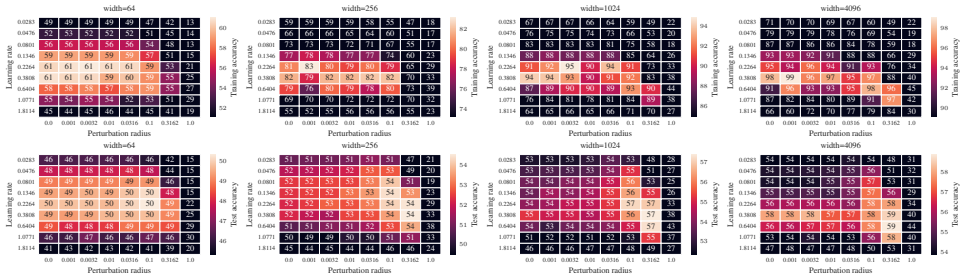


Figure H.9: Mean (over 3 runs) of training accuracy (top) and of test accuracy (bottom) after optimally stopping 20 epoch SAM training of a MLP in μP^2 as a function of learning rate and perturbation radius. At sufficient width, the optimal hyperparameters are stable in terms of test accuracy, even under severe overfitting.

H.3.2 Some variants of SP can transfer optimal hyperparameters on CIFAR-10

Surprisingly, after training MLPs to convergence on CIFAR-10, some variants of SP with naive perturbation scaling can transfer learning rate and perturbation radius, against the prediction by infinite-width theory. For SGD, this has originally been observed in GitHub issue 52 of the `mup`-package[4]. We train MLPs with SAM in variants of SP in Figure H.10 and also observe that some variants achieve transfer while for others the optimal learning rate shrinks as in Yang et al. (2022).

To achieve shrinking stable and optimal learning rates in SP, Yang et al. (2022) use weight multipliers tuned at base width 256 and normalize the initialization variance to be invariant to these weight multipliers, according to the Jupyter notebook[5] provided for reproducing their experiments. In addition, they initialize the last layer to 0 which contradicts SP scaling but results in more striking shrinkage of the optimal learning rate. We observe that both with and without weight multipliers, MLPs trained with SAM in SP-naive have surprisingly good transfer properties on CIFAR-10. With tuned multipliers but initialization that is invariant to these multipliers, the optimal learning rate shrinks. Because we are training to convergence, pure infinite-width theory does not adequately describe the training dynamics anymore (Vyas et al., 2024). Infinite-width theory implies that scaling the width further would eventually break the learning rate transfer. It remains a matter of ongoing work to understand whether this stability of SP is a finite-width or a long training time effect, and whether this empirical stability is particular to multi-epoch training on vision datasets. As shrinkage of the optimal learning rate in SP has generally been observed in language settings (see e.g. Brown et al., 2020, Table 2.1), we expect the same shrinkage for SAM in SP in such settings.

[4]Without tuned weight multipliers, MLPs trained with SGD in SP on CIFAR-10 can transfer the optimal learning rate: https://github.com/microsoft/mup/issues/52

[5]https://github.com/microsoft/mup/blob/main/examples/MLP/demo.ipynb

Note that the learning rate transfer in SP here is a much stronger observation than in Everett et al. (2024) who choose the correct layerwise learning rates for SP. Hence their SP merely deviates from μP through a larger output layer initialization and key-query normalization by $\sqrt{d}$ in SP versus by d in μP. Here however we even observe transfer in our stricter understanding of SP without any layerwise learning rates or weight multipliers. This is not a peculiarity of SAM; we observe the same learning rate transfer in plain SGD without any momentum or weight decay for MLPs on CIFAR-10 (not shown).

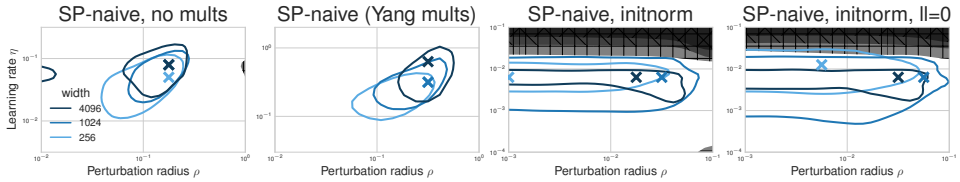
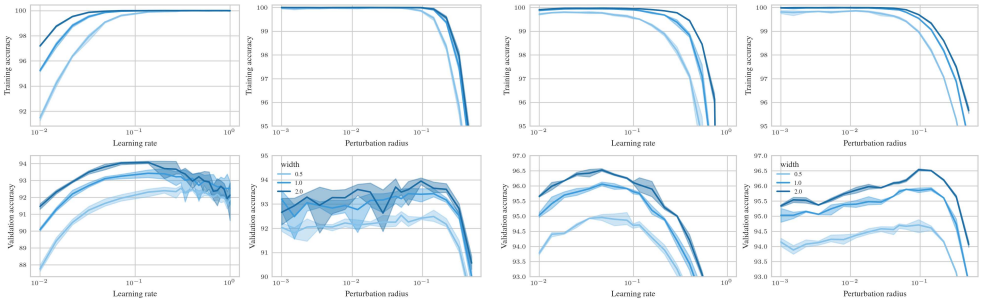


Figure H.10: Optimal learning rate and perturbation radius (cross) and regions within 1% of the optimal test accuracy (mean over 4 runs) after optimally stopping 20 epoch SAM training of a MLP in different variants of SP for varying widths (the darker, the wider). Observe transfer properties in SP almost as stable as in μP^2 (Figure 1) and against infinite-width predictions slightly growing with width, both with and without the tuned weight multipliers by Yang et al. (2022). Only when normalizing the initialization variance to be independent of the width-independent weight multipliers (initnorm), does the regime of stable learning rates shrink at the widths considered. Additionally initializing the last layer to zero (ll$=0$) (as in the Jupyter notebook provided to reproduce Figure 3 in Yang et al. (2022)) shows even more pronounced learning rate shrinkage, but does not correspond to SP scaling anymore.

ResNets in SP show hyperparameter transfer across most SAM variants too, as soon as we tune momentum, weight decay and labelsmoothing (Figure H.11). This is in line with previous empirical observations (Yang et al., 2022, Figure 16 for SGD) but contradicts infinite-width theory as for MLPs.



(a) No momentum, weight decay or labelsmoothing

(b) Tuned momentum and weight decay

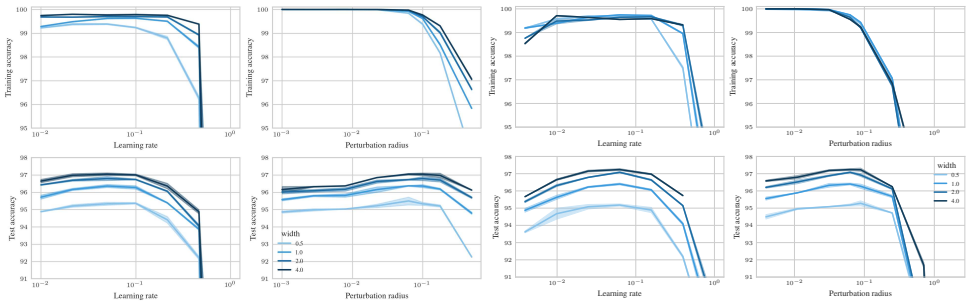
Figure H.11: Training accuracy (top) and test accuracy (bottom) after optimally stopping 100 epoch SAM training as a function of learning rate and perturbation radius in SP-naive without regularization (left) and with tuned regularization (right) using momentum 0.9, weightdecay 0.0005 and labelsmoothing 0.1. CI denote the minimal and maximal value from 4 independent runs. Without regularization, the optimal learning rate shrinks with width. Given the learning rate, the optimal perturbation radius seems quite stable, but since the optimal learning rate shifts, the performance scales worse than for μP^2 with the fixed learning rate that is tuned on the small model. With optimal regularization, both optimal learning rate and perturbation radius remain remarkably stable. We plan to investigate this mechanism in an upcoming work.

H.3.3 ResNets

In this section, we plot averages and σ-CI from 2 independent runs.

ResNets in μP^2 transfer both the optimal learning rate and perturbation radius for SAM (Figure H.12), SAM-ON (Figure H.14) and elementwise ASAM (Figure H.15), as well as different alternatives of scaling the gradient norm contributions to SAM's denominator (Figure H.18). This suggests correctness of the derived scalings. At width multipliers 2 and 4, μP^2 achieves the same or slightly better test accuracy than SP in all SAM variants.

Figure H.13 shows ResNets trained with SAM in different parameterizations. In ResNets of practical scale, ρ remains quite stable in μP^2 but surprisingly also in SP-NAIVE. In μP, for naive perturbation scaling the regime of stable perturbation radii shrinks, for global perturbation scaling, the optimal perturbation radius shifts, approaching its maximal stable value, which stays invariant to width scaling. Here, it would be interesting to see whether even larger width would lead to suboptimal performance of μP-global. μP^2 is most robust to the choice of ρ and achieves the best test accuracy.



(a) SP, no ρ scaling (b) μP^2

Figure H.12: Training accuracy (top) and test accuracy (bottom) after optimally stopping 200 epoch SAM training as a function of learning rate and of perturbation radius in SP (left) and in μP^2 (right) with optimized momentum 0.9, weight decay $5 \cdot 10^{-4}$ and labelsmoothing 0.1 for both μP^2 and SP. In μP^2, the base learning rate is $\eta = 2^{-4}$ and the base perturbation radius is $\rho = 2^{-4}$, in SP $\eta = 0.05$ and $\rho = 0.1$, respectively. Observe monotonic improvement with width in both training and test error. Optimal hyperparameters transfer across widths, surprisingly in both μP^2 and SP.

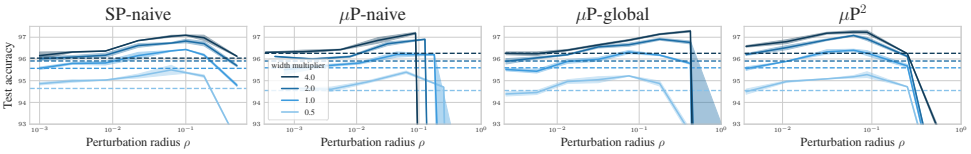


Figure H.13: Test accuracy after optimally stopping 200 epoch SAM training as a function of perturbation radius in various parameterizations. Dashed lines denote the base optimizer SGD with tuned momentum and weight decay in the respective parameterization.

H.3.4 ASAM variants

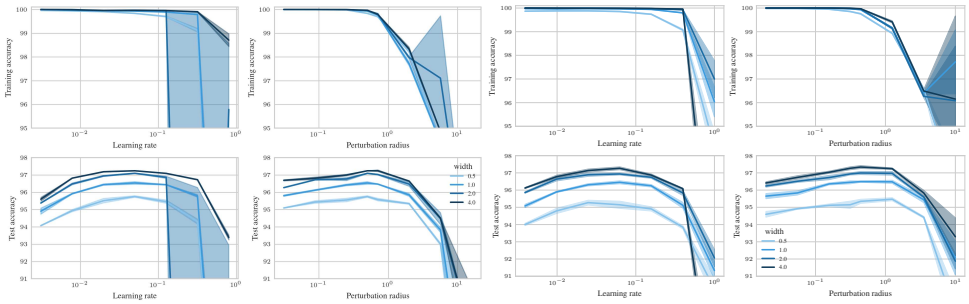
As we are not aware of any use of ASAM with MLPs in the literature and since the amount of necessary experiments for ViTs exceeds our computational budget, we only show that ResNets trained with the all of the discussed SAM variants in μP^2 transfer the optimal (η, ρ).

For the examples of elementwise ASAM and SAM-ON the global perturbation scaling $n^{1/2}$ suffices to reach μP^2. The stability of the optimal perturbation radius in the applied scaling $n^{1/2}$ shows that in μP with naive perturbation scaling the optimal perturbation radius would grow as $n^{1/2}$.

See the previous section, for a discussion of the remarkable stability of ResNets in SP. For the example of elementwise ASAM in SP, the optimal perturbation radius seems to grow.

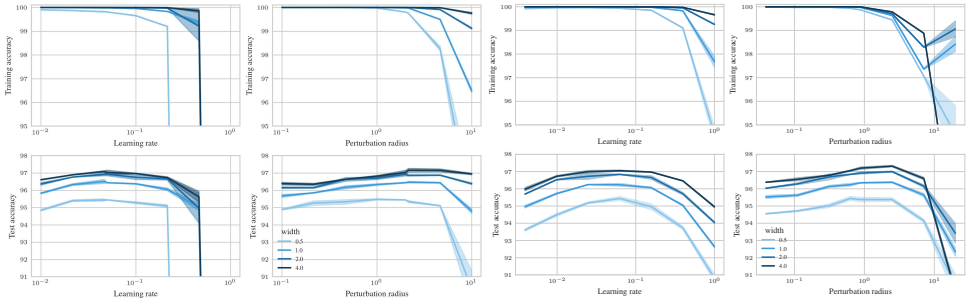
For layerwise ASAM (Figure H.16), the optimal perturbation radius seems to grow in both SP and μP^2, suggesting that our scaling condition does not perfectly apply to this variant, although μP^2 ($97.09_{\pm 0.03}(+0.83)$) still outperforms SP ($96.86_{\pm 0.05}(+0.83)$) in terms of the optimal test

accuracy. As Frobenius norms of weights are the only component that is not representable as a NE⊗OR⊤program, these Frobenius norms appear to scale differently than heuristically predicted over the course of training.



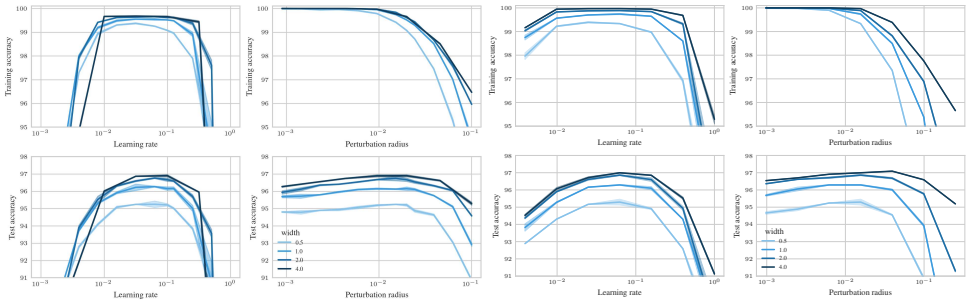
(a) SP, no ρ scaling (b) μP^2

Figure H.14: Same as Figure H.12 but for SAM-ON in SP without perturbation scaling (left) and in μP^2 (right). Both optimal learning rate and perturbation radius are remarkably stable in both μP^2 and SP. Since μP^2 for SAM-ON is just μP with global perturbation scaling $n^{1/2}$, transfer here implies that μP with width-independent scaling would not transfer.



(a) SP, no ρ scaling (b) μP^2

Figure H.15: Same as Figure H.12 but for elementwise ASAM in SP without perturbation scaling (left) and in μP^2 (right). Observe a consistent HP landscape in μP^2 but growing optimal perturbation radius in SP without perturbation scaling.



(a) SP, no ρ scaling (b) μP^2

Figure H.16: Same as Figure H.12 but for layerwise ASAM in SP without perturbation scaling (left) and in μP^2 (right). For layerwise ASAM, both μP^2 and SP seem to transfer the optimal learning rate as well as perturbation radius.

H.4 Gradient norm contributions have negligible effects on generalization performance

In this section we provide ablations concerning the question which layers should contribute non-vanishingly to the gradient norm in the denominator of the layerwise SAM perturbation rule (LP).

For MLPs, in Figure H.17 we scale all contributions to $\Theta(1)$, and then set the contribution of individual layers to zero, one by one. We observe no significant effect on the optimal test loss or hyperparameter transfer for MLPs. Any layer's contribution to the gradient normalization in the denominator of the SAM update rule can be set to 0 without a significant effect on the test loss. This raises the question which effect the gradient normalization has in μP. Does it contribute a scaling correction in SP, but may be dropped entirely in μP?

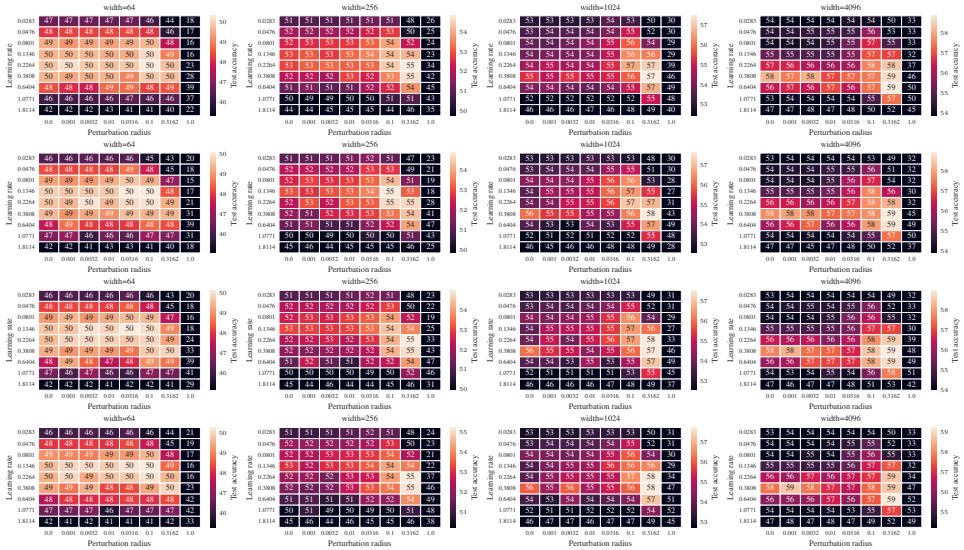
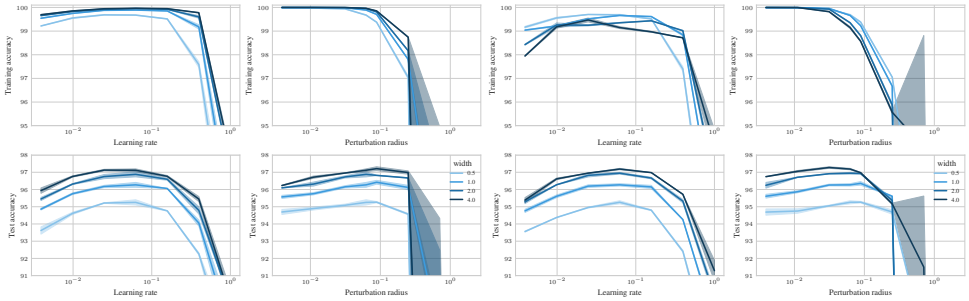


Figure H.17: Scaling the gradient norm contributions of all layers to $\Theta(1)$ (first row) and then setting the first layer gradient norm to 0 (2nd row), respectively the hidden layer (3rd row), last-layer (4th row). Each individual layer seems to have vanishing contribution to the optimal test error.

For ResNets, Figure H.18(a) shows accuracies when rescaling all layers' gradient norms to $\Theta(1)$, and Figure H.18(b) shows the results when using the original global gradient norm rescaled to $\Theta(1)$. Again, both methods achieve similar optimal test accuracy. The first variant shows cleaner hyperparameter transfer and monotonous improvement with width. When comparing to our original definition (LP) in Figure H.12, optimal performance is similar but rescaling all layers' gradient norm contributions to $\Theta(1)$ may even produce a slightly more stable hyperparameter-loss landscape for ResNets.



(a) Rescaling all of SAM's denominator terms to $\Theta(1)$ (b) Global $\|\nabla L\|$ scaling

Figure H.18: Same as Figure H.12 but with scaling of the gradient norms in the SAM perturbation (LP) denominator that scales all terms to $\Theta(1)$ (left) and only global denominator scaling $\frac{\|\nabla L\|}{n_L}$ (right). All denominator scalings achieve similar optimal accuracy, show HP transfer in learning rate and monotonic test accuracy improvement with width. In global denominator scaling, the optimal ρ shifts with width.

H.5 SAM with layerwise gradient normalization

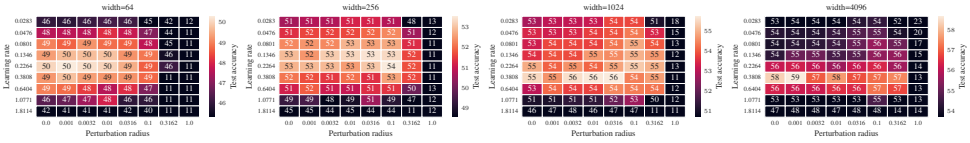
Here we consider SAM without the gradient normalization over all layers jointly. Instead we apply the layerwise perturbation rule presented in Appendix F.7,

$$\boldsymbol{\epsilon}_t^l = \rho_l \cdot \nabla_{W^l} \mathcal{L}(f(\xi_t; W_t), y_t) / \|\nabla_{W^l} \mathcal{L}(f(\xi_t; W_t), y_t)\|_F.$$

In SP, we consider a global constant $\rho_l = \rho$, whereas for μP^2 the spectral condition $(*)$ requires $\rho_l = \rho \cdot \sqrt{\texttt{fan\_out}/\texttt{fan\_in}}$.

Overall, SAM without layer coupling performs decently, but is outperformed by the original SAM in particular in ResNets, in μP^2 and at large width. But note that for ResNets we adopt the hyperparameters tuned for the original SAM with layer coupling, so that these ablations only serve as preliminary experiments.

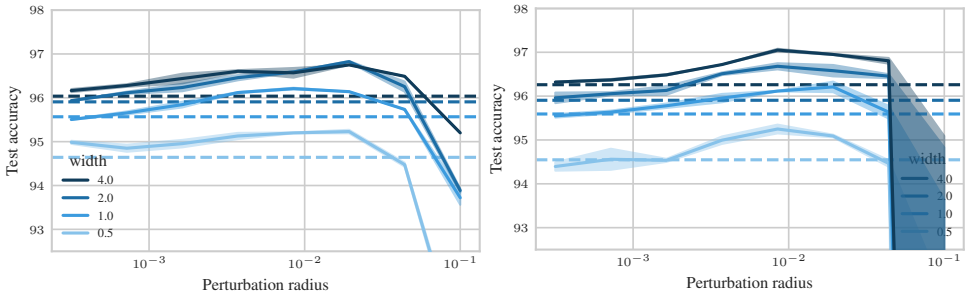
MLPs. SAM without layer coupling achieves similar optimal generalization in μP^2 at each width compared to Figure H.9. The regime of stable (η, ρ) stays width-independent, but does not transfer the optimum consistently. This suggests complex or noisy dependence of the training dynamics on ρ.



(a) μP^2

Figure H.19: **(SAM with layerwise normalization in MLPs)** Test accuracy as a function of learning rate η and perturbation radius ρ for an optimally-stopped MLP trained with SAM with layerwise normalization.

ResNets. Figure H.20 shows that decoupled SAM has decent performance, but is worse than original SAM with global normalization (Figure H.13) in both SP and μP^2, in particular at large width. As expected, ρ transfers in μP^2.



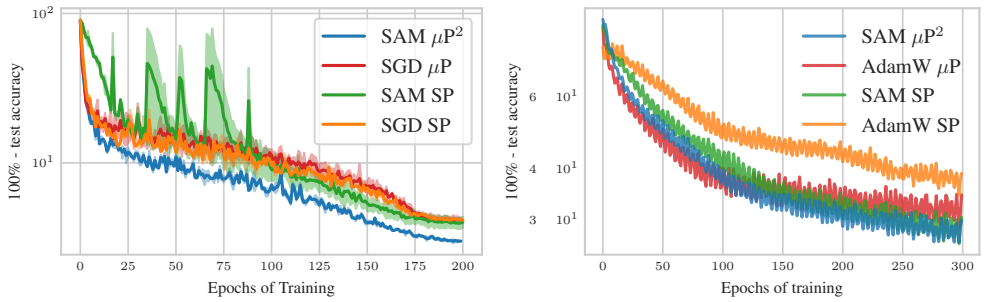
(a) SP (b) μP^2

Figure H.20: **(SAM with layerwise normalization in ResNets)** Test accuracy as a function of perturbation radius ρ for ResNets trained with SAM with layerwise normalization.

H.6 Test error over the course of training

Figure H.21 shows the test error of ResNets and ViTs over the course of training. μP^2 always achieves the best final test accuracy. In ResNets it also achieves a decent test accuracy the fastest and removes training instabilities of SAM in SP. While SGD in μP alone cannot compete with SAM in SP, SAM in μP^2 uniformly dominates over the entire course of training. Our theory suggests that in μP^2 the gradients are scaled correctly from the beginning, whereas in SP they have to self-stabilize first, which slows down convergence. We plan a closer analysis in an upcoming work.

In ViTs, μP generally achieves decent accuracy faster than SP, since gradient norms are already scaled correctly at initialization. SAM converges slower than the base optimizer AdamW in favor of drifting towards a better generalizing local minimum or saddle point. For ViTs at this moderate scale, SAM in SP catches up to SAM in μP^2 at the end of training.



(a) ResNet on CIFAR10 (b) ViT on CIFAR100

Figure H.21: Training a ResNet-18 with width multiplier 2 on CIFAR10 (left) and a ViT with width multiplier 2 on CIFAR100 (right). SGD and AdamW are the respective base optimizers.

NeurIPS Paper Checklist

1. **Claims**
   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?
   Answer: [Yes]
   Justification: In the abstract and introduction we state our main contributions while acknowledging related work. All main claims are theoretically proven and/or empirically verified.
   Guidelines:
   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.
2. **Limitations**
   Question: Does the paper discuss the limitations of the work performed by the authors?
   Answer: [Yes]
   Justification: Limitations are discussed in the future work section as well as in the section in the appendix that is related to the respective limitation.
   Guidelines:
   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.
3. **Theory Assumptions and Proofs**
   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?
   Answer: [Yes]
   Justification: We state all assumptions in the main paper and Appendix C, and provide all formal proofs in Appendix E.
   Guidelines:
   - The answer NA means that the paper does not include theoretical results.

- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental Result Reproducibility**
   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?
   Answer: [Yes]
   Justification: All experimental details are disclosed in Appendix G. Our perturbation scaling rules are clearly stated in the main paper. Their implementation with flexible `fan_in` and `fan_out` is explained in Appendix F.7, together with pseudocode for implementing our proposed scaling rule.
   Guidelines:
   - The answer NA means that the paper does not include experiments.
   - If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**
   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?
   Answer: [No]
   Justification: We only propose width-dependent scaling of hyperparameters of an existing optimization algorithm. This can be easily implemented by following the scaling rules that we clearly specify in the main paper. In Appendix F.7 we even provide a code example that

contains the essential modifications.We are working on making Python code to reproduce all of our experiments publicly available.
Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (https://nips.cc/public/guides/CodeSubmissionPolicy) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (https://nips.cc/public/guides/CodeSubmissionPolicy) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental Setting/Details**
   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?
   Answer: [Yes]
   Justification: All experimental details are disclosed in the main paper or Appendix G.
   Guidelines:
   - The answer NA means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.
7. **Experiment Statistical Significance**
   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?
   Answer: [Yes]
   Justification: As stated in Appendix G, we repeat all main experiments with multiple independent runs and report confidence bands within the empirical 2.5%- to 97.5%-quantiles. When we repeat experiments on Vision Transformers that we have also conducted on MLPs or ResNets, we do not use multiple runs due to limitations in computational resources.
   Guidelines:
   - The answer NA means that the paper does not include experiments.
   - The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).
   - It should be clear whether the error bar is the standard deviation or the standard error of the mean.
   - It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.

- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments Compute Resources**
   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?
   Answer: [Yes]
   Justification: We provide the type of GPU used and number of GPU seconds required for each experiment in Appendix G.
   Guidelines:
   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).
9. **Code Of Ethics**
   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?
   Answer: [Yes]
   Justification: We provide a theoretical analysis of a widely used optimization algorithm, point out the algorithm's limitations in large models and propose a correction. We do not foresee any ethical concerns.
   Guidelines:
   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).
10. **Broader Impacts**
    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?
    Answer: [NA]
    Justification: This paper provides fundamental research toward understanding and improving existing optimization algorithms for neural networks. We do not release any model or data and do not consider generative models.
    Guidelines:
    - The answer NA means that there is no societal impact of the work performed.
    - If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
    - The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
    - The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the

technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**
Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?
Answer: [NA]
Justification: We only use standard vision architectures and vision datasets in our experiments and do not release any data or models.
Guidelines:
- The answer NA means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**
Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?
Answer: [Yes]
Justification: We cite the standard CIFAR10, CIFAR100 (Krizhevsky et al., 2009) and ImageNet1K (Deng et al., 2009) datasets following the standard practice. We also cite the Python assets `PyTorch` (Paszke et al., 2019), `mup` (Yang et al., 2022) and the GitHub repository implementing SAM (Samuel, 2022) that we use as a basis for our experiments.
Guidelines:
- The answer NA means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New Assets**
Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?
Answer: [NA]
Justification: The paper does not release new assets.
Guidelines:
- The answer NA means that the paper does not release new assets.
- Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.

- The paper should discuss whether and how consent was obtained from people whose asset is used.
- At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and Research with Human Subjects**
    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?
    Answer: [NA]
    Justification: The paper does not involve crowdsourcing nor research with human subjects.
    Guidelines:
    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.
15. **Institutional Review Board (IRB) Approvals or Equivalent for Research with Human Subjects**
    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?
    Answer: [NA]
    Justification: The paper does not involve crowdsourcing nor research with human subjects.
    Guidelines:
    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

Chapter 6

On the Surprising Effectiveness of Large Learning Rates under Standard Width Scaling

On the Surprising Effectiveness of Large Learning Rates under Standard Width Scaling

Moritz Haas[1] **Sebastian Bordt**[1] **Ulrike von Luxburg**[1] **Leena Chennuru Vankadara**[2]

[1]University of Tübingen, Tübingen AI Center*
[2]Gatsby Computational Neuroscience Unit, University College London

Abstract

The dominant paradigm for training large-scale vision and language models is He initialization and a single global learning rate (*standard parameterization*, SP). Despite its practical success, standard parametrization remains poorly understood from a theoretical perspective: Existing infinite-width theory would predict instability under large learning rates and vanishing feature learning under stable learning rates. However, empirically optimal learning rates consistently decay much slower than theoretically predicted. By carefully studying neural network training dynamics, we demonstrate that this discrepancy is not fully explained by finite-width phenomena such as catapult effects or a lack of alignment between weights and incoming activations. We instead show that the apparent contradiction can be fundamentally resolved by taking the loss function into account: In contrast to Mean Squared Error (MSE) loss, we prove that under cross-entropy (CE) loss, an intermediate *controlled divergence* regime emerges, where logits diverge but loss, gradients, and activations remain stable. Stable training under large learning rates enables persistent feature evolution at scale in all hidden layers, which is crucial for the practical success of SP. In experiments across optimizers (SGD, Adam), architectures (MLPs, GPT) and data modalities (vision, language), we validate that neural networks operate in this controlled divergence regime under CE loss but not under MSE loss. Our empirical evidence suggests that width-scaling considerations are surprisingly useful for predicting empirically optimal learning rate exponents. Finally, our analysis clarifies the effectiveness and limitations of recently proposed layerwise learning rate scalings for standard initialization.

1 Introduction

Scaling has become the dominant paradigm in building ever more capable vision and language models (Brown et al., 2020, Dosovitskiy et al., 2021, Kaplan et al., 2020, Hoffmann et al., 2022, Grattafiori et al., 2024). Training these models requires many choices, including initialization variances, learning rates, and other optimization hyperparameters. The dominant practice for training large models is *standard parameterization* (SP): networks are initialized using He initialization (He et al., 2015) and trained with a single global learning rate tuned at each scale (OLMo Team et al., 2024).

Infinite-width theory provides principled rules for scaling hyperparameters as network size increases. In particular, *Maximal Update Parameterization (μP)* prescribes how learning rates and initialization variances should scale layer-wise so that the training dynamics remain width-independent. This allows tuning small proxy models and transferring the optimal hyperparameters to large scales (Yang and Hu, 2021, Yang et al., 2022). Crucially, infinite-width theory also predicts that under SP, network dynamics should become unstable with learning rates scaling larger than $\mathcal{O}(1/n)$ (where n

*Correspondence to: mo.haas@uni-tuebingen.de

Preprint. Under review.

is network width), and that feature learning vanishes with $\mathcal{O}(1/n)$ learning rates, causing the models to enter a kernel regime (Sohl-Dickstein et al., 2020, Yang and Hu, 2021). Empirically, however, networks trained in SP exhibit stable feature learning and excellent generalization performance, often with optimal learning rates decaying much slower than theoretically predicted (commonly around $\Omega(1/\sqrt{n})$). This is depicted in Figure 1, where we see that the optimal learning rates (solid lines) for different models trained in SP decay much slower than the theoretically predicted maximal stable scaling law (dashed gray lines). This discrepancy presents a fundamental puzzle:

Why does SP remain stable and effective at large learning rates, despite the theoretical predictions? And does there exist an infinite-width limit that corresponds more closely with the behaviour of practical finite-width networks?

One possible resolution could be that assumptions underlying infinite-width theory fail at realistic finite widths. Recent studies have highlighted phenomena like the *catapult effect* (Lewkowycz et al., 2020) and the *edge of stability* (Cohen et al., 2021), suggesting classical stability bounds underestimate viable learning rates. However, our analysis of simplified linear models under SP indicates that these finite-width effects alone cannot explain why large learning rate scaling remains stable. In a similar spirit, Everett et al. (2024) hypothesized that certain infinite-width theory predictions of alignment between weight updates and activations may break down in practice - which could also have plausibly explained the discrepancy between theory and practice. Through careful empirical measurements, we also rule out this hypothesis.

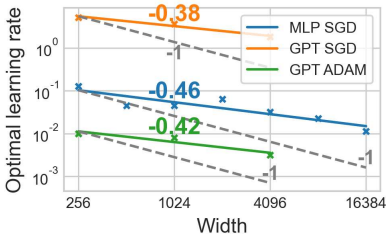


Figure 1: **Optimal learning rate exponents exceed the theoretically predicted stability threshold.** For MLPs on MNIST and GPT on language data, optimal learning rates in SP decay slower than the theoretically predicted maximal stable $\eta_n = \mathcal{O}(n^{-1})$ in gray.

Instead, we find a fundamental resolution in the previously overlooked role of the loss function. Specifically, unlike under MSE loss, where the effect of output logit divergence catastrophically cascades and destabilizes training, cross-entropy (CE) loss allows stable training under large learning rates even when output logits diverge. Consequently, the practical stability threshold in SP is determined solely by hidden or input layer stability constraints — not by output-layer divergence. Empirically, we find that while output-layer divergence under CE loss remains benign with respect to the stability threshold, it occasionally influences the optimal learning rate choice — particularly under SP with layerwise learning rates, as used by Everett et al. (2024). Moreover, we identify cases where output-layer divergence breaks the previously observed learning-rate transfer under layerwise learning rates.

Main Contributions. After reconciling the apparent contradictions between infinite-width theory and practice, we provide the first infinite-width proxy that has strong correspondence to practical finite-width networks, as they are initialized and trained in practice, in the following sense:

- Contrary to what was hypothesized in previous work (Everett et al., 2024), we show that the infinite-width alignment predictions between weights and incoming activations hold when measured with the right refined coordinate checks (RCC). Consequently, logits diverge at sufficient width in SP under empirically optimal learning rates.
- We show that the CE loss function enables a *controlled divergence* regime in which training remains stable despite logit divergence. This regime allows recovering feature learning at large widths under large learning rates $\eta_n = \Theta(n^{-1/2})$, which could explain the practical success of SP. To the best of our knowledge, this provides the first practical infinite-width limit in the feature-learning regime for SP.
- The controlled divergence regime also sheds light on the stability of other parameterizations, particularly SP with μP learning rates (SP-full-align, Everett et al., 2024). However, we empirically show that SP-full-align does not provide learning rate transfer on vision datasets due to inherent width dependence.
- We show that our width-scaling considerations provide surprisingly good predictions of maximal stable learning rate exponents, which often dominate optimal learning rates, particularly in Transformers (Vaswani et al., 2017).

Taken together, our results deepen the theoretical understanding of why SP remains effective at large scales, provide rigorous empirical validation for critical assumptions in infinite-width theory, and offer practical insights into stable hyperparameter transfer for scaling neural networks.

2 Background: Width-scaling arguments from Tensor Program theory

Before exploring plausible explanations for the empirical width-scaling properties of neural networks, we first define used notation and distill all necessary width-scaling arguments from Tensor Program (TP) theory (Yang and Hu, 2021, Yang and Littwin, 2023). We provide a more detailed introduction to TP scaling arguments in Appendix C.1, and a detailed account of related work in Appendix A.

Setting and Notation. We define an $(L+1)$-layer MLP of width n iteratively via

$$h^1(\xi) := W^1\xi, \qquad x^l(\xi) := \phi(h^l(\xi)), \qquad h^{l+1}(\xi) := W^{l+1}x^l(\xi), \qquad f(\xi) := W^{L+1}x^L(\xi),$$

for inputs $\xi \in \mathbb{R}^{d_{in}}$ with trainable weight matrices $W^1 \in \mathbb{R}^{n\times d_{in}}$, $W^l \in \mathbb{R}^{n\times n}$ for $l \in [2, L]$, and $W^{L+1} \in \mathbb{R}^{d_{\text{out}}\times n}$. We call h^l preactivations, x^l activations, and $f(\xi)$ output logits. Training the MLP with Stochastic Gradient Descent (SGD) with global learning rate $\eta > 0$ under loss function $\mathcal{L} : \mathbb{R}^{d_{\text{out}}} \times \mathbb{R}^{d_{\text{out}}} \to \mathbb{R}$ with labelled training point $(\xi_t, y_t) \in \mathbb{R}^{d_{in}} \times \mathbb{R}^{d_{\text{out}}}$ is defined as $W^l_{t+1} = W^l_t - \eta\nabla_{W^l}\mathcal{L}(f_t(\xi_t), y_t)$. We denote updates accumulated over all time steps by $\Delta h^l_t = h^l_t - h^l_0$ and the change from a single update step by $\delta h^l_t = h^l_t - h^l_{t-1}$. The fan-notation has the purpose of unifying all weight matrices and simply means $W \in \mathbb{R}^{\texttt{fan\_out}\times\texttt{fan\_in}}$. In this paper, we define *standard parameterization (SP)* to mean He initialization $(W^l_0)_{ij} \sim N(0, c_\phi/\texttt{fan\_in}(W^l_0))$ trained with SGD or Adam with a single possibly width-dependent learning rate $\eta_n = \eta \cdot n^\alpha$, $\alpha \in \mathbb{R}$, for all trainable weights $\{W^l_t\}_{l\in[L+1]}$. This models the typical practice, in which a global learning rate is tuned at each model scale. We denote the softmax function by $\sigma(f)_i = \exp(f_i) \cdot (\sum_{j\in[d_{\text{out}}]} \exp(f_j))^{-1}$. For naturally measuring the average scaling of entries in vectors $x \in \mathbb{R}^d$, we use the root-mean-squared norm $\|x\|_{RMS} := d^{-1/2} \cdot \|x\|_2$ as the standard vector norm. For matrices W, we write $\|W\|_F$ for the Frobenius norm and measure entry-wise scaling with the RMS norm $\|W\|_{RMS} := \left(\frac{1}{\texttt{fan\_in}\cdot\texttt{fan\_out}}\right)^{1/2} \|W\|_F$. The operator norm w.r.t. the RMS-norm is defined as $\|W\|_{op} := \|W\|_{RMS\to RMS} := \sup_{x\in\mathbb{R}^{\texttt{fan\_in}(W)}}(\|Wx\|_{RMS}/\|x\|_{RMS})$. We use Bachmann-Landau notation $\mathcal{O}, \Theta, \Omega$ that purely tracks dependence on width n and omits all other dependencies.

Effective and Propagating Updates. When training neural networks, weights W^l_t of layer l evolve from their initialization W^l_0 through updates ΔW^l_t, such that $W^l_t = W^l_0 + \Delta W^l_t$. Although we directly control the scaling of these initial weights and updates, we are ultimately interested in their impact on subsequent activations in the network. For standard architectures, including convolutional networks and Transformers, weights typically act linearly on incoming activations. Thus, for weights W^l_t and incoming activations x^{l-1}_t, the change in the next layer's pre-activations Δh^l_t can be decomposed into two distinct contributions: the *effective updates* arising directly from the change in weights ΔW^l_t of the current layer, and the *propagating updates*, arising indirectly from activation changes Δx^{l-1}_t in preceding layers:

$$\Delta h^l_t = \underbrace{(\Delta W^l_t)x^{l-1}_t}_{\text{Effective Updates}} + \underbrace{W^l_0(\Delta x^{l-1}_t)}_{\text{Propagating Updates}}. \tag{RCC}$$

We say the layer admits *maximal stable feature learning* if both the effective updates and propagating updates remain width-independent as network width $n \to \infty$, that is $\|(\Delta W^l_t)x_t\|_{RMS} = \Theta(1)$ and $\|W^l_0(\Delta x^{l-1}_t)\|_{RMS} = \Theta(1)$.

Identifying the correct scaling exponents. In the spirit of Everett et al. (2024), we use p_l and q_l to denote the width-scaling exponents of the *alignment ratios* of the pairs $(\Delta W^l_t, x^{l-1}_t)$ and $(W^l_0, \Delta x^{l-1}_t)$ respectively, that is,

$$\frac{\|\Delta W^l_t x^{l-1}_t\|_{RMS}}{\|\Delta W^l_t\|_{RMS} \cdot \|x^{l-1}_t\|_{RMS}} = \Theta(n^{p_l}), \quad \frac{\|W^l_0 \Delta x^{l-1}_t\|_{RMS}}{\|W^l_0\|_{RMS} \cdot \|\Delta x^{l-1}_t\|_{RMS}} = \Theta(n^{q_l}). \tag{α-rms}$$

A key insight from Yang and Hu (2021) and Yang and Littwin (2023) is that during training, correlations can emerge in certain layers between the two quantities in each pair in (RCC), causing

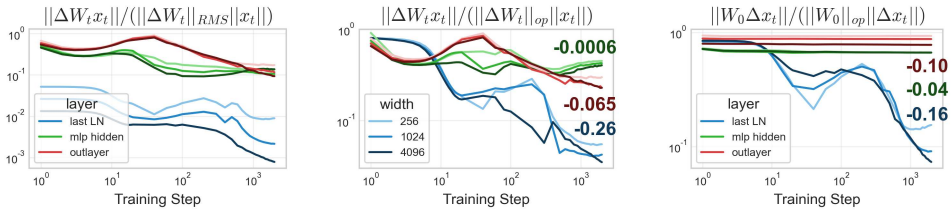


Figure 2: **Alignment has minimal width-dependence.** Alignment ratio between accumulated weight updates ΔW_t and incoming activations x_t in RMS norm (left) and operator norm (center) as well as between initial weights W_0 and activation updates Δx_t in operator norm (right) for the last layernorm layer, the first MLP layer in Transformer block 2 and the readout layer. RMS norm may be confounded by accumulated rank over the course of training (e.g. compare $(\Delta W_t, x_t)$ values for last LN). While operator norm alignment tends to decay over the course of training, it does not display strong width-dependence, even after 2000 batches (see annotated width-dependent exponents).

them to become *aligned* in the infinite-width limit and thereby inducing $p_l = 1$ and $q_l = 1$ due to a law of large numbers effect. If, instead, these quantities were uncorrelated, their product would exhibit smaller scaling exponents ($p_l = 1/2$ and $q_l = 1/2$) due to a central limit effect. In particular, infinite-width theory predicts the exponents $p_{1:L+1} = 1$, $q_{1:L} = 1/2$, and $q_{L+1} = 1$. The alignment exponents p_l, q_l are a consequence of the training dynamics and do not depend on the specific parameterization used (e.g., SP, NTP, or μP).

By adjusting the initialization variance, which controls the scale of initial weights W_0, and the learning rate, which governs the magnitude of updates ΔW_t, we can ensure that both contributions in (RCC) remain width-independent as the network width n grows. The corresponding choice of hyperparameter scaling defines the *Maximal Update Parameterization* (μP). As we will discuss in Section 4, under the theoretically predicted alignment exponents, SP with $\mathcal{O}(1/n)$ learning rates leads to vanishing activation updates in all layers $\|\Delta x_t\|_{RMS} = o(1)$ and choosing the learning rate $\omega(1/n)$ leads to logit divergence in the infinite-width limit.

3 Finite-width distortions and long-training dynamics alone do not explain the stability of large learning rates in SP

Differing optimal learning rate exponents at finite width versus in the infinite-width limit may be caused by finite-width effects accumulating over many update steps and eventually inducing a phase transition, in particular when the number of update steps exceeds the width of the network. Here we investigate two such potential explanations.

3.1 Update alignment between weights and activations is barely width-dependent

Everett et al. (2024) highlight that at finite width and over extended training times, it is a priori unclear whether the pairs $(\Delta W_t^l, x_t^{l-1})$ and $(W_0^{L+1}, \Delta x_t^L)$ remain strongly correlated or whether their alignment exponents $(p_{1:L+1}, q_{L+1})$ should rather be thought of as dynamically changing over the course of training. If the alignment exponents instead transition towards the central-limit regime and in particular if $p_{1:L+1} = -1/2$, this could explain the observed $\sqrt{n}$ gap between theoretically predicted and empirically observed optimal learning rate scalings.

Yang et al. (2023a) introduces an alignment metric that serves as a natural and unified metric to evaluate these infinite-width theory predictions,

$$\alpha_{A,x} = \frac{\|Ax\|_{\text{RMS}}}{\|A\|_{\text{RMS}\to\text{RMS}}\|x\|_{\text{RMS}}}. \qquad (\alpha\text{-op})$$

Specifically, if the alignment exponents $p_{1:L+1} = 1,\ q_{1:L} = \frac{1}{2},\ q_{L+1} = 1$ hold, both contributions in (RCC) must exhibit alignment metrics $\alpha_{\Delta W_t^l x_t^{l-1}}$ and $\alpha_{W_0^l \Delta x_t^{l-1}}$ of order $\Theta(1)$ in all layers.

In Figure 2, we plot the alignment metrics at varying widths over the course of Transformer training with AdamW in SP. It shows that while alignment can decrease over the course of training, it exhibits minimal dependence on network width. Even after accumulating approximately 2000 batches of training, the width-scaling exponents are much closer to 0 than to -0.5, indicating that infinite-width alignment predictions hold reasonably well. Hence a lack of alignment alone cannot explain the large optimal learning rate exponents observed in practice.

3.2 Does a catapult mechanism in the first update steps stabilize large learning rates in SP?

As another plausible explanation, initial divergence under large learning rates may be stabilized over the course of training at finite width. Unlike at infinite width, where there only exist a divergent regime and a lazy regime without feature learning, an intermediate catapult regime was identified by Lewkowycz et al. (2020) at finite width, for SGD training with MSE loss in Neural Tangent Parameterization (NTP). They provide theory for 2-layer linear networks. Under small learning rates $\eta \leq 2/\lambda_0$, where λ_0 denotes the largest eigenvalue of the Hessian at initialization, the network monotonically converges to a minimum. Under large learning rates $\eta > 4/\lambda_0$, training diverges. But in an *edge of stability* regime (Cohen et al., 2021, 2022) of intermediate learning rates, the loss increases in the first $\mathcal{O}(\log(n))$ update steps while the sharpness λ_t decreases. Once the sharpness lies below the edge of stability $2/\eta$, the loss decreases and the final learned function may generalize better as the solution lies in a basin with lower sharpness. But existing work does not study width-scaling with SP. May similar initial training dynamics be at play here?

In Appendix C.4 we analyse the 2-layer linear network model from Lewkowycz et al. (2020) in NTP, SP and μP trained with SGD under MSE loss, and provide loss and sharpness increase characterizations in Proposition C.17. In μP, the update equations of the learned function f_t and the sharpness λ_t are fully width-independent, which allows width-independent learning rates. In NTP, at least the conditions for loss and sharpness reduction are approximately width-independent. In SP, on the other hand, sharpness increases $\lambda_{t+1} \geq \lambda_t$ iff $\lambda_t \geq \frac{4}{n\eta_n}(1 + \frac{y}{f_t - y})$, requiring $\eta_n = \mathcal{O}(n^{-1})$ to avoid sharpness (as well as loss) divergence in the first update steps. The simulations shown in Figure C.2 validate the maximal stable learning rate scaling $\eta = \mathcal{O}(n^{-1})$. Hence catapult dynamics alone do not suffice for explaining large learning rate stability in SP.

4 Cross-entropy loss enables stable feature learning under large learning rates in standard parameterization

First, let us briefly recall why infinite-width theory predicts divergence under SGD training in SP with learning rates $\eta_n = \eta \cdot n^{-\alpha}$ for $\alpha < 1$.

Recall that the alignment exponents in (α-rms) satisfy $p_{1:L+1} = 1$. In particular, for the output layer, we have $\|\Delta W_t^{L+1} x_t^L\|_{RMS} = \Theta(n \cdot \|\Delta W_t^{L+1}\|_{RMS} \cdot \|x_t^L\|_{RMS})$. For SGD, the weight update of the last layer after 1 update step is given by $\Delta W^{L+1} = -\eta \cdot n^{-\alpha} \cdot \chi_0 \cdot (x_0^L)^T$, where $\chi_0 := \partial_f \mathcal{L}(f_0(\xi_0), y_0)$. Under SP, at initialization, both $\|x_0^L\|_{RMS} = \Theta(1)$ and $\|\chi_0\|_{RMS} = \Theta(1)$. This implies logit divergence after 1 step of SGD with learning rates $\eta_n = \omega(1/n)$:

$$\|x^L\|_{RMS} = \Theta(1),\ \|\Delta W^{L+1}\|_{RMS} = \Theta(n^{-\alpha}),\ \implies\ \|\Delta W^{L+1} x^L\|_{RMS} = \Theta(n^{1-\alpha})$$

So, *why do larger learning rates remain stable and even effective, despite logit divergence?*

Here, we demonstrate that a simple yet fundamental aspect of training, the choice of loss function, resolves the large learning rate puzzle, and enables a well-defined and practical infinite-width limit that allows feature learning under SP. The key insight is that, under cross-entropy (CE) loss, the logits f never directly appear in the training dynamics; instead, the effective output function is $\sigma(f)$. Unlike the destabilizing logit blowup encountered under mean squared error (MSE) loss, under CE loss, logit growth has a harmless effect on training stability. In particular, the CE loss introduces an intermediate *controlled divergence* regime $\alpha \in [1/2, 1)$ that is absent for the MSE loss (Figure 3).

Proposition 1. ***(Asymptotic regimes in SP, informal)*** *For fixed $L \geq 2$, $t \geq 1$, $\eta > 0$, $\alpha \in \mathbb{R}$, consider training a $(L+1)$-layer MLP of width n in SP with SGD and global learning rate $\eta_n = \eta \cdot n^{-\alpha}$ for t steps. Then the logits f_t, training loss $\mathcal{L}(f_t(\xi_t), y_t)$, loss-logit derivatives $\chi_t := \partial_f \mathcal{L}(f_t(\xi_t), y_t)$, loss-weight gradients $\nabla_t^l := \nabla_{W^l} \mathcal{L}(f_t(\xi_t), y_t)$ and activations x_t^l, $l \in [L]$, after training scale as follows in the infinite-width limit $n \to \infty$.*

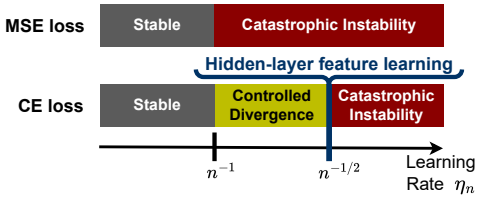


Figure 3: **Learning rate regimes for SGD in SP.** Under MSE loss, either training remains stable ($\alpha \geq 1$) or logits and activations diverge ($\alpha < 1$) in the infinite-width limit. Under CE loss, a 'controlled divergence' regime $\alpha \in [1/2, 1)$ emerges where logits diverge, but training does not diverge. At $\alpha = 1/2$, hidden layers learn features width-independently.

***Under cross-entropy (CE) loss**, three qualitatively distinct regimes arise:*

- *(a)* ***Stable regime*** *($\alpha \geq 1$): Logits, loss, gradients and activations remain stable, that is $\|f_t\|_{RMS} = \mathcal{O}(1)$, $|\mathcal{L}(f_t(\xi_t), y_t)| = \mathcal{O}(1)$, $\|\chi_t\|_{RMS} = \mathcal{O}(1)$, $\|\nabla_t^l\|_{RMS} = \mathcal{O}(n^{-1/2})$ and $\|x_t^l\|_{RMS} = \mathcal{O}(1)$ for all $l \in [L]$.*
- *(b)* ***Controlled divergence*** *($\frac{1}{2} \leq \alpha < 1$): Logits diverge $\|f_t\|_{RMS} = \Theta(n^{1-\alpha})$, but loss, gradients and activations remain stable, that is $\|x_t^l\|_{RMS} = \Theta(1)$, $|\mathcal{L}(f_t(\xi_t), y_t)| = \mathcal{O}(1)$, $\|\chi_t\|_{RMS} = \mathcal{O}(1)$ and $\|\nabla_t^l\|_{RMS} = \mathcal{O}(n^{-1/2})$ for all $l \in [L]$.*
- *(c)* ***Catastrophic instability*** *($\alpha < \frac{1}{2}$): Logits, activations and weight gradients diverge, that is $\|f_t\|_{RMS} \to \infty$, $\|x_t^l\|_{RMS} \to \infty$ and $\|\nabla_t^l\|_{RMS} \to \infty$, $l \in [2, L]$.*

***Under mean-squared error (MSE) loss**, a stable regime as in (a) above arises if $\alpha \geq 1$. If $\alpha < 1$, training is catastrophically unstable as in (c) above and, in addition, loss and loss-logit derivatives diverge, that is $|\mathcal{L}(f_t(\xi_t), y_t)| \to \infty$ and $\|\chi_t\|_{RMS} \to \infty$.*

The formal statement together with a proof can be found in Appendix C.3. For an intuitive understanding of this result, note that the only effect that the choice of loss function $\mathcal{L}(f, y)$ has on the final learned function is through the loss-logit gradients $\chi_t := \partial_f \mathcal{L}(f_t(\xi_t), y_t)$ over the course of training. Under MSE loss, the loss gradients are given by the residuals $\chi_t = f_t(\xi_t) - y_t$. But CE loss induces loss gradients $\chi_t = \sigma(f_t(\xi_t)) - y_t$. Crucially, it is the correct choice of loss function to effectively view $\sigma(f)$ as the output of the network instead of the unnormalized logits f. If one were to use $\text{MSE}(\sigma(f), y)$ as a loss function instead, additional derivative terms can induce vanishing gradients under exploding network output and not increase the optimal learning rate exponent (Appendix F.3). Under CE loss, the effective network output $\sigma(f)$ at most converges to one-hot predictions when the logits diverge, and with increasing width training points are sharply memorized after a single update step. At large learning rates $\eta_n = \Theta(n^{-1/2})$, training points are not just memorized in last-layer weights, but feature learning is recovered in the infinite-width limit:

Proposition 2 (Under CE loss, SP with large learning rates learns features at large width, informal). *Consider the setting of Proposition 1 of training a (L+1)-layer MLP with SGD in SP with global learning rate $\eta_n = \eta \cdot n^{-\alpha}$, $\alpha \in \mathbb{R}$, in the infinite-width limit $n \to \infty$.*

- *(a) Under both MSE and CE loss in the stable regime ($\alpha \geq 1$), feature learning vanishes in all layers $l \in [L]$, that is $\|\Delta x_t^l\|_{RMS} = \mathcal{O}(n^{-1/2})$.*
- *(b) Under CE loss in the controlled divergence regime ($\frac{1}{2} \leq \alpha < 1$), input layer feature learning vanishes at rate $\|\Delta x_t^1\|_{RMS} = \Theta\left(n^{-1/2-\alpha}\right)$, and hidden layers $l \in [2, L]$ learn features at rate $\|\Delta x_t^l\|_{RMS} = \Theta\left(n^{1/2-\alpha}\right)$. In particular, when $\alpha = 1/2$, the weight updates of all hidden layers induce width-independent activation updates, that is $\|\Delta x_t^l\|_{RMS} = \Theta\left(1\right)$.*

To the best of our knowledge, this provides the first infinite-width limit of SP in the practical feature learning regime. Figure 4 empirically validates that the predicted width-scaling exponents that induce maximally stable feature learning despite logit blowup under $\eta_n = \eta \cdot n^{-1/2}$ are already accurate at moderate width 512. Appendix E.4 shows that effective update predictions also hold accurately in Transformers trained with Adam. In the next section, we discuss the profound implications that training stability under logit blowup has on learning rate scaling exponents in practice.

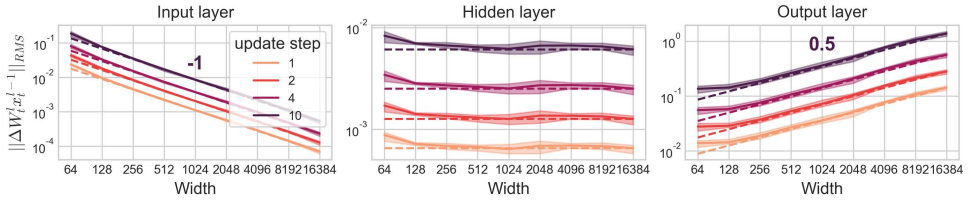


Figure 4: **Hidden-layer feature learning albeit logit divergence in SP under large learning rates.** Effective l-th layer update scalings $\|\Delta W_t x_t\|_{RMS}$ of MLPs trained with SGD in SP with $\eta_n = 0.0001 \cdot (n/256)^{-1/2}$ on CIFAR-10 under CE loss. Our TP scaling predictions are accurate: Hidden layers learn features width-independently, and input layers have vanishing feature learning. The update scaling exponents can already be accurately estimated at small width $n \leq 512$.

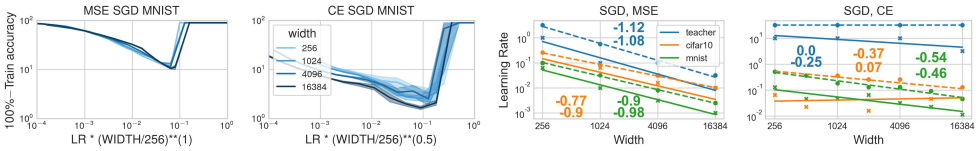


Figure 5: **Learning rates decay slower under CE loss than under MSE loss.** *Left versus center-left:* Width-scaled learning rate versus training error for MNIST showing approximate transfer with $\eta_n = \eta \cdot n^{-1}$ under MSE loss versus with $\eta_n = \eta \cdot n^{-1/2}$ under CE loss. *Center-right versus right:* Optimal learning rate (solid) and minimal unstable learning rate (dashed) for 2-layer MLPs on generated multi-index data and 8-layer MLPs on CIFAR-10 and MNIST. Optimal learning rates are often close to max-stable learning rates. Theoretical instability predictions $\mathcal{O}(n^{-1})$ for MSE loss, $\mathcal{O}(1)$ for 2-layer MLPs and $\mathcal{O}(n^{-1/2})$ for deep MLPs under CE loss are surprisingly accurate.

5 Consequences of training stability under logit divergence

In this section we perform extensive experiments to empirically evaluate the implications of the stability and feature learning predictions of our infinite-width theory from the previous section.

Experimental details. We train MLPs of varying depth up to width 16384 with plain SGD and Adam on CIFAR-10, MNIST and a generated multi-index model (reported in Appendix F). We also train Pythia-GPTs with warmup and cosine learning rate decay on the DCLM-Baseline dataset (Li et al., 2024) up to width 4096 or 1.4B parameters using both Adam with decoupled weight decay (Loshchilov and Hutter, 2019) and SGD (reported in Appendix F.2). If not stated otherwise, we consider SP with a global learning rate. All details can be found in Appendix D. Code will be made publicly available upon acceptance.

5.1 Infinite-width theory is a useful predictor of empirical optimal learning rate exponents

While our theory predicts maximal stable learning rates, in practice, we are interested in optimal learning rates. We hypothesize that maximal stable feature learning in all layers induces optimal performance at large width. However, since different layer types require different maximal stable learning rate exponents, the single global learning rate under SP is subject to opposing forces for recovering feature learning under the constraint of training stability. We now evaluate several instantiations of this hypothesis.

MLPs and Transformers with SGD. Figures 5 and 6 show that the empirical maximal learning rate exponents under CE loss closely follow $\alpha = 1/2$ for both MLPs on vision data and for GPTs on language data. The x-axis scales the learning rate with the closest clean exponent from $\{0, 0.5, 1\}$ to show that approximate empirical transfer is often enforced by the stability threshold $\mathcal{O}(n^{-1/2})$. While the theory only predicts the maximal stable exponent, Proposition 2 suggests that the optimal learning rate may follow the maximal stable exponent: $\alpha = 1/2$ since it is the only setting under which feature learning is preserved at large width in all hidden layers. The maximal stable learning rate under MSE loss also consistently scales as its infinite-width prediction $\mathcal{O}(n^{-1})$ and optimal learning rates closely follow this exponent, as under smaller exponents $\alpha > 1$, not even logits are updated

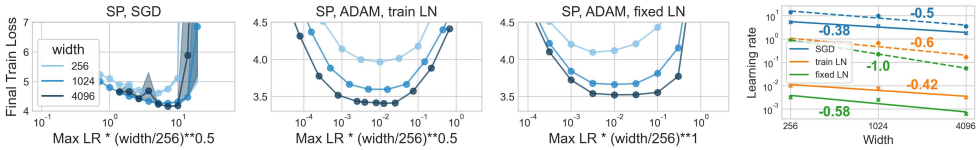


Figure 6: **Approximate learning rate transfer for GPT in SP.** *Left to center-right:* Width-scaled learning rate versus training loss for GPT trained with SGD, Adam with trainable Layernorm parameters and Adam without trainable Layernorm parameters. *Right:* Corresponding optimal (solid) and maximal stable (dashed) learning rate exponents. For SGD, hidden-layer stability $\eta_n = \mathcal{O}(n^{-1/2})$ clearly dominates the maximal stable as well as optimal learning rate scaling. For Adam without Layernorm parameters, hidden-layer stability induces a stability threshold $\eta_n = \mathcal{O}(n^{-1})$. Trainable Layernorm parameters further stabilize large learning rates and induce larger optimal learning rate scaling $\eta_n \approx \Theta(n^{-1/2})$ toward preserving input-layer feature learning at scale.

$\|\Delta f_t\|_{RMS} \to 0$. Overall, this shows that existing infinite-width theory was indeed predictive of the maximal stable learning rate exponents under MSE loss, but that CE loss induces qualitatively more favorable behaviour that is only captured by a sufficiently loss-specific analysis.

MLPs with Adam. Adam approximately normalizes the gradient and therefore further stabilizes training against misscaled gradients beyond the effect of CE loss, under sufficiently small $\varepsilon > 0$. W^l is effectively updated if the learning rate scaling counteracts the scaling accumulated in the inner product between normalized weight gradients and incoming activations. This leads to the ideal (μP) learning rates $\eta(W^l) = \eta/\texttt{fan\_in}(W^l)$. Thus Adam in SP with $\eta_n = \Theta(n^{-1})$ induces width-independent updates, except for vanishing input layer feature learning and logit divergence through $W_0^{L+1}\Delta x_t^L$. In deep MLPs, we typically observe optimal learning rates $\eta_n = \mathcal{O}(n^{-1})$, suggesting that hidden-layer stability dominates.

Transformer training with AdamW. In Transformers with trainable Layernorm parameters, which scale input-like, training is stabilized, and the exponent is increased toward input layer feature learning. Without trainable Layernorm parameters, in contrast, only the embedding layer scales input-like so that training becomes approximately width-independent under $\eta_n = \Theta(n^{-1})$. Figure 6 shows that the max-stable and optimal learning rate exponents shrink from $-1/2$ toward -1 if we remove the trainable layer-norm parameters. This suggests that trainable scale parameters in normalization layers play an essential role in maintaining learning rates in Transformers, which could explain why they are almost unanimously used in modern architectures (OLMo Team et al., 2024, Grattafiori et al., 2024, Gemma Team et al., 2024). Moreover, input layer learning vanishes at scale in SP, which may explain techniques like removing weight decay in the embedding layer (OLMo Team et al., 2024). Logit divergence under large learning rates may be a reason for regularizing techniques like the Z-loss (Chowdhery et al., 2023, Wortsman et al., 2024, OLMo Team et al., 2024).

Taken together, the empirical evidence suggests that infinite-width theory may serve as a helpful proxy for understanding practical neural networks at finite width. Since training divergence imposes a hard constraint on the optimal learning rate and activation divergence in multiple layers becomes harder to stabilize, width-scaling predictions seem to hold even more accurately on deep and sensitive architectures such as Transformers.

5.2 A novel understanding of standard initialization with layerwise learning rates

Everett et al. (2024) perform extensive Transformer experiments, and recommend training with Adam in SP with μP learning rates (SP-full-align) as the overall best performing parameterization in terms of validation loss, learning rate transfer and learning rate sensitivity. This parameterization only differs from μP through the larger last-layer He initialization $W_0^{L+1} \sim N(0, n^{-1})$. While the authors attribute the success of SP-full-align to a lack of alignment between W_0^{L+1} and Δx_t^L, they only measure the joint alignment between W_t and x_t for each layer, which confounds the individual alignment exponents of $(\Delta W_t, x_t)$ and $(W_0, \Delta x_t)$ from (RCC). We provide a detailed explanation in Appendix C.2. Our empirical alignment reevaluation in Figure 2 and Figure E.15 does not support the hypothesized lack of alignment. This implies that logits diverge through $W_0^{L+1}\Delta x_t^L$ as soon as feature learning does not vanish. Instead our theoretical results in Section 4 show that logit divergence

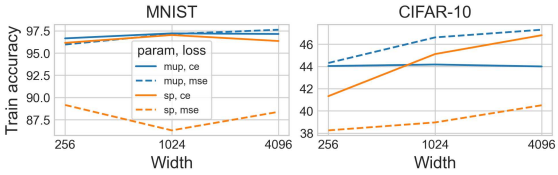


Figure 7: **Performance difference between losses is larger in SP than in μP.** Optimal training accuracy of 8-layer MLPs trained with SGD on MNIST (left) and CIFAR-10 (right). The performance in μP depends much less on the loss function since all layers learn width-independently.

is not necessarily harmful for training stability under CE loss. Just like SP with $\eta_n = \Theta(n^{-1/2})$, SP-full-align with $\eta_n = \Theta(1)$ lies at the feature learning edge of the controlled divergence regime.

Learning rate transfer of SP-full-align breaks on image datasets. Due to width-independent alignment between W_0^{L+1} and Δx_t^L, logits diverge with width in SP-full-align at sufficient width. We validate this claim for CIFAR-10 at moderate width in Figure F.34. This introduces width-dependent training dynamics. Consequently our single-pass experiments in Appendix F.8 consistently show decaying optimal learning rates in SP-full-align for both SGD and Adam on MNIST, CIFAR-10 and generated multi-index data. We also observe that the maximal stable learning rate remains width-independent as our theory would predict. This constitutes our only experiment in which the maximal stable learning rate scaling is suboptimal in deep nonlinear networks. We leave fully understanding the driving mechanism to future work.

5.3 A scaling-theoretic view on the practical success of CE loss in deep learning

Many success stories in deep learning, from computer vision to natural language processing, use the cross-entropy loss. We propose a scaling-theoretic explanation for this practical dominance. Our results show that networks trained under CE loss allow stable optimization at significantly larger learning rates in SP than under MSE loss, which recovers feature learning at large widths and consequently improves generalization. To empirically investigate this hypothesis, we compare the performance of CE and MSE losses under both SP and μP. Since μP admits asymptotically stable dynamics, both losses exhibit similar limiting behaviours. Thus we predict that CE loss only significantly outperforms MSE loss in SP, but not in μP. Our empirical findings confirm this prediction (Figure 7), which suggests that MSE loss may deserve renewed consideration as a practical choice under stable parameterizations like μP, especially given its theoretical simplicity and widespread use in theoretical analyses.

6 Discussion and future work

On the theoretical side, we have provided the first infinite-width proxy model for finite neural networks, as they are initialized and trained in practice. On the practical side, we have seen that infinite-width feature learning and instability predictions are surprisingly predictive indicators for empirical width-scaling exponents, in particular for deep Transformers.

Better understanding of the controlled divergence regime. Since practical neural networks operate at the edge of the controlled divergence regime, better understanding parameterizations beyond the stable regime from Yang and Hu (2021) is paramount. Since the NTK diverges in SP with $\eta_n = \Theta(n^{-1/2})$, studying this limit is subtle. However, investigating the rescaled NTK might still be a useful tool in better understanding this limit. While width dependence is undesirable from a transfer perspective, fast memorization under logit blowup may improve learning speed. How is generalization affected? Logit blowup may partially explain overconfidence in neural networks in SP, and suggests that wide networks in μP may be more calibrated.

Numerical considerations. In this paper, we consider the regime of sufficient numerical precision. From a numerical perspective, signals that diverge fast can leave floating point range at moderate widths. Hence implementations that ensure minimal accumulation of width-dependent factors in SP akin to Blake et al. (2025) could stabilize large-scale model training in practice.

Understanding optimal learning rate exponents. The exact conditions that induce hyperparameter transfer are still poorly understood. Without full width-independence, the optimal learning rate scaling cannot be predicted with certainty. Both vanishing feature learning in input-like layers and

logit divergence can induce strong finite-width effects, so that we would still recommend μP learning rates over SP from a width-scaling perspective. Similar to CE loss, normalization layers correct scaling in the forward pass. In combination with Adam which stabilizes the backward pass, such stabilizing components can correct most misscaled signals. Deeply understanding their interplay and effect on optimal learning rates remains an important direction for future work.

References

Emmanuel Abbe, Enric Boix Adsera, and Theodor Misiakiewicz. Sgd learning on neural networks: leap complexity and saddle-to-saddle dynamics. In *The Thirty Sixth Annual Conference on Learning Theory (COLT)*, 2023. Cited on page 17.

Jeremy Bernstein and Laker Newhouse. Modular duality in deep learning. *arXiv:2410.21265*, 2024. Cited on page 22.

Stella Biderman, Hailey Schoelkopf, Quentin Gregory Anthony, Herbie Bradley, Kyle O'Brien, Eric Hallahan, Mohammad Aflah Khan, Shivanshu Purohit, USVSN Sai Prashanth, Edward Raff, et al. Pythia: A suite for analyzing large language models across training and scaling. In *International Conference on Machine Learning (ICML)*, 2023. Cited on page 30.

Johan Bjorck, Alon Benhaim, Vishrav Chaudhary, Furu Wei, and Xia Song. Scaling optimal LR across token horizons. In *The Thirteenth International Conference on Learning Representations (ICLR)*, 2025. Cited on page 16.

Charlie Blake, Constantin Eichenberg, Josef Dean, Lukas Balles, Luke Yuri Prince, Björn Deiseroth, Andres Felipe Cruz-Salinas, Carlo Luschi, Samuel Weinbach, and Douglas Orr. u-μp: The unit-scaled maximal update parametrization. In *The Thirteenth International Conference on Learning Representations (ICLR)*, 2025. Cited on page 9, 17.

Blake Bordelon and Cengiz Pehlevan. Self-consistent dynamical field theory of kernel evolution in wide neural networks. *Advances in Neural Information Processing Systems (NeurIPS)*, 35: 32240–32256, 2022. Cited on page 16.

Blake Bordelon and Cengiz Pehlevan. Deep linear network training dynamics from random initialization: Data, width, depth, and hyperparameter transfer. *arXiv:2502.02531*, 2025. Cited on page 16.

Blake Bordelon, Hamza Tahir Chaudhry, and Cengiz Pehlevan. Infinite limits of multi-head transformer dynamics. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems (NeurIPS)*, 2024a. Cited on page 16.

Blake Bordelon, Lorenzo Noci, Mufan Bill Li, Boris Hanin, and Cengiz Pehlevan. Depthwise hyperparameter transfer in residual networks: Dynamics and scaling limit. In *The Twelfth International Conference on Learning Representations (ICLR)*, 2024b. Cited on page 16.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in Neural Information Processing Systems (NeurIPS)*, 33:1877–1901, 2020. Cited on page 1, 16, 48.

Lenaic Chizat and Francis Bach. On the global convergence of gradient descent for over-parameterized models using optimal transport. *Advances in neural information processing systems*, 31, 2018. Cited on page 16.

Lénaïc Chizat and Praneeth Netrapalli. The feature speed formula: a flexible approach to scale hyper-parameters of deep neural networks. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems (NeurIPS)*, 2024. Cited on page 16.

Lénaïc Chizat, Maria Colombo, Xavier Fernández-Real, and Alessio Figalli. Infinite-width limit of deep linear neural networks. *Communications on Pure and Applied Mathematics*, 77(10): 3958–4007, 2024. Cited on page 16.

Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, et al. Palm: Scaling language modeling with pathways. *Journal of Machine Learning Research*, 2023. Cited on page 8.

Jeremy Cohen, Simran Kaur, Yuanzhi Li, J Zico Kolter, and Ameet Talwalkar. Gradient descent on neural networks typically occurs at the edge of stability. In *International Conference on Learning Representations (ICLR)*, 2021. Cited on page 2, 5, 17.

Jeremy M Cohen, Behrooz Ghorbani, Shankar Krishnan, Naman Agarwal, Sourabh Medapati, Michal Badura, Daniel Suo, David Cardoze, Zachary Nado, George E Dahl, et al. Adaptive gradient methods at the edge of stability. *arXiv:2207.14484*, 2022. Cited on page 5, 17.

Alex Damian, Eshaan Nichani, and Jason D. Lee. Self-stabilization: The implicit bias of gradient descent at the edge of stability. In *ICLR*, 2023. Cited on page 17.

Francesco D'Angelo, Maksym Andriushchenko, Aditya Varre, and Nicolas Flammarion. Why do we need weight decay in modern deep learning? In *The Thirty-eighth Annual Conference on Neural Information Processing Systems (NeurIPS)*, 2024. Cited on page 17.

Li Deng. The mnist database of handwritten digit images for machine learning research. *IEEE Signal Processing Magazine*, 29(6):141–142, 2012. Cited on page 30, 68.

Nolan Dey, Bin Claire Zhang, Lorenzo Noci, Mufan Li, Blake Bordelon, Shane Bergsma, Cengiz Pehlevan, Boris Hanin, and Joel Hestness. Don't be lazy: Completep enables compute-efficient deep transformers. *arXiv:2505.01618*, 2025. Cited on page 16.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations (ICLR)*, 2021. Cited on page 1.

Katie Everett, Lechao Xiao, Mitchell Wortsman, Alexander A Alemi, Roman Novak, Peter J Liu, Izzeddin Gur, Jascha Sohl-Dickstein, Leslie Pack Kaelbling, Jaehoon Lee, et al. Scaling exponents across parameterizations and optimizers. *arXiv:2407.05872*, 2024. Cited on page 2, 3, 4, 8, 17, 18, 21, 45, 56, 57, 58, 59, 60, 61, 62.

Gemma Team, Thomas Mesnard, Cassidy Hardin, Robert Dadashi, Surya Bhupatiraju, Shreya Pathak, Laurent Sifre, Morgane Rivière, Mihir Sanjay Kale, Juliette Love, et al. Gemma: Open models based on gemini research and technology. *arXiv:2403.08295*, 2024. Cited on page 8.

Justin Gilmer, Behrooz Ghorbani, Ankush Garg, Sneha Kudugunta, Behnam Neyshabur, David Cardoze, George Edward Dahl, Zachary Nado, and Orhan Firat. A loss curvature perspective on training instabilities of deep learning models. In *International Conference on Learning Representations (ICLR)*, 2022. Cited on page 17.

Eugene A. Golikov. Dynamically stable infinite-width limits of neural classifiers. *arXiv:2006.06574*, 2020. Cited on page 16.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, et al. The llama 3 herd of models. *arXiv:2407.21783*, 2024. Cited on page 1, 8.

Moritz Haas, Jin Xu, Volkan Cevher, and Leena Chennuru Vankadara. μP^2: Effective sharpness aware minimization requires layerwise perturbation scaling. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems (NeurIPS)*, 2024. Cited on page 16.

Soufiane Hayou and Greg Yang. Width and depth limits commute in residual networks. In *International Conference on Machine Learning (ICML)*, 2023. Cited on page 16.

Bobby He, Lorenzo Noci, Daniele Paliotta, Imanol Schlag, and Thomas Hofmann. Understanding and minimising outlier features in transformer training. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems (NeurIPS)*, 2024. Cited on page 17.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *IEEE international conference on computer vision (ICCV)*, 2015. Cited on page 1, 30.

Jordan Hoffmann, Sebastian Borgeaud, Arthur Mensch, Elena Buchatskaya, Trevor Cai, Eliza Rutherford, Diego de Las Casas, Lisa Anne Hendricks, Johannes Welbl, Aidan Clark, et al. An empirical analysis of compute-optimal large language model training. *Advances in Neural Information Processing Systems (NeurIPS)*, 35:30016–30030, 2022. Cited on page 1, 16.

Arthur Jacot, Franck Gabriel, and Clément Hongler. Neural Tangent Kernel: Convergence and generalization in neural networks. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2018. Cited on page 16.

Dayal Singh Kalra, Tianyu He, and Maissam Barkeshli. Universal sharpness dynamics in neural network training: Fixed point analysis, edge of stability, and route to chaos. In *The Thirteenth International Conference on Learning Representations (ICLR)*, 2025. Cited on page 26.

Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. Scaling laws for neural language models. *arXiv:2001.08361*, 2020. Cited on page 1, 16.

Atli Kosson, Bettina Messmer, and Martin Jaggi. Rotational equilibrium: How weight decay balances learning across neural networks. In *Forty-first International Conference on Machine Learning (ICML)*, 2024. Cited on page 17.

Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009. Cited on page 30, 68.

Daniel Kunin, Allan Raventos, Clémentine Carla Juliette Dominé, Feng Chen, David Klindt, Andrew M Saxe, and Surya Ganguli. Get rich quick: exact solutions reveal how unbalanced initializations promote rapid feature learning. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024. Cited on page 17, 30.

Aitor Lewkowycz, Yasaman Bahri, Ethan Dyer, Jascha Sohl-Dickstein, and Guy Gur-Ari. The large learning rate phase of deep learning: the catapult mechanism. *arXiv:2003.02218*, 2020. Cited on page 2, 5, 17, 25, 26.

Jeffrey Li, Alex Fang, Georgios Smyrnis, Maor Ivgi, Matt Jordan, Samir Yitzhak Gadre, Hritik Bansal, Etash Guha, Sedrick Scott Keh, Kushal Arora, et al. Datacomp-lm: In search of the next generation of training sets for language models. *Advances in Neural Information Processing Systems (NeurIPS)*, 37:14200–14282, 2024. Cited on page 7, 31, 68.

Lightning AI. Litgpt. https://github.com/Lightning-AI/litgpt, 2023. Cited on page 30, 68.

Liyuan Liu, Haoming Jiang, Pengcheng He, Weizhu Chen, Xiaodong Liu, Jianfeng Gao, and Jiawei Han. On the variance of the adaptive learning rate and beyond. In *International Conference on Learning Representations (ICLR)*, 2020. Cited on page 17.

Peter J. Liu, Roman Novak, Jaehoon Lee, Mitchell Wortsman, Lechao Xiao, Katie Everett, Alexander A. Alemi, Mark Kurzeja, Pierre Marcenac, Izzeddin Gur, Simon Kornblith, Kelvin Xu, Gamaleldin Elsayed, Ian Fischer, Jeffrey Pennington, Ben Adlam, and Jascha Sohl-Dickstein. Nanodo: A minimal transformer decoder-only language model implementation in JAX. *GitHub repository*, 0.1.0, 2024. http://github.com/google-deepmind/nanodo. Cited on page 18.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations (ICLR)*, 2019. Cited on page 7.

Sam McCandlish, Jared Kaplan, Dario Amodei, and OpenAI Dota Team. An empirical model of large-batch training. *arXiv:1812.06162*, 2018. Cited on page 17.

Song Mei, Andrea Montanari, and Phan-Minh Nguyen. A mean field view of the landscape of two-layer neural networks. *Proceedings of the National Academy of Sciences*, 115(33):E7665–E7671, 2018. Cited on page 16.

Lorenzo Noci, Sotiris Anagnostidis, Luca Biggio, Antonio Orvieto, Sidak Pal Singh, and Aurelien Lucchi. Signal propagation in transformers: Theoretical perspectives and the role of rank collapse. *Advances in Neural Information Processing Systems (NeurIPS)*, 35:27198–27211, 2022. Cited on page 17.

Lorenzo Noci, Chuning Li, Mufan Li, Bobby He, Thomas Hofmann, Chris J Maddison, and Dan Roy. The shaped transformer: Attention models in the infinite depth-and-width limit. *Advances in Neural Information Processing Systems (NeurIPS)*, 36, 2024a. Cited on page 16.

Lorenzo Noci, Alexandru Meterez, Thomas Hofmann, and Antonio Orvieto. Super consistency of neural network landscapes and learning rate transfer. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems (NeurIPS)*, 2024b. Cited on page 16.

OLMo Team, Pete Walsh, Luca Soldaini, Dirk Groeneveld, Kyle Lo, Shane Arora, Akshita Bhagia, Yuling Gu, Shengyi Huang, Matt Jordan, et al. 2 olmo 2 furious. *arXiv:2501.00656*, 2024. Cited on page 1, 8.

Elliot Paquette, Courtney Paquette, Lechao Xiao, and Jeffrey Pennington. 4+3 phases of compute-optimal neural scaling laws. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems (NeurIPS)*, 2024. Cited on page 16.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2019. Cited on page 30, 68.

Dan Qiao, Kaiqi Zhang, Esha Singh, Daniel Soudry, and Yu-Xiang Wang. Stable minima cannot overfit in univariate relu networks: Generalization by large step sizes. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems (NeurIPS)*, 2024. Cited on page 17.

Yunwei Ren, Eshaan Nichani, Denny Wu, and Jason D Lee. Emergence and scaling laws in sgd learning of shallow neural networks. *arXiv:2504.19983*, 2025. Cited on page 17.

Christopher J Shallue, Jaehoon Lee, Joseph Antognini, Jascha Sohl-Dickstein, Roy Frostig, and George E Dahl. Measuring the effects of data parallelism on neural network training. *arXiv:1811.03600*, 2018. Cited on page 17.

Jascha Sohl-Dickstein, Roman Novak, Samuel S Schoenholz, and Jaehoon Lee. On the infinite width limit of neural networks with a standard parameterization. *arXiv:2001.07301*, 2020. Cited on page 2, 16.

Alexander Tsigler, Luiz FO Chamon, Spencer Frei, and Peter L Bartlett. Benign overfitting and the geometry of the ridge regression solution in binary classification. *arXiv:2503.07966*, 2025. Cited on page 17.

Leena Chennuru Vankadara, Jin Xu, Moritz Haas, and Volkan Cevher. On feature learning in structured state space models. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems (NeurIPS)*, 2024. Cited on page 16, 31.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in Neural Information Processing Systems (NeurIPS)*, 2017. Cited on page 2, 30.

Roman Vershynin. Introduction to the non-asymptotic analysis of random matrices. *arXiv:1011.3027*, 2010. Cited on page 42.

Nikhil Vyas, Alexander Atanasov, Blake Bordelon, Depen Morwani, Sabarish Sainathan, and Cengiz Pehlevan. Feature-learning networks are consistent across widths at realistic scales. *Advances in Neural Information Processing Systems (NeurIPS)*, 36, 2024. Cited on page 16, 17.

Jonathan Wenger, Felix Dangel, and Agustinus Kristiadi. On the disconnect between theory and practice of overparametrized neural networks. *arXiv:2310.00137*, 2023. Cited on page 16.

Mitchell Wortsman, Peter J Liu, Lechao Xiao, Katie E Everett, Alexander A Alemi, Ben Adlam, John D Co-Reyes, Izzeddin Gur, Abhishek Kumar, Roman Novak, Jeffrey Pennington, Jascha Sohl-Dickstein, Kelvin Xu, Jaehoon Lee, Justin Gilmer, and Simon Kornblith. Small-scale proxies for large-scale transformer training instabilities. In *The Twelfth International Conference on Learning Representations (ICLR)*, 2024. Cited on page 8, 17, 30, 46.

Ruibin Xiong, Yunchang Yang, Di He, Kai Zheng, Shuxin Zheng, Chen Xing, Huishuai Zhang, Yanyan Lan, Liwei Wang, and Tieyan Liu. On layer normalization in the transformer architecture. In *International Conference on Machine Learning (ICML)*, 2020. Cited on page 17.

Greg Yang. Wide feedforward or recurrent neural networks of any architecture are gaussian processes. *Advances in Neural Information Processing Systems (NeurIPS)*, 32, 2019. Cited on page 16.

Greg Yang and Edward J. Hu. Tensor programs iv: Feature learning in infinite-width neural networks. In *International Conference on Machine Learning (ICML)*, 2021. Cited on page 1, 2, 3, 9, 16, 19, 20, 21, 22, 24, 27, 57.

Greg Yang and Etai Littwin. Tensor programs ivb: Adaptive optimization in the infinite-width limit. *arXiv:2308.01814*, 2023. Cited on page 3, 16, 23, 24.

Greg Yang, Edward J Hu, Igor Babuschkin, Szymon Sidor, Xiaodong Liu, David Farhi, Nick Ryder, Jakub Pachocki, Weizhu Chen, and Jianfeng Gao. Tensor programs v: Tuning large neural networks via zero-shot hyperparameter transfer. *arXiv:2203.03466*, 2022. Cited on page 1, 16, 17, 18, 31.

Greg Yang, James B. Simon, and Jeremy Bernstein. A spectral condition for feature learning. *arXiv:2310.17813*, 2023a. Cited on page 4, 16, 31.

Greg Yang, Dingli Yu, Chen Zhu, and Soufiane Hayou. Tensor programs vi: Feature learning in infinite-depth neural networks. *arXiv:2310.02244*, 2023b. Cited on page 16.

Yang You, Jing Li, Sashank Reddi, Jonathan Hseu, Sanjiv Kumar, Srinadh Bhojanapalli, Xiaodan Song, James Demmel, Kurt Keutzer, and Cho-Jui Hsieh. Large batch optimization for deep learning: Training bert in 76 minutes. In *International Conference on Learning Representations (ICLR)*, 2020. Cited on page 17.

Zixiong Yu, Songtao Tian, and Guhan Chen. Divergence of empirical neural tangent kernel in classification problems. *International Conference on Learning Representations (ICLR)*, 2025. Cited on page 16.

Ruiqi Zhang, Jingfeng Wu, Licong Lin, and Peter L Bartlett. Minimax optimal convergence of gradient descent in logistic regression via large and adaptive stepsizes. *arXiv:2504.04105*, 2025. Cited on page 17.

Appendices

Appendix Contents.

A Detailed Related Work **16**

B Take-aways for practitioners **18**

C Theoretical considerations **19**
C.1 Distilled TP scaling arguments 19
C.2 Measuring Alignment 21
C.3 Formal statements and proofs of Propositions 1 and 2 22
C.4 Scaling dynamics in 2-layer linear networks 25

D Experimental details **30**
D.1 MLPs 30
D.2 Multi-index data 30
D.3 Language modeling 30
D.4 Figure Details 31

E Refined coordinate checks **31**
E.1 SGD 32
E.2 Adam 38
E.3 Normalization layers and Adam provide robustness to miss-initialization 41
E.4 Alignment and update scaling in Transformers 42

F Empirical learning rate exponents **44**
F.1 Summary of the MLP experiments in this section 44
F.2 Transformer experiments 45
F.3 Cross-entropy loss enables large-learning rate training 48
F.4 MLPs with SGD on MNIST 49
F.5 MLPs with ADAM on MNIST 51
F.6 MLPs with SGD on CIFAR-10 52
F.7 MLPs with ADAM on CIFAR-10 54
F.8 Effective update parameterizations beyond μP 56

A Detailed Related Work

Here we provide a more detailed account of related work than what is possible in the main body of the paper.

Neural networks in the infinite-width limit. Past work has extensively analysed the *Neural Tangent Parameterization (NTP)* (Jacot et al., 2018) due to its tractability. But due to lacking feature learning in the infinite-width limit, finite networks in NTP behave qualitatively differently and hence NTP is not the ideal model for understanding finite neural networks. Finite-width deviations already accumulate after a few steps of training (Wenger et al., 2023), in particular under CE loss (Yu et al., 2025). Considerable effort has been invested in finding a descriptive infinite-width model for SP. Sohl-Dickstein et al. (2020) note that the NTK diverges under large learning rates $\eta_n = \omega(n^{-1})$ in SP, which motivates them to consider a different parameterization which preserves a finite NTK in the infinite-width limit, but consequently does not correspond to SP anymore. Golikov (2020) studies a class of 'dynamically stable' parameterizations, allowing large learning rates under a variant of SP, they call 'sym-default' parameterization, which again is not equivalent SP. Another popular width-dependent parameterization is the Maximal Update Parameterization (μP). It achieves a width-independent effect of updates in all trainable weights on the output function. Its infinite-width limit has been observed to closely track finite networks in μP well over long periods of training in, for example, feature learning strength, the learned function, or gradient and Hessian statistics (Vyas et al., 2024, Noci et al., 2024b). As an important practical consequence, it allows to tune small proxy models and train the large model only once with the optimal HPs (Yang et al., 2022). μP was derived using *Tensor Programs (TP)* framework (Yang, 2019, Yang and Hu, 2021, Yang and Littwin, 2023) that, in theory, allows to exactly track the learning dynamics of many popular architectures like MLPs, ResNets and Transformers trained with SGD or Adam in arbitrary parameterizations in the infinite-width limit. Haas et al. (2024) derive a μP-like parameterization for sharpness aware minimization algorithms achieving transfer of the optimal learning rate and perturbation radius jointly by showing that perturbations should be scaled like updates in μP. Vankadara et al. (2024) derive an initialization and learning rate scaling rule that achieves width-independent training dynamics for the state-space model Mamba, which shows that the spectral condition on the weights and weight updates in every layer for achieving μP provided by Yang et al. (2023a) does not apply to arbitrary architectures. At sufficient numerical precision, the mean-field parameterization (Mei et al., 2018, Chizat and Bach, 2018) is equivalent to μP. While it was initially restricted to shallow neural networks, the dynamical mean-field theory (DMFT) by Bordelon and Pehlevan (2022) generalizes it to more complex architectures, including Transformers (Bordelon et al., 2024a). Although still expensive, the approximate solvers from DMFT are more computationally feasible than iteratively solving the exact TP limit equations. Chizat et al. (2024) studies deep linear networks in μP and shows convergence of gradient flow to a minimum l_2-norm solution.

Other neural network scaling limits. Beyond width scaling, depth scaling $L \to \infty$ has been studied in detail. For ResNets, Yang et al. (2023b), Hayou and Yang (2023), Bordelon et al. (2024b) show that $L^{-1/2}$-scaling of shallow residual blocks induces depth-independence and this limit commutes with width scaling, implying that depth can be scaled independent of width. Using approximative DMFT theory, Bordelon et al. (2024a) suggest that L^{-1}-depth scaling may be necessary to preserve feature learning in attention blocks although they consider a pure depth limit. Dey et al. (2025) confirm L^{-1}-block scaling to be the 'correct' scaling by providing additional desiderata and empirical evidence on Transformers. Bordelon et al. (2024a) also show that the infinite within-head dimension limit effectively leads to a single-head Transformer, an the infinite number of heads limit concentrates by aggregating over the coordinate distribution at fixed within-head size, closer to how scaling is typically performed in practice (Brown et al., 2020). Noci et al. (2024a) study a joint width and depth limit close to initialization for Transformers with the goal of preventing rank collapse. Long training time is much less understood. Bordelon and Pehlevan (2025) study the training dynamics of deep and wide linear networks trained on structureless Gaussian data. Chizat and Netrapalli (2024) considers the angle between activations and gradients to give scaling rules for hyperparameters toward automatic HP scaling. They correct output layer scaling of MLPs in μP depth-dependently, only for SGD.

Scaling laws. Robust compute-optimal scaling laws in LLMs were reported by Kaplan et al. (2020), Hoffmann et al. (2022). Paquette et al. (2024) provide theory on random feature models trained with one-pass SGD and identify 4 phases and 3 subphases depending on properties of the data and the target. Bjorck et al. (2025) observe no transfer across token horizons but a predictable scaling

law with exponent -0.32 on LLama. McCandlish et al. (2018) suggests that the optimal learning rate scales as $a/(1 + b/batchsize)$ with setting-dependent constants a, b. Hence for sufficiently large batch size the optimal learning rate is roughly constant, which is in line with the empirical observations by Shallue et al. (2018), Yang et al. (2022). Ren et al. (2025) study SGD training of 2-layer MLPs on isotropic Gaussian data under MSELoss and find that different teacher neurons are abruptly learned at different timescales leading to a smooth scaling law in the cumulative objective. Further work toward assessing the compute-optimal and data-optimal Pareto frontiers under realistic assumptions remains an important and challenging task for future work.

Finite width training dynamics. Understanding finite-width training dynamics complements infinite-width theory very well, as the former line of work operates at fixed width, while the latter ask what changes with increasing width. From a practical perspective, scaling networks with μP appears to preserve the properties from base width (Vyas et al., 2024, Noci et al., 2022). Deep understanding of neural network training dynamics is still limited to 2-layer nonlinear MLP (Ren et al., 2025, Zhang et al., 2025) or (deep) linear MLP (Kunin et al., 2024, Tsigler et al., 2025) toy models under strong distributional assumptions.

Kunin et al. (2024) explain for 2-layer networks that varying layerwise initialization variance and learning rate scaling induces differing learning regimes: fast feature learning in balanced parameterizations (desirable for linear nets), faster learning of earlier layers in upstream parameterizations with small parameter movement (desirable in nonlinear networks, as it reduces time to grokking and sample complexity of hierarchical data structures), faster learning of later layers in downstream initializations (that is initial lazy fitting followed by slow feature learning). Abbe et al. (2023) show that, opposed to lazy networks, feature learning networks can learn low rank spikes in hidden layer weights/kernels to help with sparse tasks. Qiao et al. (2024) show that large learning rates induce sparse linear spline fits in univariate gradient descent training by showing that all stable minima are flat, non-interpolating and produce small first order total variation, hence avoid overfitting and learn functions with bounded first order total variation

Edge of stability. Large learning rates have broadly been observed to induce optimal generalization. Lewkowycz et al. (2020) observe that under large learning rates at the edge of stability, $2/\lambda_0 < \eta < c_{arc}/\lambda_0$ (where $c_{arc} = 12$ for ReLU nets) an initial blowup at training time at least $\log(n)$ induces a bias towards flatter minima. (Cohen et al., 2021) find loss spikes during training, but that training self-stabilizes through sharpness reduction. Damian et al. (2023) develops some understanding of the mechanisms behind EOS dynamics. For Adam, the preconditioner matrix provides an additional mechanism by which stabilization can occur (Cohen et al., 2022, Gilmer et al., 2022).

Warmup. Warmup allows stability under larger learning rates via slow sharpness reduction (**?**). **?** also shows that warmup allows using larger learning rates than otherwise stable by constantly operating at the edge of stability. Warmup does not improve performance but stabilizes training; by allowing training with larger learning rates, these often induce improved performance. Large catapults harm Adam by persisting in its memory. Hence Adam's optimal learning rate is further away from the failure boundary than for SGD, and Adam benefits more from longer warmup. Above the optimal learning rate, Adam has a regime of training failure, where early catapults persist in the second moment and prevent learning. Warmup also widens the regime of near-optimal learning rate choices. Liu et al. (2020) find that particularly Adam needs warmup due to large initial variance.

Effective learning rates. Kosson et al. (2024) study the effect of weight decay on rotation in weight vectors, which influences the effective learning rate. Also see references therein for literature on effective learning rates, which is related to the alignment discussion in this paper.

Stability of Transformer training. More empirically, a plethora of works study the training stability of large-scale Transformers with respect to warmup, weight decay (D'Angelo et al., 2024), batch size (You et al., 2020), the optimizer (Kosson et al., 2024), the position of normalization layers (Xiong et al., 2020) and their interplay with the parameterization and numerical considerations (Wortsman et al., 2024, Blake et al., 2025, Everett et al., 2024). Wortsman et al. (2024) find that qk-Layernorm stabilizes Transformer training beyond the stabilizing effect from using μP. Xiong et al. (2020) propose pre-LN for enhanced training stability requiring less warmup. He et al. (2024) observe that outlier features (=extremely activated coordinates in activations) emerge quickly in Transformer training with AdamW and that rank-collapse under strong correlations between inputs is correlated with more outlier features. Non-diagonal preconditioning like SOAP and Shampoo resolves the issue.

Most relevant to our work, Everett et al. (2024) perform extensive and insightful experiments for NanoDO decoder-only Transformers (Liu et al., 2024) in SP, μP, NTP and mean field parameterizations with corrected layerwise learning rate scalings, questioning the infinite-width alignment predictions between weights and incoming activations at finite width over the course of long training. They recommend SP with ADAM in conjunction with μP-learning rate scaling (they call SP-full-align) as the best-performing empirical parameterization in terms of generalization, learning rate transfer and learning rate sensitivity.

B Take-aways for practitioners

In this paper, the term *parameterization* refers to width-dependent scaling of initialization and learning rate of each trainable weight tensor separately. Studying parameterizations then means applying a scaling rule for layerwise initialization variances and learning rates and understanding how relevant quantities such as update scaling in activations and logits evolves, and where instabilities may arise at large widths. At some fixed base width, all parameterizations can be considered equivalent, if we allow tuning constant multipliers.

For properly comparing the performance of parameterizations, constant weight and initialization multipliers should be tuned at some fixed base width for each parameterization at base width separately (Yang et al., 2022). This adjusts the layerwise activation and gradient size at finite width. The parameterization then prescribes the rule, by which the layerwise initialization and updates are rescaled when changing the width in relation to that base width $\texttt{width}/\texttt{base\_width}$. Alternatively, parameterizations may be equivalent at base width such that only one set of weight multipliers has to be tuned. The extensive LLM experiments in Everett et al. (2024) suggest that the advantage of large last-layer initialization may just be an artifact of the community extensively tuning performance in SP, and after also tuning all layerwise multipliers for μP, the performance difference vanishes.

While SP performs better than naive theory would predict, and can learn hidden-layer features width-independently under CE loss, feature learning still vanishes in input-like layers like embedding or Layernorm layers under both SGD and Adam. Still only μP learning rate scaling effectively updates all layers. Small last-layer initialization then recovers full width-independent and hence predictable scaling dynamics under sufficient precision in the regime $n \gg d_{\text{out}}$, whereas standard last-layer initialization induces logit blowup at sufficient width, which is not necessarily harmful for generalization but reduces predictability as scaling is not fully width-independent. Standard initialization with μP learning rates (SP-full-align) can induce 'practical transfer' and empirically update all weights effectively without logit blowup at moderate scales $n \ll d_{\text{out}}$ in the regime where the width is much smaller than the output dimension, as is relevant for NLP settings, but likely exhibits unexpected changed behaviour at sufficient scale, when logits start to diverge due to the last-layer term $W_0^{L+1}\Delta x_t^L$. This can be read off from differing dominating terms in $\|W_0^{L+1}\|_{RMS\to RMS}$, assuming width-independent alignment $\alpha_{W_0^{L+1}\Delta x_t^L} = \Theta(1)$ as we measure. For uniform non-asymptotic transfer for both $n \gg d_{\text{out}}$ and $n \ll d_{\text{out}}$, by the same argument we suggest a last-layer initialization $\sigma_{L+1} = (\frac{\texttt{fan\_in}}{\sqrt{\texttt{fan\_out}}} + \sqrt{\texttt{fan\_in}})^{-1}$, that transitions from SP initialization in the regime $n \ll d_{\text{out}}$ to μP initialization in the regime $n \gg d_{\text{out}}$.

For AdamW without using width-dependent weight multipliers, layer-balancing μP learning rates are simply given by the learning rate scaling $\eta(W) = \eta/\texttt{fan\_in}(W)$. Here, all biases as well as normalization layer weights should be understood as weights to the one-dimensional input 1, hence $\texttt{fan\_in} = 1$. For recovering width-independent weight decay, weight decay requires the inverse scaling $\texttt{wd} \cdot \texttt{fan\_in}(W)$.

TP-like width scaling arguments are very useful for identifying sources of divergence or shrinkage with scale, and architecture components such as normalization layers and training algorithms such as Adam correct most *but not all* divergent or vanishing scalings in the forward and backward pass, respectively. Of particular importance for evaluating the width-dependent signal propagation is the refined coordinate check (RCC) for disentangling effective updates in the current layer from updates propagating forward through the network. Ideally, all $W_0\Delta x_t^L$ and $\Delta W_t x_t$ should remain width-independent, which is only guaranteed in μP at sufficient width.

C Theoretical considerations

C.1 Distilled TP scaling arguments

Here we aim to provide a more detailed, comprehensive introduction to the essential width-scaling arguments inspired by Tensor Program (TP) theory.

Effective Updates. When training neural networks, we have control over the initial scaling W_0 and update scaling ΔW_t of trainable weights $W_t = W_0 + \Delta W_t$, but we are ultimately interested in their effect on the activations in the following layers. In standard architectures (including convolutional networks and Transformers), weights typically act linearly on the incoming activations. For such weights W_t and incoming activations x_t, we can decompose the next layer's (pre-)activation updates Δh_t into effective updates of W_t and activation updates Δx_t propagating forward from previous layers. Evaluating the contributions of both terms separately yields a *refined coordinate check*,

$$\Delta h_t = (\Delta W_t)x_t + W_0(\Delta x_t). \tag{RCC}$$

Note that updates of previous layers can propagate forward through the term $W_0\Delta x_t$ even when the current layer's effect on the output vanishes $\Delta W_t x_t \to 0$ as width $n \to \infty$. Hence, we say that the weight W_t is *effectively updated* only if $\Delta W_t x_t$ contributes non-vanishingly. Plotting the width-dependent scaling of $\|(\Delta W_t)x_t\|_{RMS}$ and $\|W_0(\Delta x_t)\|_{RMS}$ as a *refined coordinate check*, has been very useful for us to gain insights into the network internal signal propagation. The usefulness of (RCC) for effective update scalings is illustrated in Figure C.1. While the activations and activation updates in a Layernorm layer evolve width-independently when training GPT with Adam in SP and global learning rate scaled as $\eta_n = \eta \cdot n^{-1}$, the refined coordinate check reveals that the effective updates in the current (input-like) layer and the activation update scaling instead stems from effective updates propagating forward from previous (hidden-like) layers.

By choosing layerwise initialization variances and learning rates according to the Maximal Update Parameterization (μP), both terms in (RCC) become width-independent in all layers in each update step. Consequently, width-scaling becomes predictable, stable and feature learning is preserved even at large width. Starting from SP, μP can be realized with smaller last-layer initialization $\|W_0^{L+1}\|_{RMS} = O(n^{-1})$, larger input layer learning rate $\eta_{W^1} = \eta \cdot n$ and smaller last-layer learning rate $\eta_{W^{L+1}} = \eta \cdot n^{-1}$ for SGD.

Predicting scaling exponents. While the TP framework formally requires writing out all forward and backward pass computations performed during training and provides the exact infinite-width limit objects of output logits and activation coordinate distributions, we simplify its implications on width-scaling exponents for practical purposes as follows. A linear transformation either maps fixed to width-scaling dimension (*input-like*), width-scaling to width-scaling (*hidden-like*) or width-scaling to fixed dimension (*output-like*). Here, all bias vectors and normalization layer weights can be understood as input-like weights to the one-dimensional input 1. Any sum of length $n \to \infty$ that occurs in individual terms in (RCC) either accumulates a factor $n^{1/2}$ under sufficient independence of 0-mean summands (CLT-like behaviour) or a factor n when the summands are correlated or have non-zero mean (LLN-like behaviour). Crucially, not any sum may be evaluated with this heuristic but only weight and activation (update) pairs as in (RCC) (see Yang and Hu (2021, Appendix H)). If, for example, we considered the confounded term $(W_0 + \Delta W_t)x_0$, the initial part W_0x_0 clearly scales CLT-like but $\Delta W_t x_0$ scales LLN-like; evaluating the scaling of their sum might result in wrong scaling predictions.

At sufficient width, all width-scaling inner products $(\Delta W_t, x_t)$ from (RCC), however, are expected to behave LLN-like, that is $\|\Delta W_t x_t\|_{RMS} = \Theta(n \cdot \|\Delta W_t\|_{RMS} \cdot \|x_t\|_{RMS})$.

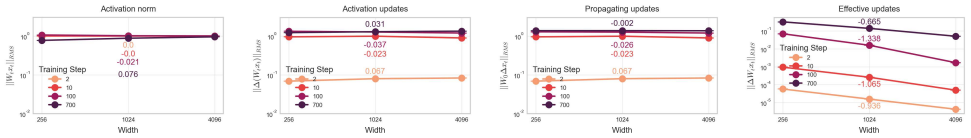


Figure C.1: **(Tracking effective updates requires refined coordinate check)** Activation norm $\|W_t x_t\|_{RMS}$, activation update norm $\|\Delta(W_t x_t)\|_{RMS}$, propagating update norm $\|W_0 \Delta x_t\|_{RMS}$ and effective update norm $\|\Delta W_t x_t\|_{RMS}$ for the last normalization layer in GPT trained with Adam and learning rate scaling $\eta_n = 0.01 \cdot n^{-1}$ for width-independent hidden layer feature learning. While activations and activation updates appear width-independent due to propagating updates, our refined coordinate check (RCC) reveals that Layernorm weight updates have vanishing effect in SP. Over time, effective updates accumulate effective rank, but do not lose alignment with width (Figure 2).

Concrete examples. Complementing the more generic introduction to TP scaling arguments above, we now provide more concrete examples for illustrating how weight updates affect activations in subsequent forward passes. Consider the Tensor Program for training MLPs with SGD from Yang and Hu (2021, Appendix H). We restate the relevant update scalings when using large learning rates in SP that induce output blowup. Since divergence is not allowed in the TP framework, it does not formally cover the unstable case, but we can still heuristically write down the scaling predictions, assuming that correlations still induce LLN-like exponents and independence still induces CLT-like exponents, as we have measured to hold empirically. The crucial insight is that training with cross-entropy loss effectively means that we are considering $f_t(\xi) = \sigma(W_t^{L+1} x_t^L(\xi))$ as the output function and the loss derivative also becomes $\chi_t := \frac{\partial \mathcal{L}_t}{\partial f} = \sigma(W_t^{L+1} x_t^L) - y_t$. Hence, from a stability point of view, we can allow $\tilde{f}_t := W^{L+1} x^L \to \infty$, which results in a saturated softmax. Under one-hot labels $y \in \{0,1\}^C$ with $\sum_c y_c = 1$, this means fast memorization of training points (x_i, y_i). For width-independent hidden-layer feature learning, we may still require activations to have width-independent coordinate-scaling, but let the output function be arbitrary, since the softmax renormalizes.

Definition C.1 (Activation stability). A parameterization is *activation stable* iff $\|x_t^l\|_{RMS} = \Theta(1)$ for all times $t \geq 0$ and all layers $l \in [L]$. ◀

We now show heuristically that MLPs trained with SGD in SP are activation stable and feature learning under global learning rate scaling $\eta = \Theta(n^{-1/2})$.

Backward pass. Here we denote the entry-wise scaling in width-scaling vectors as $v = \Theta(n^c)$, meaning $\|v\|_{RMS} = \Theta(n^c)$. Assuming $\|\phi'(h_t^l)\|_{RMS} = \Theta(1)$ as for ReLU (otherwise we would get vanishing gradients), the entries of the following width-scaling vectors scale as

$$\frac{\partial f}{\partial x_t^L} = \qquad W_t^{L+1} = W_0^{L+1} - \Delta W_t^{L+1} = O(n^{-1/2}), \qquad \frac{\partial f}{\partial h_t^l} = \frac{\partial f}{\partial x_t^l} \odot \phi'(h_t^l) = \Theta(\frac{\partial f}{\partial x_t^l}),$$

$$\frac{\partial f}{\partial x_t^{l-1}} = \quad (W_0^l)^\top \frac{\partial f}{\partial h_t^l} - \eta \theta_{W^l} \sum_{s=0}^{t-1} \chi_s \frac{(\frac{\partial f}{\partial h_s^l})^\top \frac{\partial f}{\partial h_s^l}}{n} x_s^{l-1} = \Theta(\max(\frac{\partial f}{\partial h_t^l}, \eta(\frac{\partial f}{\partial h_s^l})^2 x_s^{l-1}) = \Theta(n^{-1/2}).$$

Note that any larger learning rate scaling would induce exploding gradients. For example, $\eta = \Theta(1)$ induces $\delta W_1^{L+1} = \Theta(1)$, so $\frac{\partial f}{\partial x_1^L} = \Theta(1)$ and $\frac{\partial f}{\partial x_1^{L-k}} = \Theta(n \frac{\partial f}{\partial x_1^{L-k+1}}) = \Theta(n^{2k-1})$ for $k \geq 1$. This results in exploding activations in the next forward pass, and even larger gradients in the following backward pass.

We therefore continue with $\eta = \Theta(n^{-1/2})$, and get the activation updates

$$\delta h_t^1 = \qquad -\eta \chi_{t-1} \frac{\partial f}{\partial h_{t-1}^1} (\xi_{t-1})^\top \xi = \Theta(n^{-1/2} \cdot 1 \cdot n^{-1/2} \cdot 1) = \Theta(n^{-1}),$$

$$\delta h_t^l = \qquad W_{t-1}^l \delta x_t^{l-1} + \delta W_t^l x_t^{l-1}$$

$$= \theta_x \left(W_0^l \delta x_t^{l-1} - \eta \theta_{W^l} \sum_s \chi_{s-1} \underbrace{\frac{\partial f}{\partial h_{s-1}^l}}_{\Theta(n)^{-1/2}} \underbrace{(x_{t-1}^{l-1})^\top \delta x_t^{l-1}}_{O(n)} \right)$$
$$- \eta \chi_{t-1} \theta_W \underbrace{\frac{\partial f}{\partial h_{t-1}^l}}_{\Theta(n^{-1/2})} \underbrace{(x_{t-1}^{l-1})^\top x_t^{l-1}}_{\Theta(n)} = \Theta(1),$$

The output updates are

$$\begin{aligned}
\delta \tilde{f}_t(\xi) &= \delta W_t^{L+1} x_t^L(\xi) + (W_0^{L+1} + \Delta W_{t-1}^{L+1}) \delta x_t^L(\xi) \\
&= -\eta \chi_{t-1} \underbrace{x_{t-1}^L x_t^L(\xi)}_{\Theta(n)} + \underbrace{W_0^{L+1} \delta x_t^L(\xi)}_{\Theta(1)} + \underbrace{\Delta W_{t-1}^{L+1} \delta x_t^L(\xi)}_{\Theta(n^{1/2})} = \Theta(n^{1/2}), \\
\delta f_t(\xi) &= \sigma(\tilde{f}_{t-1} + \delta \tilde{f}_t) - \sigma(\tilde{f}_{t-1}) = \Theta(1).
\end{aligned}$$

2 layer networks. Observe that in 2 layer nets, there are no hidden layers, so that a larger learning rate can be chosen. Let $\eta = \Theta(n^c)$. Then in the first step, $\delta h_1^1 = \Theta(\eta \frac{\partial f}{\partial h_0^1}) = \Theta(n^c n^{-1/2})$. But note that the gradient scaling may grow after the first step, $\frac{\partial f}{\partial x_1^L} = W_1^{L+1} = \Theta(n^c)$, so that $\delta h_2^1 = \Theta(n^c n^c)$. Hence activation stability requires $\eta = O(1)$, which results in feature learning after 2 steps $\delta x_2^1 = \Theta(1)$. Then $\tilde{f}_t = \Theta(\eta (x_{t-1}^L)^\top x_t^L(\xi)) = \Theta(n)$.

Random feature models. In random feature models, we only train the last layer and keep all other weights fixed $W_t^l = W_0^l$ for all $l \leq L$. There, by definition, we do not get feature learning and the backward pass does not matter. The only gradient that matters is the last-layer gradient which has fixed scaling $\Theta(\chi_{t-1} x_{t-1}^L) = \Theta(1)$ at all times $t \geq 0$. The function update becomes $\delta W_t^{L+1} x^L(\xi) = -\eta \chi_{t-1} (x^L(\xi_{t-1}))^\top x^L(\chi) = \Theta(n^c n)$, where the inner product between activations converges to the NNGP kernel in the infinite-width limit. Hence large learning rates $\eta = \omega(n^{-1})$ result in immediate extreme memorization of the training points $f_t(\xi_{t-1}) \to$ one-hot(y_{t-1}) as $n \to \infty$, and $\eta_n = \Theta(n^{-1})$ results in fully width-independent dynamics.

Adam. Adam with small enough ε normalizes the gradients in each layer before updating the weights. Since the gradients $\nabla_{W^l} \mathcal{L} = \chi \frac{\partial f}{\partial h^l} (x^{l-1})^\top$ are generally correlated with the incoming activations x^{l-1}, their inner product accumulates $\Theta(\texttt{fan\_in})$. Non-vanishing correlation persists when only recovering the signs of the gradient. Hence for a width-independent effect on the output of the current layer, the learning rate should always be chosen as $\eta(W) = \frac{\eta}{\texttt{fan\_in}(W)}$. Since both hidden and output layers have $\texttt{fan\_in} = n$, activation stability requires a global learning rate $\eta = O(n^{-1})$, which results in effective hidden and output layer learning, but vanishing input layer updates. Networks recover input layer feature learning under $\eta = \Theta(1)$, where $\tilde{f}_t = \Theta(n)$. In random feature models, η just determines the extremeness of memorization of the training labels, where $\eta = \Theta(n^{-1})$ induces width-independence and $\eta = \omega(n^{-1})$ increasing memorization.

C.2 Measuring Alignment

Everett et al. (2024, Fig. 2) provides RMS-alignment exponents between weights W_t and incoming activations x_t. But only measuring the alignment between ΔW_t and x_t as well as W_0 and Δx_t from (RCC) separately allows to evaluate the width-scaling predictions from Yang and Hu (2021). For example hidden layers in μP scale as $(W_0^l)_{ij} = \Theta(n^{-1/2})$ at initialization, as 0-mean independence induces CLT-like scaling $W_0^l x_0^{l-1} = \Theta(n^{1/2} \cdot \|W_0^l\|_{RMS} \cdot \|x_0^{l-1}\|_{RMS})$. But updates are correlated with incoming activations, so that $\Delta W_t x_t = \Theta(n \cdot \|\Delta W_t\|_{RMS} \cdot \|x_t\|_{RMS})$ which necessitates $\|\Delta W_t\|_{RMS} = \Theta(n^{-1})$. This implies that the entry size of $W_t = W_0 + \Delta W_t$ is dominated by the initialization and confounds $\|W_t\|_{RMS}$ for accurately measuring the alignment exponent of the layer's updates ΔW_t. For correct width-scaling of the layer's learning rate, the influence of W_0 is irrelevant so that the joint alignment between W_t and x_t does not reveal the alignment exponent

that is relevant for correct learning rate scaling. Additionally, replacing the RMS-norm $\|A\|_{RMS}$ by the operator norm $\|\Delta W_t\|_{RMS\to RMS}$ provides a more natural measure of alignment (Bernstein and Newhouse, 2024), since the RMS-norm is confounded by accumulated rank whereas under maximal alignment for the operator norm it holds that $\|\Delta W_t x_t\|_{RMS} = \|\Delta W_t\|_{RMS\to RMS}\|x_t\|_{RMS}$, and the left-hand side is smaller under less alignment. Under perfect alignment we expect the ratio $\frac{\|\Delta W_t x_t\|_{RMS}}{\|\Delta W_t\|_{RMS\to RMS}\|x_t\|_{RMS}}$ to remain width-independent. We are not interested in constant prefactors, but only width-dependent scaling.

C.3 Formal statements and proofs of Propositions 1 and 2

Before providing the full formal statements of Proposition 1 and Proposition 2, we formally introduce all definitions and assumptions.

C.3.1 Definitions

In this section, we collect all definitions that do not appear in the main text. We adopt all definitions from Yang and Hu (2021), up to minor modifications. If not stated otherwise, limits are taken with respect to width $n \to \infty$.

Definition C.2 (Big-O Notation). Given a sequence of scalar random variables $c = \{c_n \in \mathbb{R}\}_{n=1}^{\infty}$, we write $c = \Theta(n^{-a})$ if there exist constants $A, B \geq 0$ such that for almost every instantiation of $c = \{c_n \in \mathbb{R}\}_{n=1}^{\infty}$, for n large enough, $An^{-a} \leq |c_n| \leq Bn^{-a}$. Given a sequence of random vectors $x = \{x_n \in \mathbb{R}^n\}_{n=1}^{\infty}$, we say x has coordinates of size $\Theta(n^{-a})$ and write $x = \Theta(n^{-a})$ to mean the scalar random variable sequence $\left\{\sqrt{\|x_n\|^2/n}\right\}_n$ is $\Theta(n^{-a})$. For the definition of $c = O(n^{-a})$ and $c = \Omega(n^{-a})$, adapt the above definition of $c = \Theta(n^{-a})$ by replacing $An^{-a} \leq |c_n| \leq Bn^{-a}$ with $|c_n| \leq Bn^{-a}$ and $An^{-a} \leq |c_n|$, respectively. We write $x_n = o(n^{-a})$ if $n^a \cdot \sqrt{\|x_n\|^2/n} \to 0$ almost surely. ◀

Definition C.3 (SGD update rule). Given a $(L+1)$-layer MLP with layerwise initialization variances $\{\sigma_l\}_{l\in[L+1]}$ and (potentially) layerwise learning rates $\{\eta_{W^l}\}_{l\in[L+1]}$, we define the *SGD update rule* as follows:

(a) Initialize weights iid as $(W_0^l)_{ij} \sim \mathcal{N}(0, \sigma_l^2)$.
(b) Update the weights via

$$W_{t+1}^l = W_t^l - \eta_{W^l} \cdot \nabla_{W^l} \mathcal{L}\left(f_t\left(\xi_t\right), y_t\right).$$

◀

Definition C.4 (Parameterization). We define a *width-scaling parameterization* as a collection of exponents $\{b_l\}_{l\in[L+1]} \cup \{c_l\}_{l\in[L+1]}$ that determine layerwise initialization variances $\sigma_l^2 = c_l \cdot n^{-b_l}$ and layerwise learning rates $\eta_l = \eta \cdot n^{-c_l}$, with width-independent constants $c_l, \eta > 0$ for all $l \in [L+1]$. ◀

Definition C.5 (Training routine). A *training routine* is a combination of base learning rate $\eta \geq 0$, training sequence $\{(\xi_t, y_t)\}_{t\in\mathbb{N}}$ and a continuously differentiable loss function $\mathcal{L}(f(\xi), y)$ using the SGD update rule. ◀

Definition C.6 (Stability). We say a parametrization of a $(L+1)$-layer MLP is *stable* if

1. For every nonzero input $\xi \in \mathbb{R}^{d_{in}}\backslash\{0\}$,

$$h_0^l, x_0^l = \Theta_\xi(1),\ \forall l \in [L], \quad \text{and} \quad \mathbb{E} f_0(\xi)^2 = O_\xi(1),$$

where the expectation is taken over the random initialization.
2. For any training routine, any time $t \in \mathbb{N}$, $l \in [L]$, $\xi \in \mathbb{R}^{d_{in}}$, we have

$$h_t^l(\xi) - h_0^l(\xi), x_t^l(\xi) - x_0^l(\xi) = O_*(1), \quad \text{and} \quad f_t(\xi) = O_*(1),$$

where the hidden constant in O_* can depend on the training routine, t, ξ, l and the initial function f_0.

◀

Definition C.7 (Nontriviality). We say a parametrization is *trivial* if for every training routine, $f_t(\xi) - f_0(\xi) \to 0$ almost surely for $n \to \infty$, for every time $t > 0$ and input $\xi \in \mathbb{R}^{d_{in}}$. Otherwise the parametrization is *nontrivial*. ◀

Definition C.8 (Feature learning). We say a parametrization *admits feature learning in the l-th layer* if there exists a training routine, a time $t > 0$ and input ξ such that $x_t^l(\xi) - x_0^l(\xi) = \Omega_*(1)$, where the constant may depend on the training routine, the time t, the input ξ and the initial function f_0 but not on the width n. ◀

Definition C.9 (σ-gelu). Define σ-gelu to be the function $x \mapsto \frac{x}{2}\left(1 + \operatorname{erf}\left(\sigma^{-1}x\right)\right) + \sigma\frac{e^{-\sigma^{-2}x^2}}{2\sqrt{\pi}}$. ◀

In order to apply the Tensor Program Master Theorem, all Nonlin and Moment operations in the NE⊗OR⊤ program (Yang and Littwin, 2023), which do not only contain parameters as inputs, are required to be pseudo-Lipschitz in all of their arguments. For training with SGD, this is fulfilled as soon as ϕ' is pseudo-Lipschitz. σ-`gelu` fulfills this assumption.

Definition C.10 (Pseudo-Lipschitz). A function $f : \mathbb{R}^k \to \mathbb{R}$ is called *pseudo-Lipschitz of degree d* if there exists a $C > 0$ such that $|f(x) - f(y)| \le C\|x - y\|(1 + \sum_{i=1}^k |x_i|^d + |y_i|^d)$. We say f is *pseudo-Lipschitz* if it is so for any degree d. ◀

C.3.2 Full formal statements of Propositions 1 and 2

Assumptions. For all of the results in this section, we assume that the used activation function is σ-`gelu` for $\sigma > 0$ sufficiently small. For small enough $\sigma > 0$, σ-`gelu` (Definition C.9) approximates ReLU arbitrarily well. We assume constant training time $t \ge 1$ as width $n \to \infty$. We assume batch size 1 for clarity, but our results can be extended without further complications to arbitrary fixed batch size.

Proposition C.11. ***(Asymptotic regimes in SP)*** *For fixed $L \ge 2$, $t \ge 1$, $\eta > 0$, $\alpha \in \mathbb{R}$, consider training a $(L+1)$-layer MLP of width n in SP with SGD and global learning rate $\eta_n = \eta \cdot n^{-\alpha}$ for t steps. Then the logits f_t, training loss $\mathcal{L}(f_t(\xi_t), y_t)$, loss-logit derivatives $\chi_t := \partial_f \mathcal{L}(f_t(\xi_t), y_t)$, loss-weight gradients $\nabla_t^l := \nabla_{W^l}\mathcal{L}(f_t(\xi_t), y_t)$ and activations x_t^l, $l \in [L]$, after training scale as follows in the infinite-width limit $n \to \infty$. The hidden constants in O_*, Ω_* and ω_* below can depend on the training routine, t, ξ, l and the initial function f_0.*

***Under cross-entropy (CE) loss**, three qualitatively distinct regimes arise:*

- *(a)* ***Stable regime*** *($\alpha \ge 1$): For any training routine, all $l \in [L]$ and any $\xi \in \mathbb{R}^{d_{in}}$, it holds that $\|f_t(\xi)\|_{RMS} = O_*(1)$, $|\mathcal{L}(f_t(\xi_t), y_t)| = O_*(1)$, $\|\chi_t\|_{RMS} = O_*(1)$, $\|\nabla_t^l\|_{RMS} = O_*(n^{-1/2})$ and $\|x_t^l(\xi)\|_{RMS} = O_*(1)$.*
- *(b)* ***Controlled divergence*** *($\frac{1}{2} \le \alpha < 1$): For any training routine, all $l \in [L]$ and any $\xi \in \mathbb{R}^{d_{in}}$, it holds that $\|n^{\alpha-1} \cdot f_t(\xi)\|_{RMS} = O_*(1)$, $\|x_t^l(\xi) - x_0^l(\xi)\|_{RMS} = O_*(1)$, $|\mathcal{L}(f_t(\xi_t), y_t)| = O_*(1)$, $\|\chi_t\|_{RMS} = O_*(1)$ and $\|\nabla_t^l\|_{RMS} = O_*(n^{-1/2})$. In addition, there exists a training routine and input ξ such that $\|n^{\alpha-1} \cdot f_t(\xi)\|_{RMS} = \Omega_*(1)$.*
- *(c)* ***Catastrophic instability*** *($\alpha < \frac{1}{2}$): For any $l \in [L]$, there exists a training routine and a $\xi \in \mathbb{R}^{d_{in}}$, such that $\|f_t(\xi)\|_{RMS} = \omega_*(1)$, $\|x_t^l(\xi)\|_{RMS} = \omega_*(1)$ and $\|\nabla_t^l\|_{RMS} = \omega_*(1)$.*

***Under mean-squared error (MSE) loss**, a stable regime as in (a) above arises if $\alpha \ge 1$. If $\alpha < 1$, training is catastrophically unstable as in (c) above and, in addition, there exists a training routine such that $|\mathcal{L}(f_t(\xi_t), y_t)| = \omega_*(1)$ and $\|\chi_t\|_{RMS} = \omega_*(1)$.*

Proposition C.12 (Under CE loss, SP with large learning rates learns features at large width). *Consider the setting of Proposition 1 of training a (L+1)-layer MLP with SGD in SP with global learning rate $\eta_n = \eta \cdot n^{-\alpha}$, $\alpha \in \mathbb{R}$, in the infinite-width limit $n \to \infty$. The hidden constants in O_*, Ω_* and ω_* below can depend on the training routine, t, ξ, l and the initial function f_0.*

- *(a) Under both MSE and CE loss in the stable regime ($\alpha \ge 1$), for any training routine, $l \in [L]$ and $\xi \in \mathbb{R}^{d_{in}}$ it holds that $\|\Delta x_t^l(\xi)\|_{RMS} = O_*(n^{-1/2})$.*
- *(b) Under CE loss in the controlled divergence regime ($\frac{1}{2} \le \alpha < 1$), for any training routine, $l \in [L]$ and $\xi \in \mathbb{R}^{d_{in}}$ it holds that $\|\Delta x_t^1(\xi)\|_{RMS} = O_*\left(n^{-1/2-\alpha}\right)$, and $\|\Delta x_t^l(\xi)\|_{RMS} = O_*\left(n^{1/2-\alpha}\right)$. For any $l \in [L]$, there exists a training routine and $\xi \in \mathbb{R}^{d_{in}}$ such that $\|\Delta x_t^1(\xi)\|_{RMS} = \Omega_*\left(n^{-1/2-\alpha}\right)$, and $\|\Delta x_t^l(\xi)\|_{RMS} = \Omega_*\left(n^{1/2-\alpha}\right)$.*

Remark C.13 (2-layer networks recover stable training dynamics and width-independent feature learning at $\alpha = 0$). Similarly, it can be shown that 2-layer MLPs remain activation stable under width-independent learning rate scaling $\eta_n = \Theta(1)$. The controlled divergence regime is given by $0 \leq \alpha < 1/2$, with width-independent input layer feature learning at $\alpha = 0$. ◀

Remark C.14 (Adam recovers stable training dynamics and width-independent hidden-layer feature learning at $\alpha = 1$). For Adam and $L \geq 2$, an analogous NE⊗OR⊤-based proof (Yang and Littwin, 2023) would show that $\eta_n = \Theta(n^{-1})$ recovers feature learning in all hidden layers $l \in [2, L]$, stable activations and loss-logit gradients, while logits blow up only through $W_0^{L+1}\Delta x_t^L = \Theta(n^{1/2})$. To avoid logit blowup, $\eta_n = \Theta(n^{-3/2})$ would be necessary. In that case, only the term $W_0^{L+1}\Delta x_t^L$ would contribute non-vanishingly to the logit updates. Hence, for Adam under CE loss, the controlled divergence regime is given by $1 \leq \alpha < 3/2$, with hidden-layer feature learning at $\alpha = 1$. ◀

C.3.3 Proof of Propositions 1 and 2

The proof in Yang and Hu (2021) for general stable abc-parameterizations directly covers the stable regimes of both losses, showing a kernel regime and vanishing feature learning for $\alpha \geq 1$.

For the controlled divergence regime under CE loss, however, note that the TP framework does not allow diverging computations. Here, we need to replace the logit updates by rescaled logit updates, before computing the softmax limit outside of the TP framework.

Formally, under standard initialization, $W_0^{L+1} \sim N(0, 1/n)$ is replaced in the TP by $\hat{W}_\varepsilon^{L+1}$ constructed via Nonlin, conditioning on $f_0(\xi)$ (see Yang and Hu (2021, Appendix H) for all details). For stable parameterizations, the function updates are defined in the TP as

$$\delta \hat{f}_t = \theta'_{L+1} \frac{\delta W_t^{L+1} x_t^L}{n} + \theta'_{Lf} \frac{\hat{W}_{t-1}^{L+1} \delta x_t^L}{n},$$

where $\theta'_{L+1} = n^{1-\alpha}$ and $\theta'_{Lf} = n^{1-r-b_{L+1}}$. In the controlled divergence regime $\alpha < 1$, we now define rescaled logit updates in the TP as

$$\delta \hat{f}_t = \hat{\theta}_{L+1} \frac{\delta W_t^{L+1} x_t^L}{n} + \hat{\theta}_{Lf} \frac{\hat{W}_{t-1}^{L+1} \delta x_t^L}{n},$$

by replacing θ'_{L+1} by $\hat{\theta}_{L+1} := \theta_\alpha \theta'_{L+1}$ and replacing θ'_{Lf} by $\hat{\theta}_{Lf} := \theta_\alpha \theta'_{Lf}$, where $\theta_\alpha := n^{\alpha-1}$. The adapted pre-factors ensure that $\delta \hat{f}$ remains $O_*(1)$ for a well-defined TP. The TP master theorem now implies almost sure convergence of the rescaled logit updates $\delta \hat{f}_t \to \mathring{\delta} \hat{f}_t \in \mathbb{R}^{d_{\text{out}}}$ a.s.

Now we compute the softmax limit outside of the TP framework, as we want to recover the softmax values of the original diverging logits. Thus, given the convergent sequence $\delta \hat{f}_t \to \mathring{\delta} \hat{f}_t \in \mathbb{R}^{d_{\text{out}}}$ a.s., due to the smoothness and saturation properties of the softmax it follows that there exists a $\mathring{\chi}_t \in \mathbb{R}^{d_{\text{out}}}$ such that $\sigma(\theta_\alpha^{-1} \cdot \delta \hat{f}_t) - y_t \to \mathring{\chi}_t$ a.s. Since $|\sigma(\theta_\alpha^{-1} \cdot \delta \hat{f}_t) - y_t| \leq 1 + |y_t|$ and $|\mathring{\chi}_t| \leq 1 + |y_t|$, this sequence can again be used as a TP scalar. Now the last-layer weights are TP vectors updated with $\delta W_t^{L+1} = -\eta_n \chi_t x_t^L$ which do not change the scaling of $\hat{W}_t^{L+1} = \hat{W}_{t-1}^{L+1} + \theta_{L+1/f} \cdot \delta W_1^{L+1}$ with $\theta_{L+1/f} \leq 1$ as long as $\alpha \geq 1/2$. Thus the backward pass scalings are not affected and the rest of the TP can remain unchanged.

For larger learning rates $\alpha < 1/2$ under CE loss, we provide heuristic scaling arguments. Observe that preactivations diverge after the first update step $\delta h_1^2 = -\eta_n \frac{\partial f_0}{\partial h^2}(x_0^1)^\top x_1^1 = \Theta(n^{1/2-\alpha})$. The updates of the next hidden layer's preactivations scale even larger, that is $\delta h_1^3 = -\eta_n \frac{\partial f_0}{\partial h^3}(x_0^2)^\top x_1^2 = \Theta(n^{2(1/2-\alpha)})$. In this way, the exponent growth continues throughout the forward pass. But even if there is only a single hidden layer, the scaling of the backpropagated gradient is increased after the second step, $\frac{\partial f_2}{\partial x^L} = W_0 - \eta_n \chi_0 x_0^L - \eta_n \chi_1 x_1^L = \Omega(\eta_n \chi_1 \delta x_1^L) = \Omega(n^{1/2-2\alpha}) = \omega(n^{-1/2})$. This, in turn, increases the preactivation update scaling $\delta h_3^2 = -\eta_n \frac{\partial f_2}{\partial h^2}(x_2^1)^\top x_3^1 = \Omega(n^{-\alpha} \frac{\partial f_2}{\partial x^L} n) = \Omega(n^{3(1/2-\alpha)})$, which in turn increases the gradient scaling in the next step, inducing a feedback loop of cascading exponents between diverging activations and gradients, inducing fast training divergence.

Under MSELoss, observe how, already for $\alpha < 1$, diverging logits $\delta f_1 = W_0^{L+1} \delta x_1^L - \eta_n \chi_0 (x_0^L)^\top x_0^L = \Theta(n^{1-\alpha})$ increase the gradient scaling through $\chi_1 = f_1 - y_1 = \Theta(n^{1-\alpha})$ which in

turn increases the activation as well as logit scaling in the next step, and induces a divergent feedback loop even worse than above.

C.4 Scaling dynamics in 2-layer linear networks

Here, we rederive the training dynamics of the minimal model from Lewkowycz et al. (2020) that shows an initial catapult mechanism in NTP. They observe that the training dynamics of repeatedly updating a 2-layer linear network in NTP on the same training point is fully captured by update equations of the current function output t_t and the current sharpness λ_t.

C.4.1 Deriving the update equations for SP, NTP and μP

NTP. The original model by Lewkowycz et al. (2020) is given by

$$f = n^{-1/2} vux,$$

where $u \in \mathbb{R}^{n\times d}, v \in \mathbb{R}^n$ are initialized as $u_{ij}, v_i \sim N(0,1)$ and trained with MSE loss $L(f,x,y) = \frac{1}{2}(f(x)-y)^2$, loss derivative $\chi_t = f(x) - y$ and a global learning rate η.

Repeated gradient descent updates using (x,y), then results in the update equations,

$$\begin{aligned} f_{t+1} = \quad & f_t(1 + n^{-1}\eta^2\chi^2\|x\|^2) - \eta\chi_t\lambda_t, \\ \lambda_{t+1} = \quad & \lambda_t + n^{-1}\eta\chi_t\|x\|^2(\eta\chi_t\|x\|^2\lambda_t - 4f_t), \end{aligned}$$

where the *update kernel* is defined as

$$\tilde{\Theta}(x,x') = n^{-1}(\|u\|^2 + \|v\|^2).$$

Note that the width-dependence in f_t and λ_t results in qualitatively different behaviour in the infinite-width limit. In particular, in the limit, the sharpness cannot evolve over the course of training, $\lambda_t = \lambda_0$.

Maximal Update Parameterization. We define a *2-layer linear network in μP with arbitrary weight multipliers* as

$$f = \bar{v}\bar{u}x,$$

with *reparameterization-invariant weights* $\bar{u}_{ij} \sim N(0, 1/d_{in})$ and $\bar{v}_i \sim N(0, 1/n^2)$, $\bar{u} = n^{-a_u}u, \bar{v} = n^{-a_v}v$, and the *original weights* u, v are trained with MSE loss and layerwise learning rates $\eta_u = \eta n^{1+2a_u}$ and $\eta_v = \eta n^{-1+2a_v}$, which results in reparameterization-invariant layerwise learning rates $\bar{\eta}_u = \eta n$ and $\bar{\eta}_v = \eta n^{-1}$.

Formally, we now perform updates on u and v, but we can work with $\bar{u}$ and $\bar{v}$ instead. For gradients it holds that $\frac{\partial f}{\partial u} = \frac{\partial f}{\partial \bar{u}}\frac{\partial \bar{u}}{\partial u} = \frac{\partial f}{\partial \bar{u}} n^{-a_u}$; this width scaling has to be accounted for when transitioning between representatives of the μP equivalence class. For updates $\bar{\eta}_u, \bar{\eta}_v$ should be used instead of η_u, η_v, as the layerwise learning rate rescaling was exactly chosen to cancel out the effect of the weight rescaling,

$$\bar{u}_{t+1} - \bar{u}_t = -n^{-a_u}\eta_u \frac{\partial f}{\partial \bar{u}}\frac{\partial \bar{u}}{\partial u} = -n^{-2a_u}\eta_u \frac{\partial f}{\partial \bar{u}} = -\bar{\eta}_u \frac{\partial f}{\partial \bar{u}}.$$

The derivatives for backpropagation are given by,

$$\chi_t := \frac{\partial L}{\partial f} = f(x_t) - y_t, \qquad \frac{\partial f}{\partial \bar{v}} = x^\top \bar{u}^\top, \qquad \frac{\partial f}{\partial \bar{u}} = \bar{v}^\top x^\top.$$

The updated weights are then given by

$$\bar{v}_{t+1} = \bar{v}_t - \eta n^{-1}\chi_t x^\top \bar{u}^\top, \qquad \bar{u}_{t+1} = \bar{u}_t - \eta n \chi_t \bar{v}^\top x^\top.$$

In the case $d_{in} = 1$, the updated function is then given by

$$\begin{aligned} f_{t+1} = \quad & \bar{v}_{t+1}\bar{u}_{t+1}x = f_t + \eta^2\chi_t^2 x^\top \bar{u}^\top \bar{v}^\top x^\top x - \bar{\eta}_u\chi_t \bar{v}\bar{v}^\top x^\top x - \bar{\eta}_v\chi_t x^\top \bar{u}^\top \bar{u}x \\ = \quad & f_t\big(1 + \eta^2\chi_t^2\|x\|^2\big) - \eta\chi_t\big(n\|\bar{v}\|^2 + n^{-1}\|\bar{u}\|^2\big)\|x\|^2 \end{aligned}$$

$$= f_t\big(1+\eta^2\chi_t^2\|x\|^2\big) - \eta\chi_t\tilde\Theta(x,x),$$

where we call $\tilde\Theta$ the reparameterization-invariant *update kernel* defined as

$$\tilde\Theta(x,x') = \sum_l \frac{\eta_l}{\eta}\frac{\partial f(x)}{\partial W^l}\frac{\partial f(x')}{\partial W^l} = x^\top\big(n^{-1}\|\bar u\|^2 + n\|\bar v\|^2\big)x'.$$

The update kernel evolves via the reparameterization-invariant update equation

$$\begin{aligned}
\lambda_{t+1} &= \tilde\Theta_{t+1}(x,x) = \|x\|^2\big(n\|\bar v_{t+1}\|^2 + n^{-1}\|\bar u_{t+1}\|^2\big)\\
&= \|x\|^2\Big(n\|\bar v_t\|^2 + n^{-1}\|\bar u_t\|^2 + n^{-1}\bar\eta_u^2\chi_t^2\bar v\bar v^\top x^\top x\\
&\qquad -2n\bar\eta_v\chi_t\bar v\bar u x + n\bar\eta_v^2\chi_t^2 x^\top\bar u^\top\bar u x - 2n^{-1}\bar\eta_u\chi_t\bar v\bar u x\Big)\\
&= \lambda_t + \|x\|^2\big(\eta^2\chi_t^2\|x\|^2(n^{-1}\|\bar u\|^2 + n\|\bar v\|^2) - 4\eta\chi_t f_t\big)\\
&= \lambda_t + \|x\|^2\Big(\eta^2\chi_t^2\lambda_t - 4\eta\chi_t f_t\Big)\\
&= \lambda_t + \|x\|^2\eta\chi_t\Big(\eta\chi_t\lambda_t - 4f_t\Big).
\end{aligned}$$

Now note that, even under $f_0 = 0$, we get non-trivial, width-independent dynamics. Due to the LLN, at initialization, we have $n^{-1}\|\bar u_0\|^2 \approx 1$ and $n\|\bar v_0\|^2 \approx 1$ (=n^{-1} times sum over n iid χ^2 variables), hence $\lambda_0 \approx 2$.

To conclude, the training dynamics for repeatedly updating with the same training point (x, y) are fully described by the update equations,

$$f_{t+1} = f_t\big(1+\eta^2\chi_t^2\|x\|^2\big) - \eta\chi_t\lambda_t, \tag{C.1}$$

$$\lambda_{t+1} = \lambda_t + \|x\|^2\eta\chi_t\Big(\eta\chi_t\lambda_t - 4f_t\Big). \tag{C.2}$$

This can be rewritten in terms of the error (or function-loss derivative under MSE loss) $\chi_t = f_t - y$, akin to Kalra et al. (2025), as

$$\chi_{t+1} = \chi_t\big(1 - \eta\lambda_t + \eta^2\|x\|^2\chi_t(\chi_t+y)\big), \tag{C.3}$$

$$\lambda_{t+1} = \lambda_t + \|x\|^2\eta\chi_t\Big(\eta\chi_t\lambda_t - 4(\chi_t+y)\Big). \tag{C.4}$$

First observe that all terms in the update equations become width-independent in μP. Only the initial conditions are width-dependent with vanishing variance, $f_0 = \Theta(n^{-1/2})$. As opposed to NTP, the sharpness update term $\eta^2\chi_t^2\|x\|^2$ is not vanishing anymore. While Lewkowycz et al. (2020) simply use labels $y = 0$, non-trivial dynamics in μP require $y \neq f_0 \to 0$.

Importantly, $\eta\chi_t$ always appear jointly, so that interpolation effectively reduces the learning rate.

Remark C.15 (Characterizing sharpness increase: Critical threshold depends on the labels.). When both sharpness and the loss increase, then training diverges as the learning rate lies even further from its edge of stability. In μP, since $f_0 \to 0$, λ_t will grow in the first step. For subsequent steps, the sharpness update equation (C.4) implies that sharpness increases ($\lambda_{t+1} \geq \lambda_t$) if and only if $\lambda_t \geq \frac{4}{\eta}(1+\frac{y}{\chi}) = \frac{4}{\eta}\frac{f_t}{\chi_t}$. Kalra et al. (2025) provides a more extensive analysis of the dynamics and fixed points of this model in μP. ◀

Remark C.16 (Weight multipliers). A natural choice of weight multipliers for μP can be considered to be $a_l = 1/2\cdot\mathbb{I}(l = L+1) - 1/2\cdot\mathbb{I}(l=1)$, as this choice allows using a width-independent global learning rate $\eta_n = \eta\cdot n^0$, and the update kernel does not require width-dependent scaling factors, $\tilde\Theta(x,x') = x^\top\big(\|u\|^2 + \|v\|^2\big)x'$. In other words, under these weight multipliers, width-independence in parameter space translates into width-independence in function space. ◀

Standard Parameterization. We define training a 2-layer linear network in SP with global learning rate scaling n^{-c} as

$$f = \bar v\bar u x,$$

with initialization $\bar{u} \sim N(0, 1/d_{in}), \bar{v} \sim N(0, 1/n)$ and global learning rate $\bar{\eta}_u = \bar{\eta}_v = \eta n^{-c}$. Parameter multipliers affect all scalings in the same way as for μP. Only the learning rate has a different prefactor, and the last layer has larger initialization. The adapted update equations become

$$\begin{aligned} f_{t+1} = & \quad f_t(1 + n^{-2c}\eta^2\chi^2\|x\|^2) - n^{1-c}\eta\chi_t\lambda_t, \\ \lambda_{t+1} = & \quad \lambda_t + \|x\|^2 n^{-c}\eta\chi_t(n^{-c}\eta\chi_t\lambda_t - 4n^{-1}f_t), \end{aligned}$$

where we define, as for NTP,

$$\tilde{\Theta}(x, x') = n^{-1}(\|\bar{u}\|^2 + \|\bar{v}\|^2),$$

where $n^{-1}\|\bar{v}\|^2 \approx n^{-1}$ at initialization (n^{-2} times sum over n iid χ^2-variables with positive mean).

Choosing $c < 1$ results in output blowup of the term $n^{1-c}\eta\chi_t\lambda_t$. While this can in principle be counteracted by shrinking λ_t at finite width, a well-defined stable and non-trivial infinite-width limit is only attained at $c = 1$, where $f_{t+1} = f_t - \eta\chi_t\lambda_t$ and $\lambda_{t+1} = \lambda_t$. We now show that also at finite width, stable training with a constant learning rate in SP requires $\eta = O(n^{-1})$.

C.4.2 Finding the maximal stable learning rate scaling by characterizing the conditions for loss and sharpness decrease

The following proposition characterizes the choices of η that result in a decrease in loss at any present state.

Writing $n_{sp} = \begin{cases} n, & \text{in SP,} \\ 1, & \text{else,} \end{cases}$ $n_{ntp} = \begin{cases} n, & \text{in NTP,} \\ 1, & \text{else,} \end{cases}$, we can write the update equations of parameterizations jointly as

$$\begin{aligned} \chi_{t+1} = & \quad \chi_t(1 - n_{sp}\eta\lambda_t + n_{ntp}^{-1}\eta^2\chi\|x\|^2(\chi_t + y)), \\ \lambda_{t+1} = & \quad \lambda_t + \eta\chi_t\|x\|^2 n_{ntp}^{-1}(\eta\chi_t\lambda_t - 4n_{sp}^{-1}(\chi_t + y)). \end{aligned}$$

Proposition C.17 (Characterizing loss decrease in SP and NTP). *Let $\eta \geq 0$. For n_{sp} or n_{ntp} large enough, we write the update equations of repeatedly updating the uv-model with SGD on the training point (x, y) with $\|x\| = 1$ in SP or NTP jointly as provided above. The loss decreases at any step, omitting time t,*

1. *in the case $f(f - y) \geq 0$, if and only if $\eta \leq \frac{2}{n_{sp}\lambda} + O(n_{sp}^{-3}n_{ntp}^{-1})$ or $\eta \in [\frac{n_{sp}n_{ntp}\lambda}{\|x\|^2\chi f} - \frac{2}{n_{sp}\lambda} - O(n_{sp}^{-3}n_{ntp}^{-1}), \frac{n_{sp}n_{ntp}\lambda}{\|x\|^2\chi f}]$,*
2. *in the case, $f(f - y) < 0$, if and only if $\eta \leq \frac{2}{n_{sp}\lambda} - O(n_{sp}^{-3}n_{ntp}^{-1})$.*

It holds that $\lambda_{t+1} \geq \lambda_t$ if and only if $\lambda_t \geq \frac{4}{n_{sp}\eta_t}(1 + \frac{y}{\chi_t})$.

Remark C.18 (Instability in SP). The crucial insight from Proposition C.17 for SP is that both loss and sharpness increase early in training as soon as $\eta = \omega(n^{-1})$, unless an extensively large learning rate that depends on the current sharpness, training point and output function, is accurately chosen at each time step in a slim interval of benign large learning rates, which is unlikely to hold in practice. Figure C.2 shows that in simulated training with constant learning rates, the maximal stable learning rate indeed scales as $\Theta(n^{-1})$. This instability prediction is in line with the infinite-width prediction from Yang and Hu (2021), and hence does not explain large learning rate stability in SP in practice. ◀

Proof. From the update equations, it holds that $\lambda_{t+1} \geq \lambda_t$ if and only if $\eta_t n_{ntp}\chi_t(\eta_t\chi_t\lambda_t - \frac{4}{n_{sp}}f_t) \geq 0$ if and only if $\lambda_t \geq \frac{4}{n_{sp}\eta_t}(1 + \frac{y}{\chi_t})$.

Observe that the loss decreases if and only if $|\chi_{t+1}| \leq |\chi_t|$, which holds if and only if $|1 - n_{sp}\eta\lambda_t + \eta^2 n_{ntp}^{-1}\chi\|x\|^2(\chi_t + y)| \leq 1$, which can be written as, omitting all subscripts $\cdot_t$,

$$\eta^2 n_{ntp}^{-1}\|x\|^2\chi f - \eta n_{sp}\lambda \in [-2, 0].$$

Assuming $\eta \geq 0$, the above holds if and only if $\big(\eta\chi f \leq \frac{n_{sp}n_{ntp}\lambda}{\|x\|^2}$ and $n_{ntp}^{-1}\eta^2\|x\|^2\chi f - \eta n_{sp}\lambda \geq -2\big)$. The first constraint is a mild one that states $\eta = O(n)$. We will now focus on the second one.

Solving for the roots of this polynomial in η, we get

$$\eta_{1,2} = \frac{1}{2\|x\|^2\chi f}\left(n_{ntp}n_{sp}\lambda \pm \sqrt{n_{ntp}^2 n_{sp}^2\lambda^2 - 8\|x\|^2 n_{ntp}\chi f}\right).$$

Assuming $n_{sp}^2 n_{ntp}\lambda^2 \gg 8\|x\|^2\chi f =: C$, we get $n_{sp}n_{ntp}\lambda\sqrt{1-\frac{C}{n_{sp}^2 n_{ntp}\lambda^2}} \approx n_{sp}n_{ntp}\lambda(1 - \frac{C}{2n_{sp}^2 n_{ntp}\lambda^2} - \frac{1}{4}(\frac{C}{n_{sp}^2 n_{ntp}\lambda^2})^2)$. In that case $\eta_1 \approx \frac{2}{n_{sp}\lambda}$ and $\eta_2 \approx \frac{n_{sp}n_{ntp}\lambda}{\|x\|^2\chi f} - \frac{2}{n_{sp}\lambda}$.

Hence, if $\chi f \geq 0$, we get loss decrease if $\eta \leq \frac{2}{n_{sp}\lambda} + O(n_{sp}^{-3}n_{ntp}^{-1})$ or $\eta \in [\frac{n_{sp}n_{ntp}\lambda}{\|x\|^2\chi f} - \frac{2}{n_{sp}\lambda} - O(n_{sp}^{-3}n_{ntp}^{-1}), \frac{n_{sp}n_{ntp}\lambda}{\|x\|^2\chi f}]$.

If $\chi f < 0$, we get loss decrease if $\eta \in [\frac{n_{sp}n_{ntp}\lambda}{\|x\|^2\chi f} - \frac{2}{n_{sp}\lambda} + O(n_{sp}^{-3}n_{ntp}^{-1}), \frac{2}{n_{sp}\lambda} - O(n_{sp}^{-3}n_{ntp}^{-1})]$, where the left end of the interval is negative. The upper end resembles the edge of stability that vanishes as n_{sp}^{-1} for SP but not for NTP.

□

Note the interesting slim regime of benign large learning rates $\eta \approx \frac{n\lambda}{\|x\|^2\chi f} - \frac{1}{n_{sp}\lambda} = \Theta(n)$ when $f(f-y) > 0$. As all of the involved quantities are known at training time, an adaptive learning rate schedule may significantly speed up training by stable learning with excessive learning rates. However it remains unclear whether a similar regime exists in practical architectures under CE loss. In that case, the sharpness computation would also be much more computationally expensive.

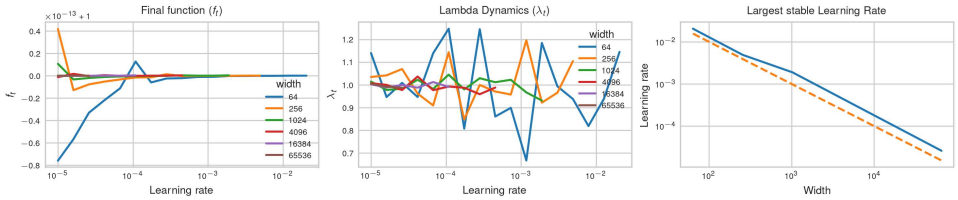


Figure C.2: **Stable SP.** f_t (left) and λ_t (center) after training to convergence for several widths. The largest stable learning rate for SP indeed scales as n^{-1} (*right*). When lines end, training diverged for larger learning rates. The first subplot shows that training has succeeded in memorizing the training label $y = 1$ at the optimal learning rate at all widths. The second subplot shows that the randomness in λ_t due to random initial conditions vanishes with increasing width, as SP is approaching its kernel regime.

C.4.3 μP converges faster to its limit than SP and NTP

Here we study the convergence speed of the uv-model from Appendix C.4 to the infinite-width limit in SP, NTP and μP through simulations in the gradient flow regime $\eta \in \{0.001, 0.01\}$. We draw new data points without signal $x, y \sim N(0, I_2)$ for every update. Convergence of the function to $y = 1$ would confound the findings. If we draw the initial $\mathring{f}_0$ from $N(0,1)$ for SP and NTP versus $N(0, n^{-1})$ for μP independent of f_0 at finite width, we only see convergence $f_t \to \mathring{f}_t$ for μP due to non-vanishing variance in the initial output function in SP and NTP (Figure C.6). Therefore, in all following experiments in this section, we start the update equations of the limit from the same f_0 to just measure the deviations over training. We let λ_0 differ from $\mathring{\lambda}_0$, as otherwise μP at finite width would exactly coincide with its limit, as it follows width-independent update equations. Observe that μP still converges faster to its limit than NTP, which converges faster than SP (Figure C.5). The convergence exponent of μP seems to lie around -0.45. Observe large variance of the difference to the limit between random initial seeds (Figures C.3 and C.4). This requires many runs for an accurate estimation of the exponent. Note that the slow convergence of SP here may be due to the kernel regime enforced by MSE loss, and convergence to our feature learning limit under $\eta_n = \Theta(n^{-1/2})$ may be much faster.

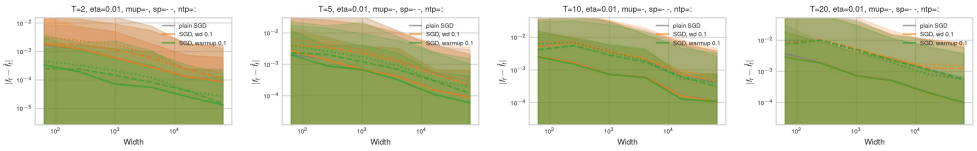


Figure C.3: **(μP remains closest to its limit across training)** Difference between finite networks and their infinite-width limit from the same initial condition across 100 random seeds for μP (line), SP (dotted line) and NTP (dashed line) after $T = 2$, 5 or 10 steps (left to right), running plain SGD in gray, 0.1 warmup in green and 0.1 weight decay in orange. Initially, warmup retains the most closeness to the limit, as the learning rate is very small. In later steps, μP clearly remains closer to its limit. All parameterizations converge to their limit, but with large variance.

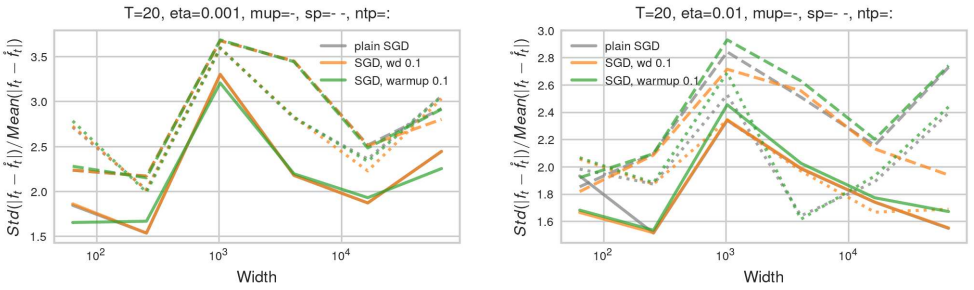


Figure C.4: **(Large variance in convergence to limit)** Standard deviation divided by mean of $|f_t - \mathring{f}_t|$ across random initialization. Note that in all paramerizations the variance of the difference to the limit is very large.

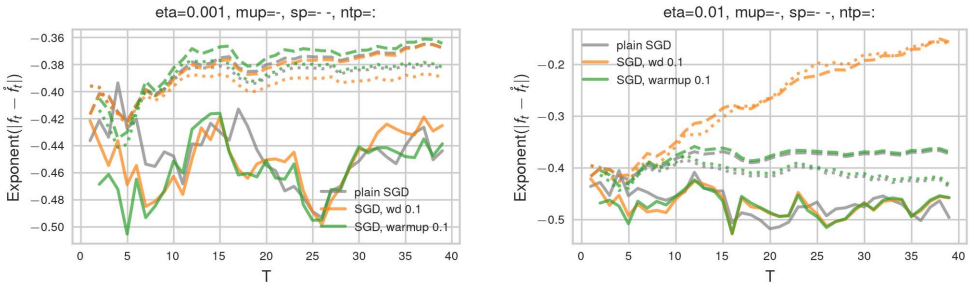


Figure C.5: **(μP converges faster to its limit)** Width convergence exponent of the decay of the difference of learned finite neural networks to their limit derived between smallest and largest width 64 and 65536 for learning rate 0.001 (left) and 0.01 (right) across 100 random seeds for μP (line), SP (dotted line) and NTP (dashed line) against number of training iterations T, running plain SGD in gray, 0.1 warmup in green and 0.1 weight decay in orange. While the exponents are still noisy even from means of 100 random seeds, μP clearly converges faster to its limit than NTP which converges faster than SP, even when starting from the same initial conditions. SP and NTP with weight decay seem to systematically deviate from their limit only at large learning rate and late in training as their exponent decreases with the amount of training.

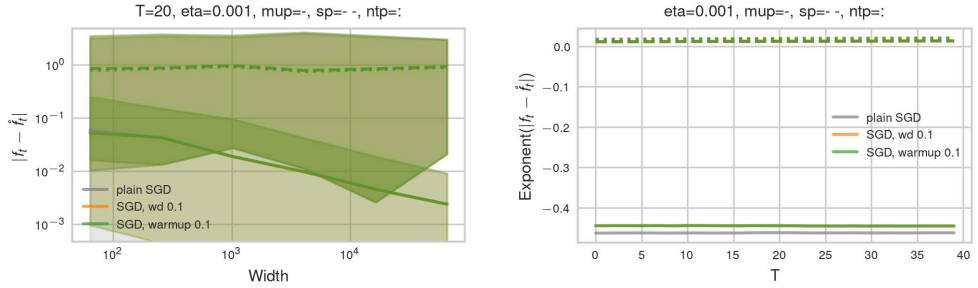


Figure C.6: **(Non-vanishing initial variance in SP and NTP prevent convergence)** Difference to limit after 20 steps (left) and corresponding exponents (right) for differing initial $\mathring{f}_0$ drawn independently from f_0. Only μP converges to its limit at exponent around -0.45.

D Experimental details

We will make PyTorch code to reproduce all of our experiments publicly available upon acceptance.

If not otherwise specified, we train a single epoch to prevent confounding from multi-epoch overfitting effects.

D.1 MLPs

We implement our MLP experiments on MNIST (Deng, 2012) and CIFAR-10 (Krizhevsky et al., 2009) in PyTorch (Paszke et al., 2019). We train ReLU MLPs with the same width n in all hidden dimensions with plain SGD/Adam with a single learning rate for all trainable parameters, batch size 64 without learning rate schedules, weight decay or momentum to prevent confounding. We use Adam with the PyTorch standard hyperparameters. By standard initialization we mean He initialization variance $c_\phi/\texttt{fan\_in}$ with $c_\phi = 2$ for the ReLU activation function (He et al., 2015).

D.2 Multi-index data

We generate multi-index teacher data, inspired by Kunin et al. (2024), but setting a deterministic teacher for ensuring a balanced classification task.

We draw the covariates $\xi \sim \mathcal{U}(\mathbb{S}^{d_{in}-1})$ i.i.d. from the uniform distribution on the unit sphere in $\mathbb{R}^{d_{in}}$ with input dimension $d_{in} = 100$. The training set consists of 10^3 training points. We also draw a test set consisting of 10^4 test points.

For the target function f^*, drawing 3 random directions as in Kunin et al. (2024) results in heavily unbalanced classes and $f^* = 0$ on large part of the support with high probability. Instead, we set 4 teacher neurons deterministically for less noisy results. The teacher net is a shallow ReLU network given by $f^*(\xi) = \text{sign}(\sum_{i=1}^4 s_i \phi(w_i^\top \xi))$ with unit vectors $w_1 = e_1$, $w_2 = e_2$, $w_3 = -e_1$, $w_4 = -e_2$ and signs $s_1 = s_3 = +1$ and $s_2 = s_4 = -1$. This results in the nonlinear target function $f^*(\xi) = \text{sign}(\xi_1 - \xi_2)$ for all $\xi \in \mathbb{R}^{d_{in}}$ with $\xi_1 > 0$ or $\xi_2 > 0$, but $f^*(\xi) = \text{sign}(\xi_2 - \xi_1)$ for all $\xi \in (-\infty, 0) \times (-\infty, 0)$. We do not use label noise.

This dataset requires learning to attend to the first 2 covariate dimensions (ξ_1, ξ_2), where all of the signal for the labels $f^*(\xi)$ lies. If the input layer does not learn to align with these dimensions, the sparse signal is obscured in the activations (random features) after the first layer due to the large variance in the remaining covariate dimensions.

D.3 Language modeling

We train small Transformer models (Vaswani et al., 2017) using LitGPT (Lightning AI, 2023). We adapt the Pythia (Biderman et al., 2023) architecture with 6 Transformer blocks, standard $\texttt{d\_head}^{-1/2}$ attention scaling, pre-attention and qk-Layernorm (Wortsman et al., 2024). We purely scale width, proportionally scaling the number of attention heads and the MLP hidden size while keeping the

number of layers and head dimension `d_head`$= 32$ fixed. For widths 256, 1024 and 4096, this results in 8, 32 and 128 heads per Transformer block and a total of $30M$, $167M$ and $1.4B$ parameters.

Standard training means AdamW with a single, tuned maximal learning rate, $(\beta_1, \beta_2) = (0.9, 0.95)$, $\varepsilon = 10^{-12}$, sequence length 512, batch size 256, 700 steps of warmup followed by cosine learning rate decay to 10% of the maximal learning rate, weight decay 0.1, gradient clipping. We train for 10681 steps in mixed precision on the DCLM-Baseline dataset (Li et al., 2024). We train all models on the same number of tokens to prevent confounding effects from increased training time.

D.4 Figure Details

Figure 1: The training accuracy of 8-layer MLPs is averaged over 4 runs to reduce noise from random initialization. The training loss of GPT trained with SGD is averaged over 3 runs. GPT with Adam was only run once.

Figure 2: The readout layer and last Layernorm layer are chosen due to their particular importance for logit blowup. The MLP layer was chosen to add a layer that scales hidden-like. This layer was not cherry picked. We observe other MLP layers to have similar scaling properties.

Figure 4: 3-layer MLP trained with SGD on CIFAR-10 with width-dependent learning rate $\eta_n = 0.0001 \cdot n^{-0.5}$. Averages over 4 random seeds.

Figure 5: Minimal unstable learning rates are defined as the smallest learning rates to produce NaN entries when using MSE loss, or, under CE loss, accuracy $< 20\%$ on MNIST and CIFAR-10, and $< 54\%$ on binary multi-index data. For 2-layer MLPs, our theory predicts $\eta_n = O(1)$ as the instability threshold, since there are no hidden layers, and input layers are updated width-independently at $\eta_n = \Theta(1)$. The x-axes showing learning rates are scaled as $(n/256)^\alpha$. In this way, the learning rate at base width 256 remains the same for comparability of the constants. If the optimal or maximal stable learning rate indeed scales as $\eta_n = \eta \cdot n^{-\alpha}$, then the width-dependent scaling of the x-axis $\eta_n \cdot n^\alpha$ shows learning rate transfer.

Figure 6: SGD runs are averaged over 3 random seeds, due to noisy individual outcomes. The results for Adam stem from a single random seed due to limited computational resources. Minimal unstable learning rates are defined as the smallest learning rates to produce loss worse than (optimal CE loss +1) at each width. The x-axes showing learning rates are scaled as $(n/256)^\alpha$. In this way, the learning rate at base width 256 remains the same for comparability of the constants. If the optimal or maximal stable learning rate indeed scales as $\eta_n = \eta \cdot n^{-\alpha}$, then the width-dependent scaling of the x-axis $\eta_n \cdot n^\alpha$ shows learning rate transfer.

Figure 7: Shown is the training accuracy at the end of training for one epoch for the optimal learning rate at each width.

E Refined coordinate checks

The standard coordinate check as provided in the readme of the `mup`-package Yang et al. (2022) may be considered the plot of activation norms $\|x_t^l\|_{RMS}$ after t steps of training for all layers l and the network output norm $\|f\|_{RMS}$ with $f := W_t^{L+1} x_t^L$ as a function of width. Completely width-independent dynamics under μP then result in an approximately width-independent coordinate check of all layers. However, width-dependence in the activations of previous layers would confound the l-th layer activation scaling, so that measuring the effective l-th layer updates requires measuring $\|\Delta W_t^l x_t^l\|_{RMS}$ in each layer, where one may be interested in the weight updates accumulated over the entire course of training $\Delta W_t^l = W_t^l - W_0^l$ or the update in a single step $\delta W_t^l = W_t^l - W_{t-1}^l$. In standard architectures, one can equivalently measure the operator norm of the weight updates $\|\Delta W_t^l\|_{2\to 2} \cdot \sqrt{\texttt{fan\_in}(W_t^l)/\texttt{fan\_out}(W_t^l)} \overset{!}{=} \Theta(1)$ (Yang et al., 2023a); however in non-standard architectures such as Mamba this spectral condition has been shown to fail, so that, in the general case, care should be taken in how exactly weight updates affect the output function (Vankadara et al., 2024). The difference between $\|\Delta W_t x_t\|$ and the preactivation updates $\|\Delta(W_t x_t)\|$ is precisely $\|W_0 \Delta x\|_{RMS}$ which measures the effect of updates propagating from previous layers.

All coordinate checks are run over 4 random seeds either at small learning rate or the optimal learning rate η_{256} at base width 256 (after 1 epoch of training) on CIFAR-10. The learning rate is then scaled in relation to that base width $\eta_n = \eta \cdot (n/256)^\alpha$ with $\alpha \in \{-1, -0.5, 0\}$.

E.1 SGD

Figure E.1 shows the refined coordinate check for a 3-layer MLP in SP with global learning rate scaling $\eta_n = \eta \cdot n^{-1/2}$. As predicted by Proposition 2, the input layer updates decay as n^{-1}, the hidden layer learns features width-independently, and the output scales as $n^{1/2}$ which results in one-hot predictions after the softmax in wide models, but not necessarily unstable training dynamics.

Both $\|\Delta W_t^l x_t^l\|_{RMS}$ and $\|\Delta W_t^l\|_{RMS \to RMS}$ measure the effective update effect in the l-th layer equivalently and accurately even in narrow MLPs of width 64. Naively tracking the activation updates $\Delta x_t^l = x_t^l - x_0^l$ however is confounded by non-vanishing feature learning in narrow models, and only shows the correct hidden- and last-layer scaling exponents for $n \geq 4000$, even after only a single update step.

Figure E.2 shows a refined coordinate check for a 3-layer MLP in SP with width-independent global learning rate scaling $\eta_n = 0.0001 \cdot n^0$. While infinite-width theory predicts the input layer to learn width-independently and the hidden layer to explode as $\Theta(n)$, both empirical exponents are $n^{-1/2}$ smaller, so that the input layer has vanishing feature learning and the hidden layer is still exploding. This ostensible contradiction is resolved when repeating the coordinate check but initializing the last layer to 0 (Figure E.3). Now the predicted scaling exponents are recovered, already at small width. The reason for this subtle but important difference is that the gradient that is back-propagated is given by the last-layer weights, $\partial f / \partial x^L = W_t^{L+1} = W_0^{L+1} + \Delta W_t^{L+1}$. Under standard initialization at the optimal learning rate, the initialization $W_0^{L+1} = \Theta(n^{-1/2})$ still dominates the updates $\Delta W_t^{L+1} = \Theta(\eta_n)$ in absolute terms after a few update steps at widths up to 16384. Comparing the absolute scales of $\|\Delta W_t^l x_t^l\|_{RMS}$ or $\|W_t^l\|_*$ in both figures confirms this hypothesis. The pure update effects in Figure E.3 have lower order of magnitude in the constant before the scaling law, but follow clear scaling exponents. Therefore the faster scaling law under last-layer zero initialization can be extrapolated with certainty to induce a phase transition under standard initialization around width $4 \cdot 10^7$. We do not have sufficient computation resources to validate this but arrive at this order of magnitude irrespective of whether we extrapolate the scaling laws of $\|\Delta W_t^l x_t^l\|_{RMS}$ or $\|W_t^l\|_*$ as well as of the input or hidden layer laws. For base width n_0 and width-dependent statistics Δ_n^1 and Δ_n^2 with differing scaling exponents c_1 and c_2, Δ_n^1 and Δ_n^2 intersect at width $n_0 \cdot (\Delta_{n_0}^2 / \Delta_{n_0}^1)^{1/(c_1 - c_2)}$.

This consequential difference in empirical scaling exponents at realistic widths due to a subtle difference in last-layer initialization highlights the attention to detail that is required to make accurate scaling predictions from infinite-width limit theory, but, as we show in this paper, apparent contradictions can often be reconciled with enough attention to detail, and the clean scaling laws we arrive at as a result already hold at moderate scales and prove the usefulness of investing this extra effort.

Hence, one reason why scaling exponents in SGD can be larger than predicted up to very large widths, is due to differing orders of magnitude in the constant pre factors in the initialization versus update terms in the backward pass. Without our refined coordinate check, the phase transition around width 10^7 is hard to predict.

As predicted, the width-exponents of 2-layer nets behave like the input and output layer in 3-layer nets (Figure E.4).

When choosing the optimal learning rate $\eta_{256} = 0.03$ at width 256, stronger finite-width effects due to non-vanishing input layer feature learning already occur after a few steps and make the update scaling exponents after 10 steps only visible at larger width $n \geq 2048$ (Figure E.5). As long as divergence is prevented in the first few steps, self-stabilization mechanisms such as activation sparsification can quickly contain the initial catapult (Figure E.6). In deeper networks, explosion of several hidden layers is increasingly difficult to stabilize, and finite width effects are reduced.

Figure E.5 shows the effective update rank and the alignment between activations at initialization versus at time t for the same input training points under unstable width-independent learning rate scaling. The updates in each layer are remarkably strongly dominated by a single direction. As hidden-layer activations are slowly diverging, their alignment is only beginning to decrease at large widths $n \geq 4096$. The beginning instability of $\|\Delta x^2\|_{RMS}$ will eventually induce training instability and suboptimal accuracy at large width, which is hard to predict without tracking the layerwise effective update scaling across widths.

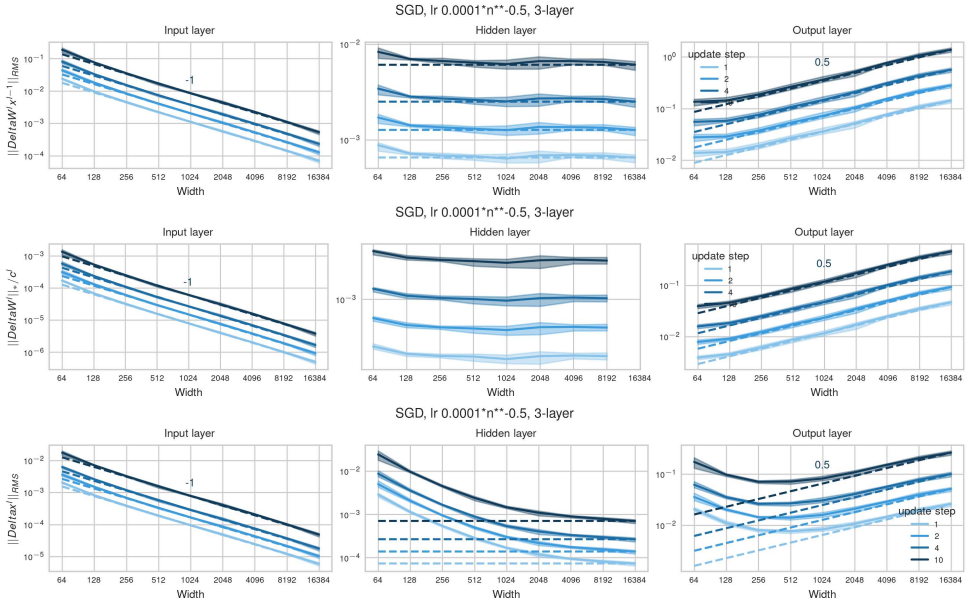


Figure E.1: **(Hidden layer feature learning in SP under intermediate learning rate scaling)** Effective l-th layer update scalings $\|\Delta W_t^l x_t^{l-1}\|_{RMS}$ (top), weight update spectral norm $\|\Delta W_t^l\|_*$ (2nd row) and activation updates δx^l (bottom) of MLPs trained in SP with small learning rate $\eta_n = 0.0001 \cdot (n/256)^{-1/2}$ scaled to preserve hidden-layer feature learning. The TP scaling predictions are accurate. Hidden layers learn features width-independently, and input layers have vanishing feature learning. At moderate widths, activation updates are confounded by previous layer updates, and thus do not provide an accurate metric for effective update scaling.

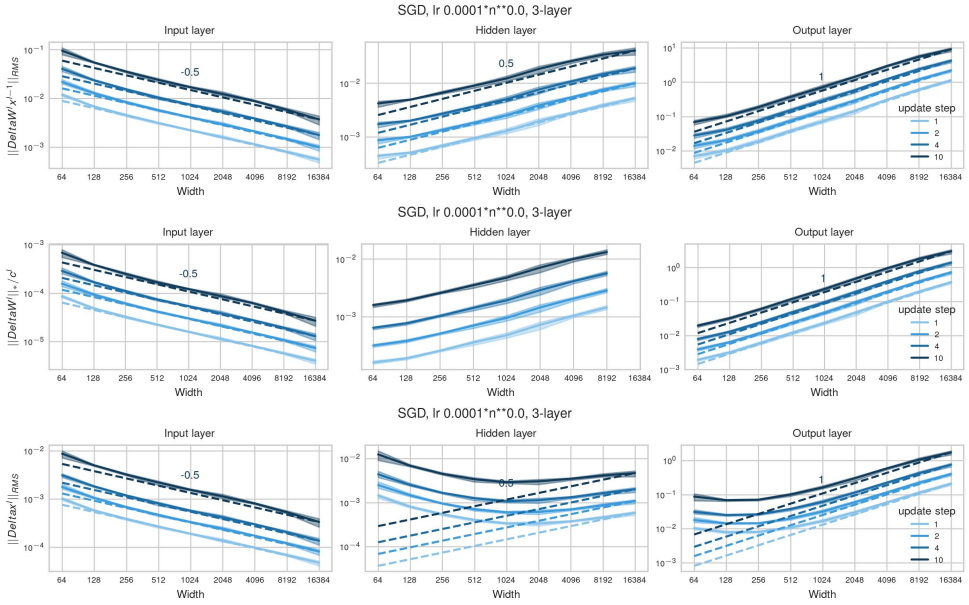


Figure E.2: **(Inaccurate exponent predictions under standard initialization with large learning rate scaling)** Effective l-th layer update scalings $\|\Delta W_t^l x_t^{l-1}\|_{RMS}$ (top), weight update spectral norm $\|\Delta W_t^l\|_*$ (2nd row) and activation updates δx^l (bottom) of 3-layer MLPs trained in SP with width-independent $\eta_n = 0.0001$. Hidden layer activation updates explode, and input layers have vanishing feature learning. By TP scaling predictions, however, the input layer should learn features width-independently. Instead, the TP scaling exponents are **only accurate under last-layer zero initialization, not under standard initialization** (see Figure E.3 for last-layer zero initialization) as the initialization scaling $W_0^{L+1} = \Theta(n^{-1/2})$ still dominates the update scaling $\Delta W_t^{L+1} = \Theta(\eta_n)$ at realistic widths after a few updates under the optimal learning rate. Hence, the backpropagated gradient $\partial f / \partial x^L = W_t^{L+1}$, relevant for the hidden and input layer updates, behaves for a several steps like it should only behave in the first step. By comparing the absolute scales here versus those in Figure E.3 it becomes apparent that this is indeed a finite-width effect, as the absolute scale of $\|\Delta W x\|_2$ here is on the order 10^{-1} and 10^{-2} for input and hidden layer, respectively, whereas the pure update effects under last-layer zero initialization are of at most order 10^{-4} for both layer types. Clearly for sufficient width, the differing scaling exponents will induce a phase transition toward the predicted scaling exponents. While the input layer learns features width-independently under last-layer zero initialization, as predicted by TP theory, this is not the case at realistic scales under standard initialization. The qualitative statement that standard parameterization with width-independent learning rates is not activation stable in deep networks is still accurate at moderate width.

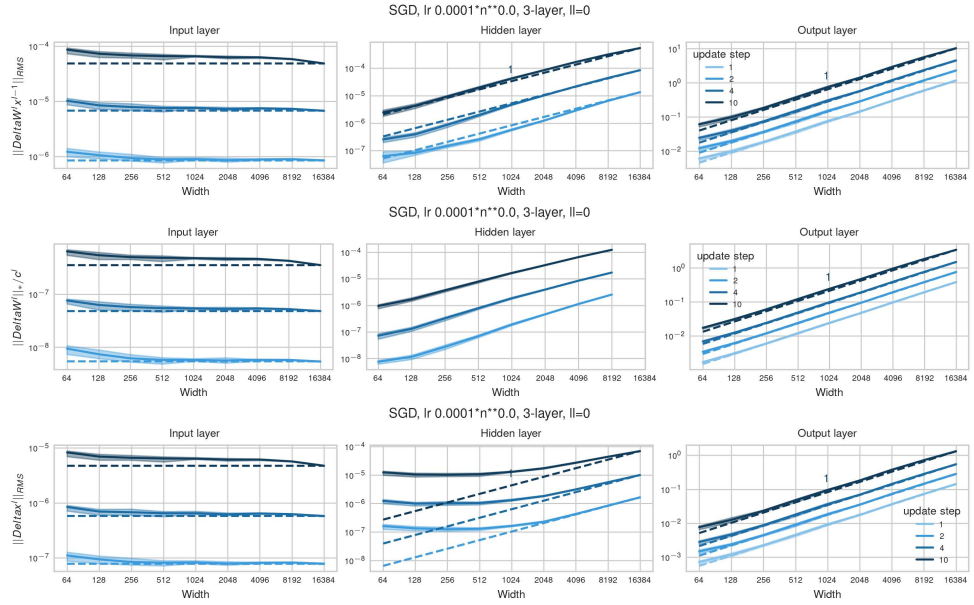


Figure E.3: **(Accurate exponent predictions in SP with last-layer zero initialization under large learning rate scaling)** Same as Figure E.2 with width-independent $\eta_n = 0.0001$ but initializing the last layer to zero. Here, the TP scaling predictions are accurate. Hidden layer activation updates explode as n^1, and input layers learn features width-independently. Observe a smaller absolute scale of the pure update effects here versus in Figure E.1 that explains the differing exponents there. The updates in the input and hidden layers vanish in the first step, as the gradient for backprop is $W_0^{L+1} = 0$.

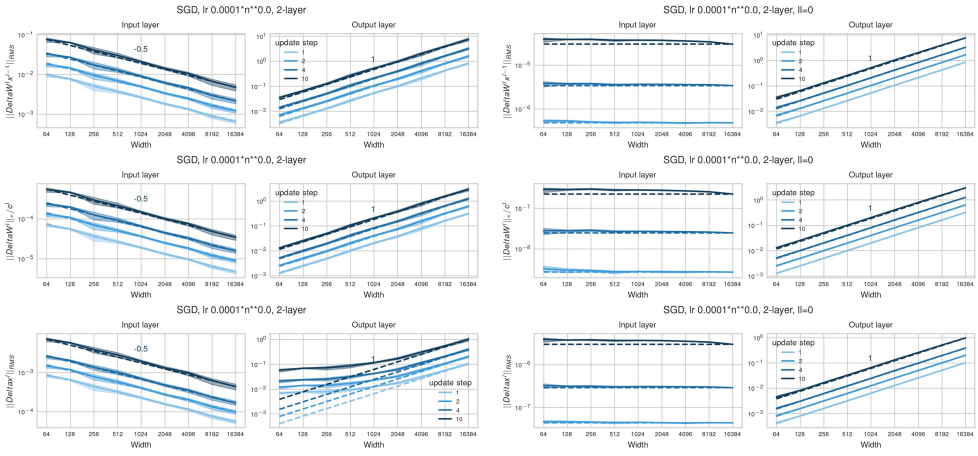


Figure E.4: **(Shallow nets learn features width-independently under large learning rate scaling)** Same as Figure E.1 but for 2-layer MLPs trained in SP with width-independent $\eta_n = 0.0003$ with standard initialization (left) and last-layer initialized to 0 (right). The input layer and output layer scalings behave as in the 3-layer nets. Since there is no exploding hidden layer, activation stability is preserved in 2-layer nets under $\eta_n = \Theta(1)$.

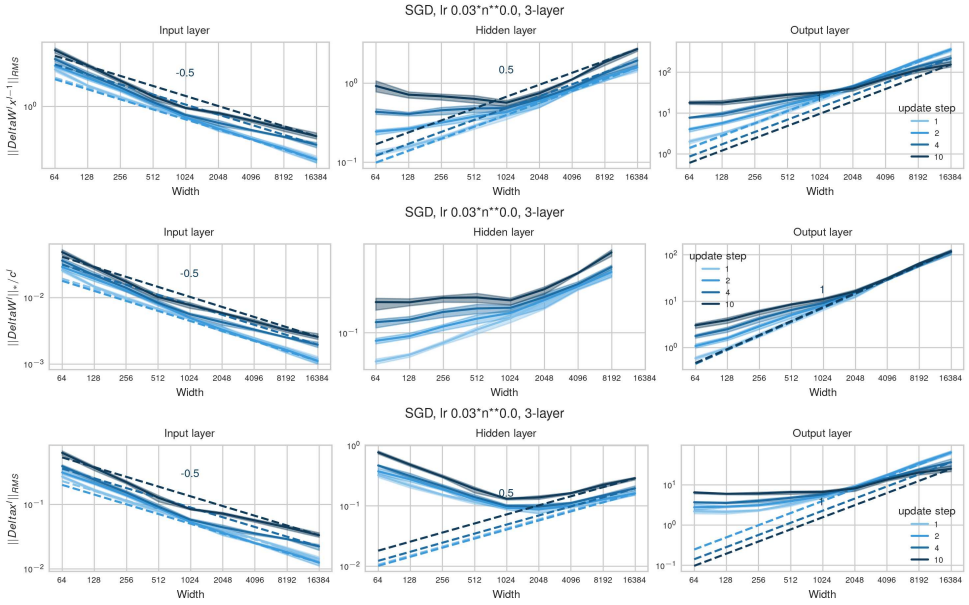


Figure E.5: **(Large finite-width effects at optimal learning rate in shallow 3-layer MLPs)** At the optimal learning rate $\eta_{256} = 0.03$ with width-independent scaling, non-vanishing input layer feature learning confounds the scalings after few update steps up to moderate widths $n \leq 1024$, similar to Adam (Figure E.9).

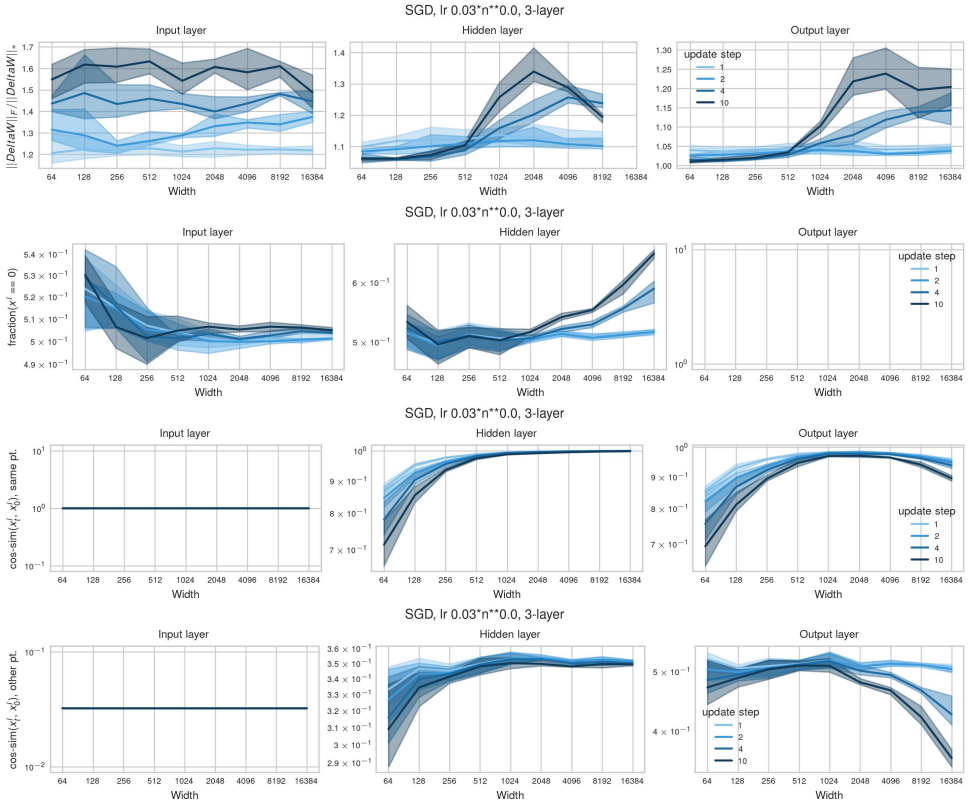


Figure E.6: **(Activation sparsification at the optimal learning rate)** Effective l-th layer update ranks $\|\Delta W_t^l\|_F/\|\Delta W_t^l\|_*$, activation sparsity and cosine similarity between activations to each layer comparing time 0 and time t on the same input training point and on differing training points in the same batch of 3-layer MLPs trained with SGD in SP with width-independent learning rate $\eta_n = 0.03$ as in Figure E.5. As opposed to the gradient flow regime, at the optimal learning rate, there are significant self-stabilization effects at large width already after 10 steps through activation sparsification but less through activation rotation.

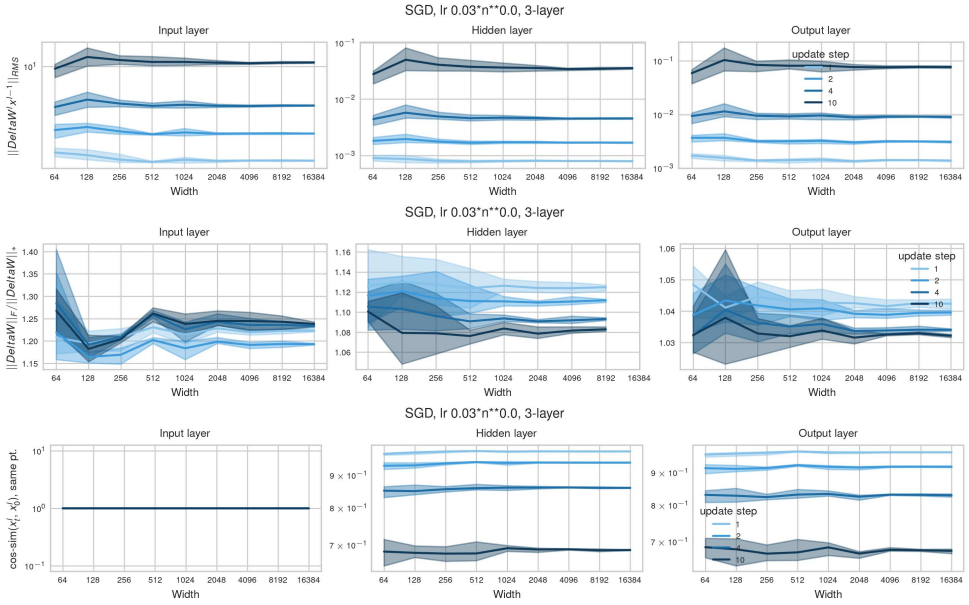


Figure E.7: **(Full width-independence in μP)** Effective l-th layer updates (top), effective update ranks $\|\Delta W_t^l\|_F/\|\Delta W_t^l\|_*$ (second row) and cosine similarity between activations to each layer comparing time 0 and time t on the same input training point (bottom) of 3-layer MLPs trained with SGD in μP with width-independent learning rate $\eta_n = 0.03$. As expected, all statistics behave width-independently. The effective update rank is remarkably small, as for SP. The activation are rotated quite quickly.

E.2 Adam

With $\eta_n = \Theta(n^{-1/2})$, the optimal learning rate scaling for 3-layer MLPs with Adam on CIFAR-10 is larger than predicted (Figure F.22). Figure E.9 shows that this may be due to large finite-width effects for Adam at optimal learning rate multiplier $\eta_{256} = 0.0003$ and moderate width $n \leq 8192$. While the weight update spectral norm scales as predicted, the input-layer gets large updates at moderate width (Figure E.10) and induces a strong rotation of the activations. As a result, the activation explosion only sets in at large width $n \geq 8192$. This qualitative change toward vanishing input layer feature learning will result in a phase transition toward unstable scaling at large widths which is hard to predict at small scale from measurements alone, except when measuring both $\|\Delta W^l\|_*$ and the alignment $\|\Delta W^l x^{l-1}\|_{RMS}$.

As opposed to SGD, observe large finite-width effects in the activation updates even under small absolute learning rate 10^{-6} at moderate width $n \leq 8192$ (Figure E.8).

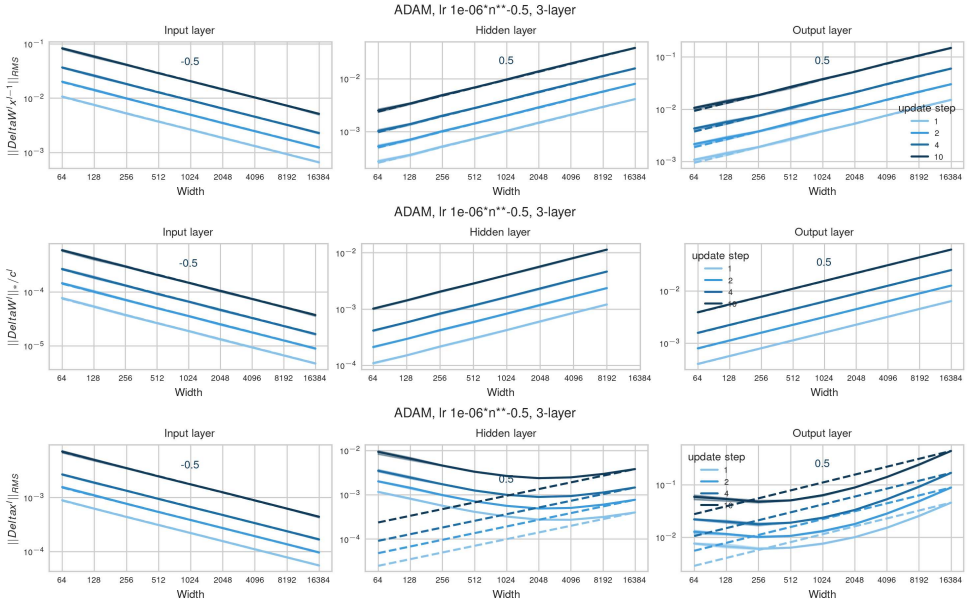


Figure E.8: **(Large finite-width effects from input-layer updates in Adam)** Effective l-th layer update scalings $\|\Delta W_t^l x_t^{l-1}\|_{RMS}$ (top), weight update spectral norm $\|\Delta W_t^l\|_*$ (2nd row) and activation update norm $\|\delta x^l\|_{RMS}$ (bottom) of 3-layer MLPs trained with Adam in SP with $\eta_n = 10^{-6} \cdot n^{-1/2}$. Observe the theoretically predicted exponents in $\|\Delta W^l\|_*$ do not transfer to the activation updates at moderate width $n < 8192$ due to large non-vanishing input layer updates at moderate width. Even the effective updates $\|\Delta W_t^l x_t^{l-1}\|_{RMS}$ do not perfectly align with the scaling law at infinite width, indicating that the alignment between ΔW_t^l and x_t^{l-1} evolves non-trivially across width and that the spectral norm $\|\Delta W^l\|_*$ and pure infinite-width predictions are less useful for explaining the behaviour of Adam at moderate width.

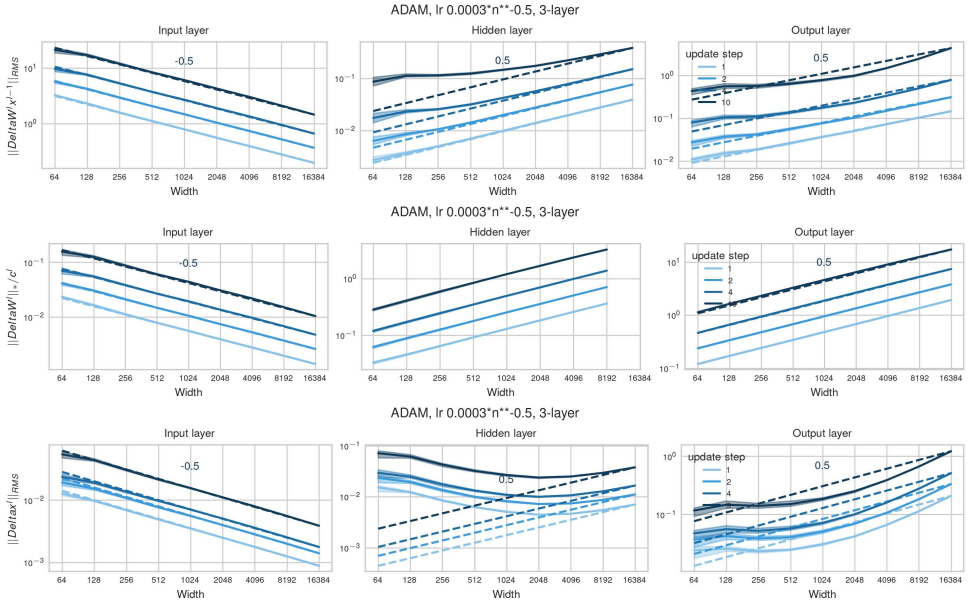


Figure E.9: **(Large finite-width effects from input-layer updates in Adam)** Effective l-th layer update scalings $\|\Delta W_t^l x_t^{l-1}\|_{RMS}$ (top), weight update spectral norm $\|\Delta W_t^l\|_*$ (2nd row) and activation update norm $\|\delta x^l\|_{RMS}$ (bottom) of 3-layer MLPs trained with Adam in SP with large $\eta_n = 0.0003 \cdot n^{-1/2}$. Observe the theoretically predicted exponents in $\|\Delta W^l\|_*$ do not transfer to the activation updates at moderate width $n < 8192$ due to large non-vanishing input layer updates at moderate width. Even the effective updates $\|\Delta W_t^l x_t^{l-1}\|_{RMS}$ do not perfectly align with the scaling law at infinite width, indicating that the alignment between ΔW_t^l and x_t^{l-1} evolves non-trivially across width and that the spectral norm $\|\Delta W^l\|_*$ and pure infinite-width predictions are less useful for explaining the behaviour of Adam at moderate width.

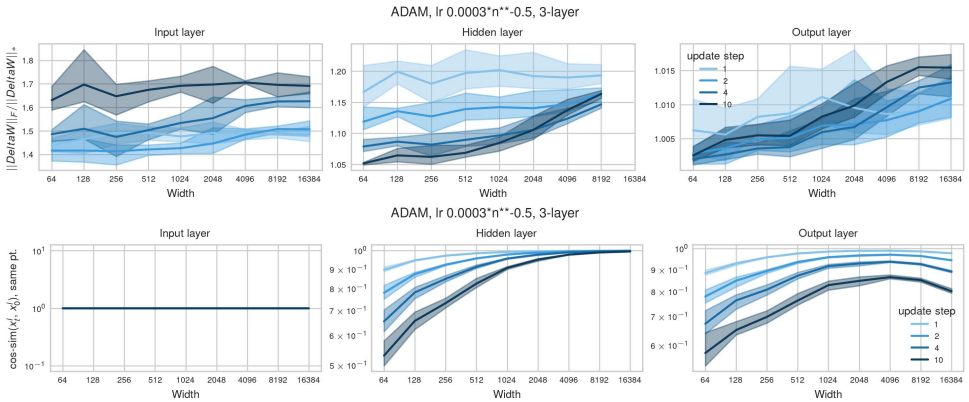


Figure E.10: **(Strong activation rotation under Adam at moderate width)** Effective l-th layer update ranks $\|\Delta W_t^l\|_F / \|\Delta W_t^l\|_*$ (top) and cosine similarity between activations to each layer comparing time 0 and time t on the same input training point (bottom) of 3-layer MLPs trained with ADAM in SP with large $\eta_n = 0.0003 \cdot n^{-1/2}$. The effective update rank is mostly growing in time in the input layer. Already after a few steps, the first-layer activation coordinates are drastically rotated at moderate widths. This induces a u-curve in the hidden-layer activations that inherit large rotation from the input layer at moderate width and update too much at large width under $\eta_n = \Theta(n^{-1/2})$.

E.3 Normalization layers and Adam provide robustness to miss-initialization

For MLPs trained with SGD, initialization greatly impacts the training dynamics as both the forward and the backward pass are affected (Figure E.11). Large input layer initialization induces update instability at large width, which is stabilized by extreme activation sparsification (Figure E.13).

By adding normalization layers, the forward pass can be enforced to scale width-independently. This may affect the gradients. But the gradient norms become irrelevant under Adam with sufficiently small ε. Adding both normalization layers and Adam to MLPs, observe that initialization is barely relevant for update scalings (Figure E.12), and other downstream statistics such as activation sparsity (Figure E.14). Here we use RMSNorm to fairly compare activation sparsity, but we expect LayerNorm to induce the same scaling behaviour.

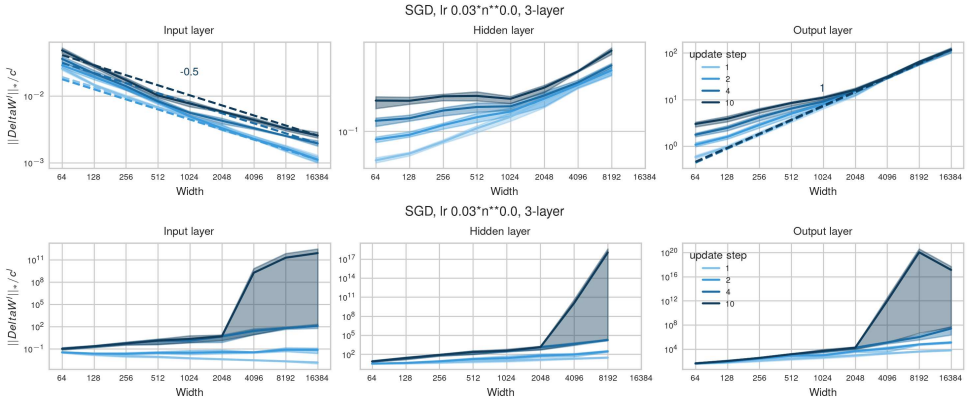


Figure E.11: **(Initialization matters in MLPs with SGD)** SP (top), SP with large input layer variance 2 (bottom). The initializations induce significant differences in the training dynamics. Large input layer normalization becomes unstable at large width.

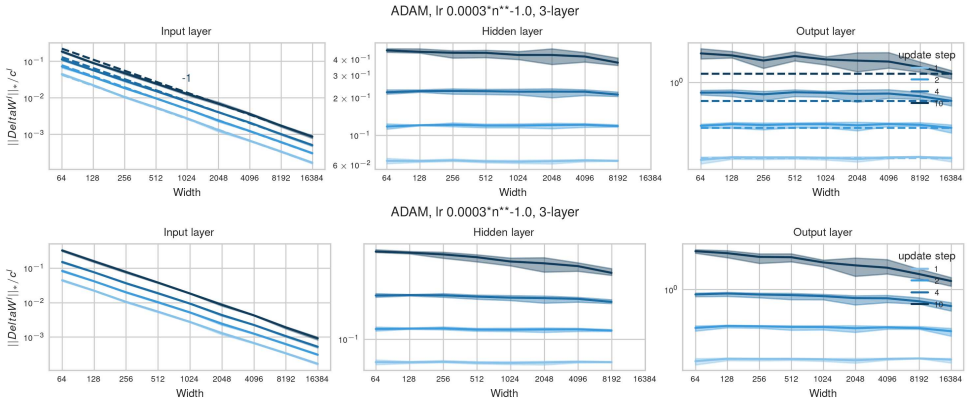


Figure E.12: **(Differing initialization barely matters with normalization layers and Adam)** Update spectral norms of MLPs with the most basic normalization layer RMSNorm after every layer trained with Adam and initialized with SP (top) versus SP with large input layer variance 2 (bottom). Here, initialization barely impacts the update scaling.

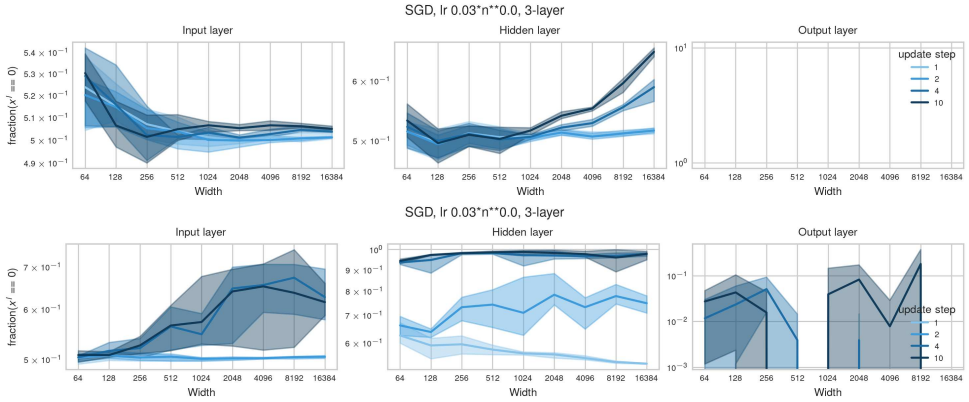


Figure E.13: **(Big difference in activation sparsity under SGD)** SP (top), SP with large input layer variance 2 (bottom). Large input variance has to be stabilized by increased activation sparsity.

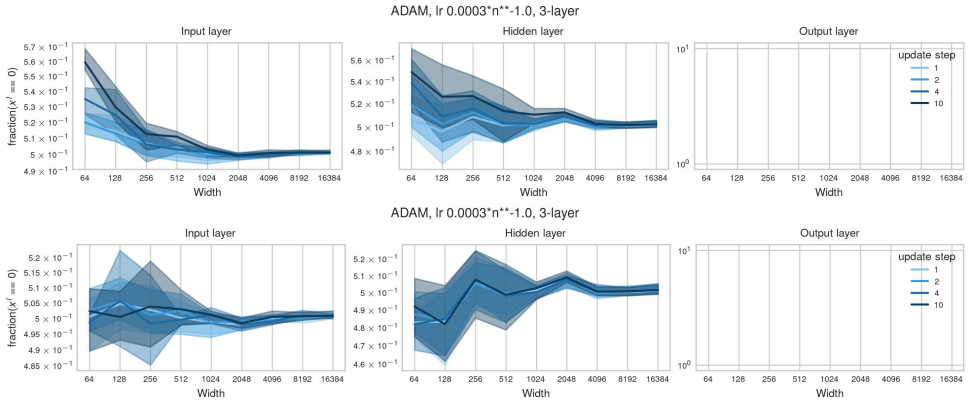


Figure E.14: **(Activation sparsity barely affected under normalization)** Same as Figure E.12 but showing the fraction of activation entries that equal 0. Both initializations do not significantly sparsify activations beyond 50%.

E.4 Alignment and update scaling in Transformers

Since we measure width-independent alignment $\alpha_{W_0^{L+1}\Delta x_t^L} = \Theta(1)$ (Figures 2 and E.15), under large output dimension $d_{\text{out}} \gg n$ (as is typical in language settings), $\|W_0\|_{op}$ approximately scales $\Theta(1)$ (Vershynin, 2010), as opposed to $\Theta(n^{1/2})$ at sufficient width $n \gg d_{\text{out}}$. The term $W_0^{L+1}\Delta x_t^L$ therefore induces approximately width-independent logit updates even under standard last-layer initialization, in the regime $d_{\text{out}} \gg n$ (cf. Figure E.17), but it induces logit divergence at sufficient width $d_{\text{out}} \ll n$.

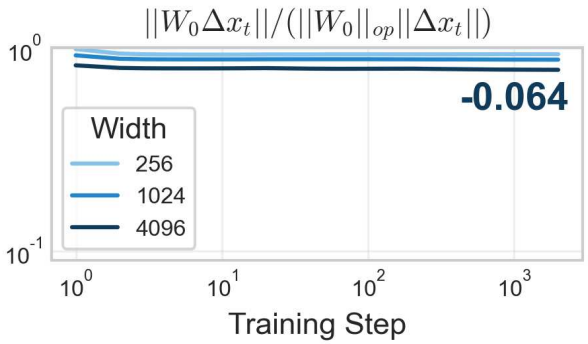


Figure E.15: **(Updates propagate maximally in the readout layer in SP-full-align)** The operator norm ratio for propagating activations in the readout layer for training GPT with AdamW in SP-full-align with near-optimal learning rate $\eta_n = 0.00316$. The ratio is barely width dependent so that propagated activations can be computed when knowing both $\|W_0\|_{op} = \|W_0\|_{RMS\to RMS}$ and $\|\Delta x_t\|_{RMS}$.

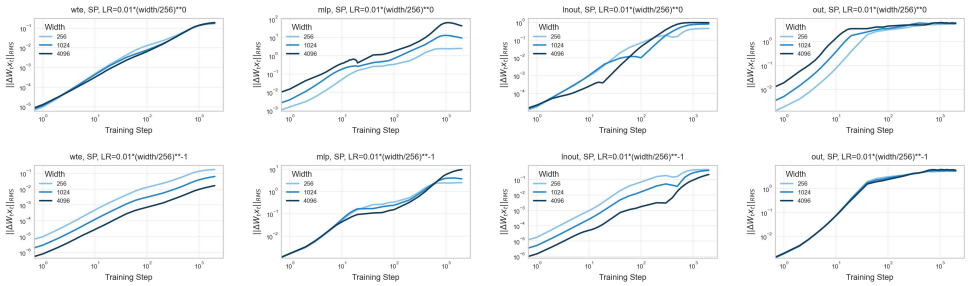


Figure E.16: **(Effective updates follow predictions)** Effective updates $\|\Delta W_t x_t\|$ for constant learning rate scaling $\eta_n = 0.01$ (top) and stable learning rate scaling $\eta_n = 0.01 \cdot (n/256)^{-1}$ (bottom) in GPT models of varying width (the darker, the wider) for the embedding layer, the first MLP layer in the Transformer block 2, the last Layernorm before the readout layer and the readout layer (from left to right). At constant learning rate, hidden and output layers diverge with width. At optimal learning rate, embedding and normalization layer updates vanish with width.

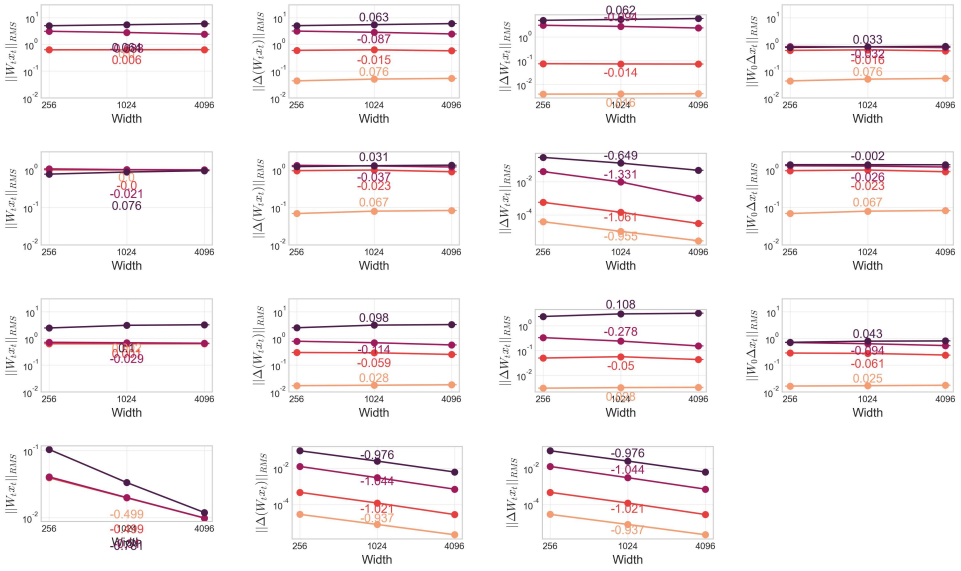


Figure E.17: **(Refined coordinate checks for GPT in SP with Adam and $\eta_n = 0.01 \cdot n^{-1}$)** *From left to right:* Activation norm, activation updates, effective updates $\|\Delta W_t x_t\|_{RMS}$, propagating updates $\|W_0 \Delta x_t\|_{RMS}$ after 2, 10, 100 and 700 batches of training (the darker, the more batches). *Layers from top to bottom:* readout, last Layernorm, first MLP layer in Transformer block 2, embedding layer. Infinite width-scaling predictions are accurate in all effective update terms $\|\Delta W_t x_t\|_{RMS}$: Embedding and Layernorm layers scale input-like and their updates vanish as $\Theta(n^{-1})$, all hidden and output layers are effectively updated width-independently. Against the infinite-width prediction, logit updates do not explode, not because of miss-alignment but because output dimension is much larger than width $d_{\text{out}} \gg n$, which changes the approximate scaling of $\|W_0^{L+1}\|_{RMS \to RMS}$ from $\Theta(n^{1/2})$ in the infinite-width limit, to $\Theta(1)$ in the large output dimensional regime.

F Empirical learning rate exponents

F.1 Summary of the MLP experiments in this section

In general, the optimal learning rate exponent appears to be architecture- as well as data-dependent. We conjecture that the optimal learning rate scaling is subject to opposing objectives. Ideally, the effective updates in all layers scale width-independently. Since this cannot be achieved with a single learning rate for input, hidden and output layers, the layer types act on the optimal learning rate scaling as opposing forces.

SGD under MSE loss. For SGD under MSE loss, output blowup results in unstable training dynamics so that the maximal stable and optimal learning rate robustly scales as $\eta_n = \Theta(n^{-1})$ across architectures and datasets. As a consequence of vanishing feature learning, neither training nor test loss monotonically improve with scale under MSE loss.

Random feature models. When only training the last layer, fully width-independent training dynamics are achieved with $\eta_n = \eta \cdot n^{-1}$. Figure F.18 shows that this exponent clearly results in learning rate transfer for 2-layer ReLU random feature networks on CIFAR-10. Also observe that since all learning rate scalings recover activation-stability, larger than optimal learning rates still result in non-trivial classification accuracy.

Deep MLPs. With an increasing amount of hidden layers, their width-independence eventually outweighs input layer feature learning in vision datasets. For at least 6 layers, we see approximate learning rate transfer under $\eta_n = \Theta(n^{-1/2})$ for SGD and $\eta_n = \Theta(n^{-1})$ for Adam as predicted for width-independent hidden layer feature learning for both CIFAR-10 and MNIST.

Shallow ReLU MLPs at moderate width and (deep) linear networks are not useful proxy models for deep nonlinear networks. For shallow MLPs, we often observe stronger finite-width effects than for deeper networks causing larger than predicted optimal learning rate scaling at moderate width, as divergence in fewer hidden layers can be stabilized over the course of training up to larger widths (cf. Appendix E). In linear networks, on the other hand, feature learning is not essential as the learned function always remains linear. Consequently we often observe that optimal learning rates decay faster than maximal-stable learning rates in (deep) linear networks even under CE loss (Figures F.12 and F.31). These differences between deep non-linear networks and toy architectures suggest that shallow MLPs and deep linear networks do not serve as useful proxy models for practical non-linear networks in terms of optimal learning rate exponents at moderate width.

Input layer task. Under multi-index data with a sparse signal and high-dimensional isotropic covariates (explained in Appendix D.2), learning the two signal input dimensions is particularly useful for good generalization. Appendix F.3 shows the predicted exponent $\eta_n = \eta \cdot n^0$ for input layer learning in 2-layer MLPs. Deeper MLPs recover hidden layer stability with optimal learning rate scaling $\eta_n = \Theta(n^{-1/2})$. Observe that generalization suffers when the input layer does not learn to align with the signal dimensions, so that only the 2-layer MLP with CE loss generalizes well at large width.

Standard initialization with μP learning rates (SP-full-align). While Everett et al. (2024) report good transfer properties of SP-full-align, Appendix F.8 shows that the optimal learning rate clearly shrinks across image datasets and our multi-index data. We also introduce a variant of this parameterization that matches the $n^{1/2}$ logit blowup rate from the term $W_0^{L+1}\Delta x_t$ in the effective last-layer updates by increasing the last-layer learning rate. This variant performs similarly well as SP-full-align. In particular, both variants seem to be less learning rate sensitive than μP.

Adam learns features with $\eta_n = \eta \cdot n^{-1}$. Adam simplifies the learning rate scaling for weight W to $\eta_W = \eta/\texttt{fan\_in}(W)$, because the gradient is normalized but still correlated with the incoming activations since the sign is preserved in each entry. Thus $\eta_n = \eta/n$ is expected to induce width-independent hidden- and output-layer learning, but vanishing input-layer learning since here `fan_in` is fixed and hence would require constant learning rate scaling. As for SGD, we still observe the optimal learning rate scaling $\eta_n = \eta \cdot n^{-1}$ in deep MLPs on MNIST (Appendix F.5) and on CIFAR-10 (Appendix F.7), indicating that width-independence in hidden- and output-layer dominates input layer feature learning.

F.2 Transformer experiments

As we consider single-pass training, training and validation loss approximately coincide, so that statements about the training loss transfer to statements about the validation loss irrespective of the optimizer. All figures in this section show training loss on the left and validation loss on the right.

Stabilizing techniques like gradient clipping can improve the absolute learning rate multiplier in front of the scaling law, but do not seem to change the width-scaling exponent for SGD (Figure F.4 vs Figure F.5).

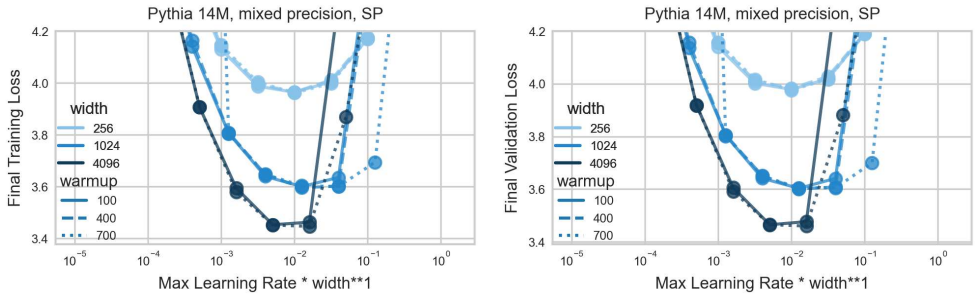


Figure F.1: **(Instability without qk-Layernorm)** Train loss (left) and validation loss (right) of single-pass AdamW training without qk-Layernorm. Training and validation loss approximately coincide. Optimal learning rate scaling is dominated by the maximal stable learning rate scaling that is at most $\Theta(n^{-1})$. But without qk-Layernorm, the stability threshold is decreasing faster than $\Theta(n^{-1})$ even when increasing warmup length, so that it may be that the instability threshold would decay beyond the ideal learning rate and performance suffers. As our computational budget does not allow us to scale further, this setting remains inconclusive.

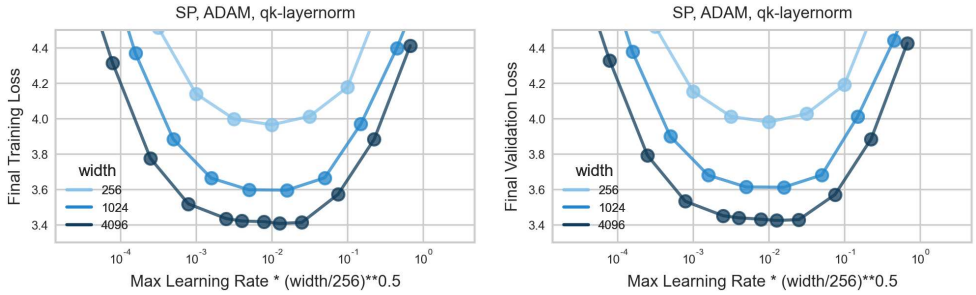


Figure F.2: **Large learning rate stability with qk-Layernorm** Same as Figure F.1 but with qk-Layernorm as recommended by Wortsman et al. (2024). Training and validation loss approximately coincide. The optimal learning rate seems to approximately transfer under $\eta_n = \eta \cdot n^{-1/2}$, so the added Layernorm appears to stabilize learning at larger learning rate scaling, similar to the softmax in CE loss.

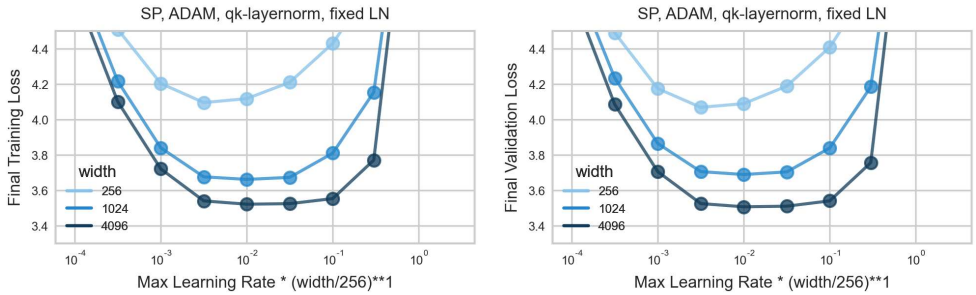


Figure F.3: Same as Figure F.1 with qk-Layernorm as recommended by Wortsman et al. (2024), but all trainable Layernorm parameters are fixed to initialization. Here only the embedding layer behaves input-like, so that all other parameters learn width-independently under learning rate scaling $\Theta(n^{-1})$. While the optimum is drifting toward larger learning rates, an increasingly large plateau of near-optimal learning rates emerges at large width. $\Theta(n^{-1})$ still approximately captures the maximal stable learning rate scaling.

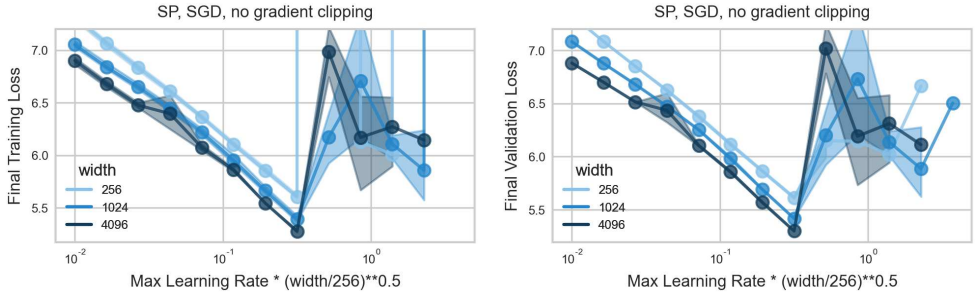


Figure F.4: **(GPT trained with SGD has $\Theta(n^{-1/2})$-learning rate scaling)** Train loss (left) and validation loss (right) of single-pass SGD training (averaged over 3 random seeds affecting weight initialization and data shuffling). Training and validation loss approximately coincide. Hence also validation-optimal learning rate scaling is dominated by maximal stable learning rate scaling $\Theta(n^{-1/2})$ for hidden-layer stability.

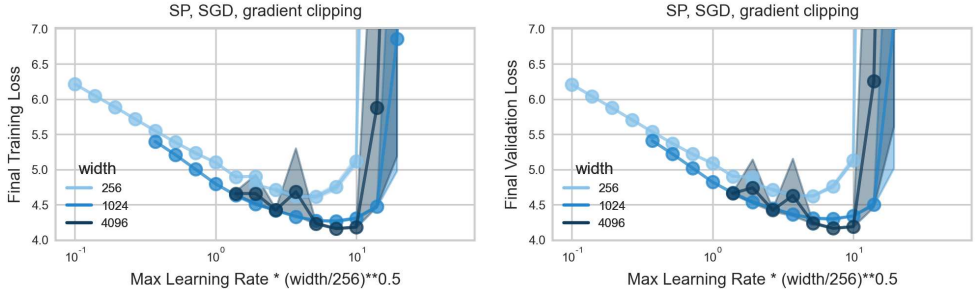


Figure F.5: Same as Figure F.4 but with gradient clipping. Performance is significantly improved as larger learning rate constants are stable (observe similar performance as without gradient clipping at the same learning rate). Optimal learning rate scaling is still dominated by the maximal stable learning rate scaling $\eta_n = \eta \cdot n^{-1/2}$ for hidden-layer stability.

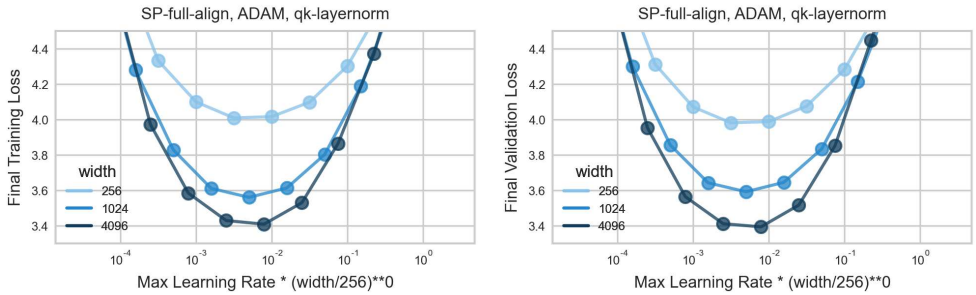


Figure F.6: Same as Figure F.2 but in SP-full-align. AdamW in SP-full-align and SP with a global learning rate seem to have similar performance without multiplier tuning. SP-full-align approximately transfers the learning rate here in the $d_{\text{out}} \gg n$ regime, but not in the $d_{\text{out}} \ll n$ regime in Appendix F.8.

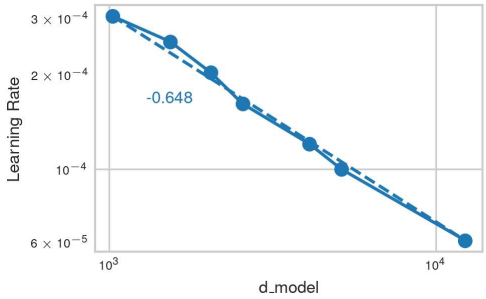


Figure F.7: **Large learning rate exponent in original GPT paper.** Just plotting the reported learning rate and d_model values from Brown et al. (2020) results in quite a stable scaling law with exponent -0.648, which is larger than -1 required for hidden-layer stability but significantly smaller than 0 required for width-independent input layer learning. But note that jointly increasing batch size, n_layers and n_heads might be confounding factors here.

F.3 Cross-entropy loss enables large-learning rate training

MLPs on multi-index data. Here we train 2-layer and 3-layer ReLU MLPs on generated multi-index teacher data as detailed in Appendix D. These data crucially differ from the other considered datasets in that the target function only depends on the first 2 input dimensions. Due to the isotropic covariate distribution, input layer feature learning is necessary for good generalization. Hence we observe a clear $\eta_n = \Theta(1)$ scaling for 2-layer MLPs with CE loss, necessary for preserving input layer feature learning (Figure F.8). 3-layer MLPs attain the maximal activation-stable exponent $\eta_n = \Theta(n^{-1/2})$ in CE loss (Figure F.9). 2-layer MLPs preserve a better validation accuracy compared to their training accuracy than deeper nets, as input layer learning gets increasingly inhibited by $\Theta(1)$-learning rate instability in the presence of hidden layers. Both for shallow and deeper MLPs with MSE loss, we lose feature learning under the maximal output-stable scaling $\eta_n = \Theta(n^{-1})$, as expected.

In this setting, it becomes particularly apparent that using the MSE loss with a softmax applied to the output of the network is not desirable. Ultimately, the only difference to CE loss is that the loss derivative with respect to the network output $f(\xi) := W^{L+1}x^L(\xi)$ becomes

$$\left(\frac{\partial \mathcal{L}}{\partial f}\right)_j = \sum_{i\in[C]} (\sigma(f)_i - y_i)\sigma(f)_i(\delta_{ij} - \sigma(f)_j),$$

where the inner derivative of the softmax $\sigma_i(\delta_{ij} - \sigma_j)$ vanishes as soon as the outputs diverge $|f_i(\xi) - f_j(\xi)| \to \infty$ on a training point ξ. Hence, while the softmax still mitigates output blowup in the forward pass, the gradients vanish under output blowup. The CE loss, on the other hand, is exactly the correct choice of loss function to cancel out the inner derivative of the softmax and effectively view $\sigma(f)$ as the output of the network, resulting in $\left(\frac{\partial \mathcal{L}}{\partial f}\right)_j = \sigma(f)_j - y_j$.

Here vanishing gradients under output blowup in the MSE+softmax setting is so severe that output blowup prevents learning under large learning rates and the optimal learning rate scales as $\Theta(n^{-1})$.

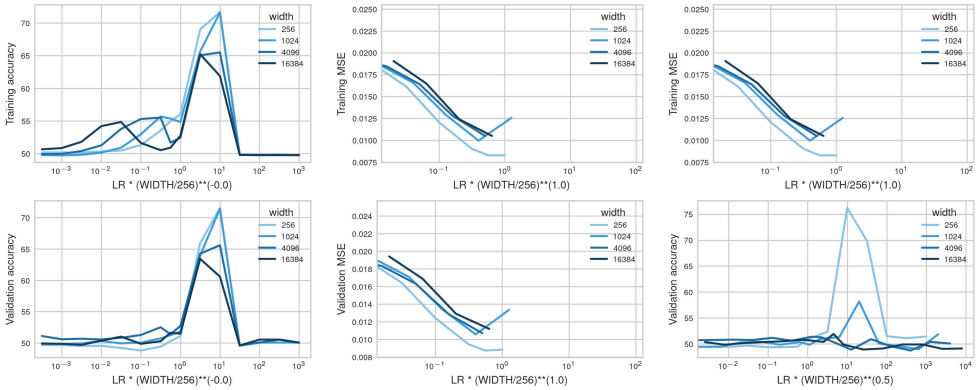


Figure F.8: **(Cross-entropy loss increases maximal stable learning rate scaling to approximately $\Theta(1)$ in 2-layer nets)** Training accuracy (top) and validation accuracy (bottom) for a 2-layer MLP on generated multi-index teacher data (mean over 4 seeds) with CE loss (left), MSE loss (center) and MSE loss with softmax (right). The x-axis scales the learning rate with width-dependent exponents; observe approximate transfer under $\Theta(1)$, $\Theta(n^{-1})$ and $\Theta(n^{-1})$ scaling, respectively. In the MSE plot, ending lines indicate divergence for larger learning rates. MSE loss with softmax on the output does not increase optimal learning rate scaling due to vanishing gradients and gets worse due to a lack of input layer feature learning.

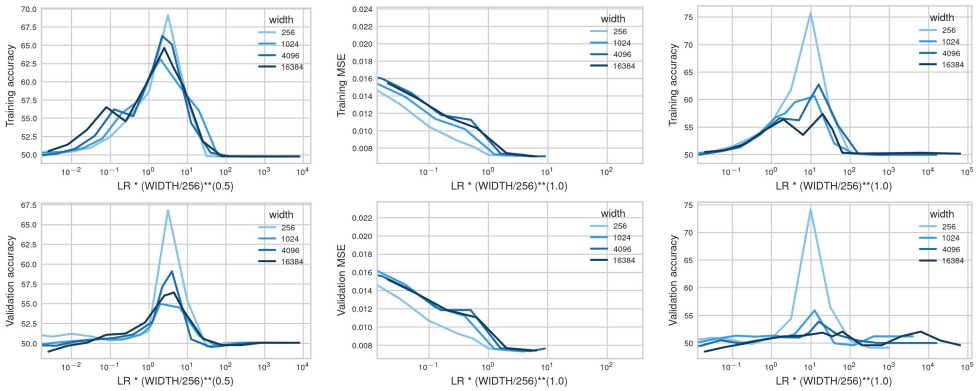


Figure F.9: **(Cross-entropy loss increases maximal stable learning rate scaling to approximately $\Theta(n^{-1/2})$ in 3-layer nets)** Same as Figure F.8 but for a 3-layer MLP. The x-axis scales the learning rate with width-dependent exponents; observe approximate transfer of the maximal stable learning rate under $\Theta(n^{-1/2})$, $\Theta(n^{-1})$ and $\Theta(n^{-1})$ scaling, respectively. In the MSE plot, ending lines indicate divergence for larger learning rates. Observe that wider networks generalize worse with scale as they lose input layer feature learning.

F.4 MLPs with SGD on MNIST

With MSE loss, observe a clear $\mathcal{O}(n^{-1})$ optimal and maximal stable learning rate exponent for all network variants (Figure F.10).

With CE loss, observe that 2-layer MLPs transfer the optimal and maximal stable learning rate n^0 (Figure F.11). 3 and 4 layer MLPs are still able to transfer the maximal stable learning rate, indicating self-stabilization of activation blowup at moderate width. In 6, 8 and 10 layer MLPs the maximal stable learning rate scaling $n^{-1/2}$ becomes increasingly pronounced, as it becomes increasingly difficult to stabilize activation blowup in an increasing amount of hidden layers while an increasing amount of layers is learning under $n^-1/2$ learning rate scaling.

3 and 4 layer linear MLPs clearly show how the maximal stable learning rate scales as n^0 whereas the optimal learning rate scales as n^{-1} (Figure F.12). While activations can be shrunk for self-stabilization in linear nets, they lack the ability to learn non-linear features for the improved generalization at large learning rates. Hence losing feature learning but preventing the necessity to shrink activations under small learning rates n^{-1} is optimal. Observe that also deeper linear MLPs are only stable under $n^{-1/2}$ as theoretically predicted.

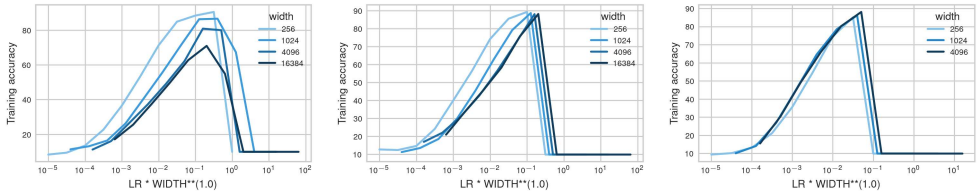


Figure F.10: **(MSE loss on MNIST approximately transfers under $\Theta(n^{-1})$ learning rate scaling)** Both the optimal as well as the maximal stable learning rate approximately transfer under global $\Theta(n^{-1})$ learning rate scaling when training 2, 3 or 10 layer MLPs (from left to right) with MSE loss on MNIST. Loss is not improving as feature learning is lost under $\Theta(n^{-1})$ scaling. Especially in 2 layer nets, the input layer is learning features at small width, but not at large width.

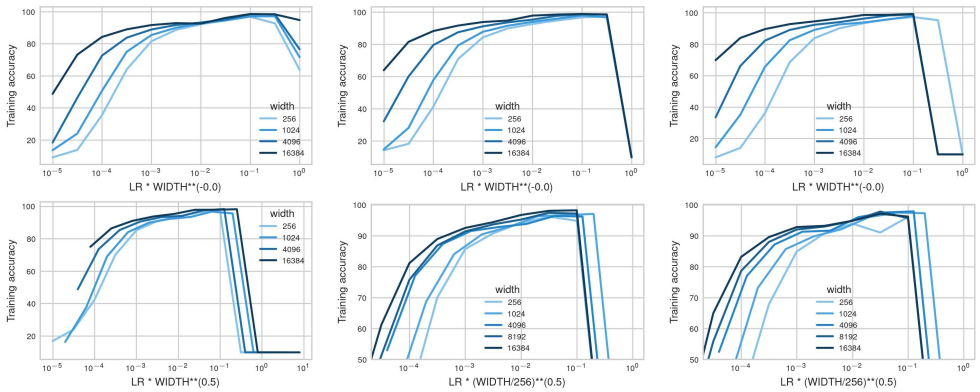


Figure F.11: **(Deeper nets follow infinite width theory increasingly accurately)** Training accuracy after 1 epoch of training MLPs with 2, 3, 4, 6, 8 and 10 layers (from top-left to bottom right) on MNIST. While 2, 3 and 4 layer MLPs self-stabilize under large learning rates $\Theta(n^0)$ and approximately transfer the optimum as well as max-stable learning rate, in 6, 8 and 10 layer MLPs it becomes increasingly apparent that the maximal stable learning rate transitions towards $\Theta(n^{-1/2})$ to prevent hidden layer blowup, which also forces the optimal learning rate to be $\mathcal{O}(n^{1/2})$ for at least feature learning in the hidden layers. Hence the theoretical activation stability predictions hold more accurately in deeper nets, with too many hidden layers to stabilize blowup in all of them.

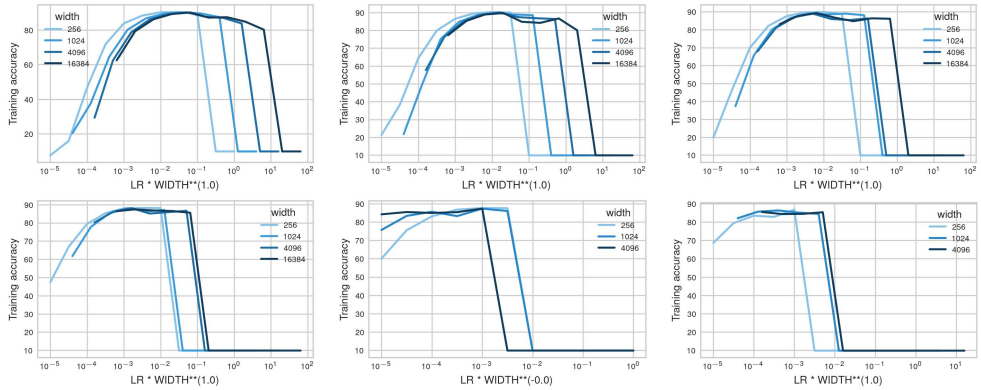


Figure F.12: **(In linear nets on MNIST, the optimal learning rate shrinks faster than the maximal stable learning rate)** Same as Figure F.11 but for linear nets. The maximal stable learning rate scales similarly as for the non-linear nets, but the optimum approximately follows $\Theta(n^{-1})$. Irrespective of the depth, linear MLPs can only learn a linear transformation; hence under sufficient width, feature learning under large learning rates does not provide a benefit over mere last-layer learning.

F.5 MLPs with ADAM on MNIST

Figure F.13 shows that for deep MLPs trained with Adam on MNIST the optimal learning rate scales at most as $\eta_n = \mathcal{O}(n^{-1})$.

MLPs with 2 or 3 layers tend to have larger optimal learning rate scaling exponents around $n^{-1/2}$, but with an increasing amount of layers the conflicting objectives of first layer versus hidden layer width-independent learning are dominated by the increasing number of hidden layers.

For all depths, the instability threshold appears to scale as $\eta_n \approx \Theta(n^{-1/2})$, but since Adam has a wide regime of suboptimal large learning rates where its moments are already harmed (**?**), the maximal stable learning rate threshold is often less clear cut compared to SGD.

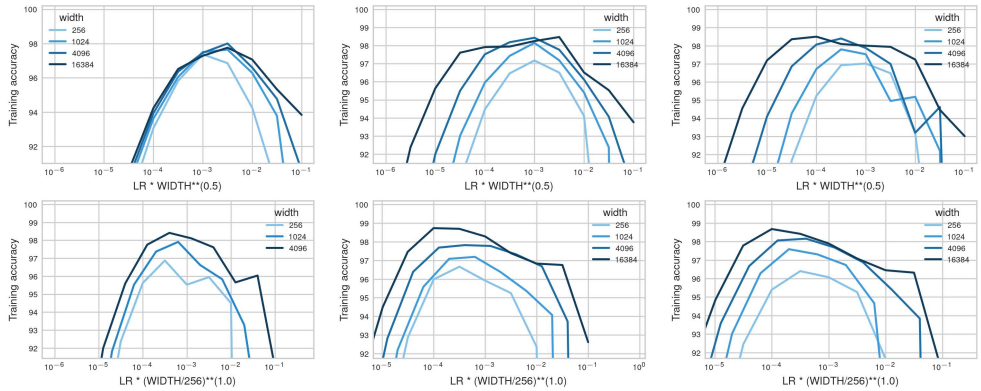


Figure F.13: **(Learning rate transfer in deep MLPs for ADAM on MNIST)** MLPs trained with ADAM on MNIST with 2, 3, 4, 6, 8, 10 layers (from top left to bottom right). In the first row, the x-axis is width-dependently scaled to show approximate transfer under $n^{-1/2}$ learning rate scaling. In the bottom row, the x-axis is width-dependently scaled to show approximate transfer under $\eta_n \approx \Theta(n^{-1})$. Observe the optimal learning rate scaling transitioning from larger than $\Theta(n^{-1/2})$ in 2-layer MLPs toward at most $\Theta(n^{-1})$ with increasing depth.

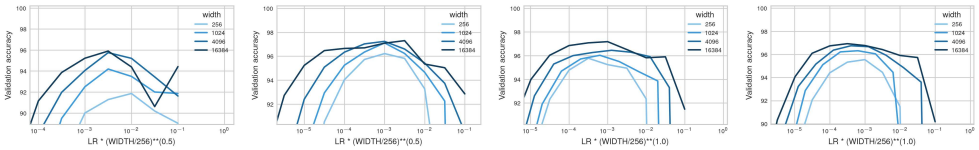


Figure F.14: **(Transfer in validation accuracy in deep MLPs for ADAM on MNIST)** Validation accuracy of MLPs trained with ADAM on MNIST with 2 layer random feature, 3, 8, 10 layers (from left to right). Validation-optimal learning rate in deep MLPs scales as $\eta_n = \mathcal{O}(n^{-1})$. 2 layer RF and 3 layer nets appear to approximately transfer under $\eta_n \approx \Theta(n^{-1/2})$ but lose monotonic improvement and predictability at scale.

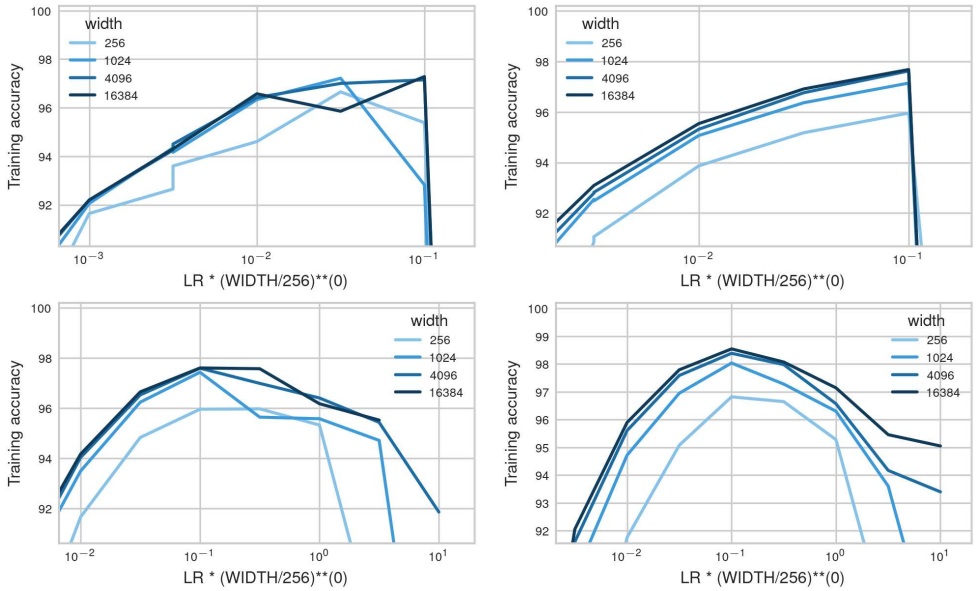


Figure F.15: **(μP as a baseline for transfer)** 8-layer MLPs trained on MNIST with SGD (top) and ADAM (bottom) under CE loss (left) and MSE loss (right). No systematic learning rate shifts in μP; saturating drifts may occur. Transfer and monotonic improvement looks less noisy under MSE loss.

F.6 MLPs with SGD on CIFAR-10

2-layer random feature ReLU MLPs very clearly transfer under $\eta_n = \eta \cdot n^{-1}$ learning rate scaling under any loss. Under CE loss, also larger learning rates result in non-trivial learning as saturating the softmax does not harm training stability. Under MSE loss on the other hand, training diverges above the edge of stability and results in trivial accuracy of 10%. Under MSE with softmax on the output logits, a exploding logits induce vanishing gradients which also inhibits learning (see Appendix F.3 for more details), and results in worse accuracy than under CE loss.

For 2-layer ReLU MLPs under CE loss, the maximal stable learning rate scales as $\eta_n = \Theta(1)$ as predicted under activation stability. The optimal learning rate however is a trade-off between input layer feature learning and output layer stability. Here an output layer that explodes too sharply appears to perform suboptimally. But under small learning rates $\eta_n = O(n^{-1/2})$, feature learning is lost and accuracy gets worse with scale.

For 3-layer RELU MLPs with CE loss, the maximal stable learning rate and the optimal learning rate transfer over many widths before beginning to shrink at width 16384. Over 782 updates, an initial catapult can be stabilized at moderate width as long as training does not diverge early such as under MSE loss. Apparently, large hidden layer updates can be stabilized over the course of training. As

an additional inductive bias that self-stabilizes large gradients, activations are sparsified which may enhance generalization under large learning rates.

Figure F.19 shows very slow decay of the optimal learning rate also in 4- and 6-layer MLPs, but deeper MLPs eventually require $\eta_n = \Theta(n^{-1/2})$ for stable hidden layer updates.

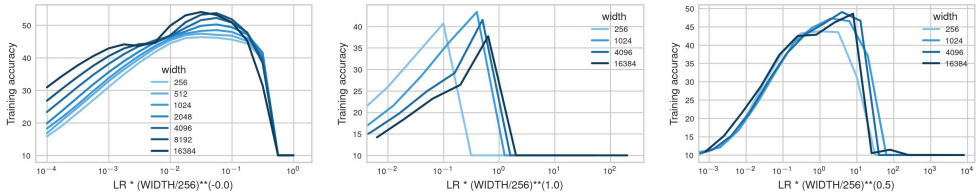


Figure F.16: **(Softmax increases maximal stable learning rate scaling)** Training accuracy after one epoch of training 3-layer MLPs on CIFAR 10 with CE loss (left), MSE loss (center) and MSE loss with softmax (right). The x-axis scales the learning rate with width-dependent exponents to show approximate transfer under $\Theta(1)$, $\Theta(n^{-1})$ and $\Theta(n^{-1/2})$ scaling, respectively.

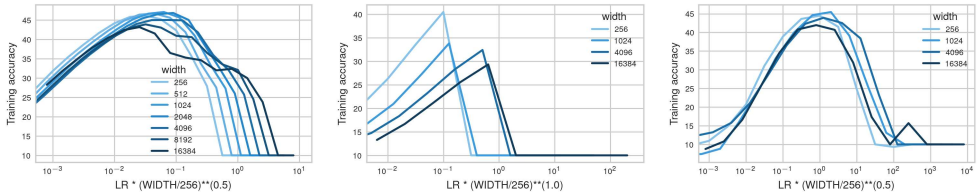


Figure F.17: **(Softmax increases maximal stable learning rate scaling)** Same as Figure F.16 but with 2-layer MLPs and approximate transfer under $\Theta(n^{-0.5})$, $\Theta(n^{-1})$ and $\Theta(n^{-1/2})$. Note how the maximal stable learning rate rather scales as $\Theta(1)$, but the optimal learning rate rather scales as $\Theta(n^{-1/2})$.

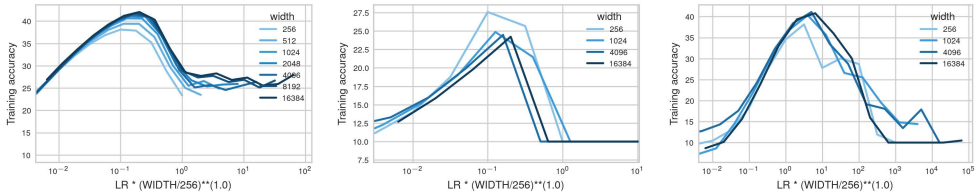


Figure F.18: **(Random feature models approximately transfer under $\eta_n = \Theta(n^{-1})$ for SGD)** Same as Figure F.16 but with 2 layers and only training the last layer results in approximately width-independent dynamics with $\eta_n = \eta \cdot n^{-1}$ independent of the loss function or architecture used. Note that also larger learning rates result in non-trivial generalization because there is no instability caused by activation blowup. The larger learning rates are not optimal, because the usual benefits of larger learning rates like increased feature learning or activation sparsity do not apply to random feature models.

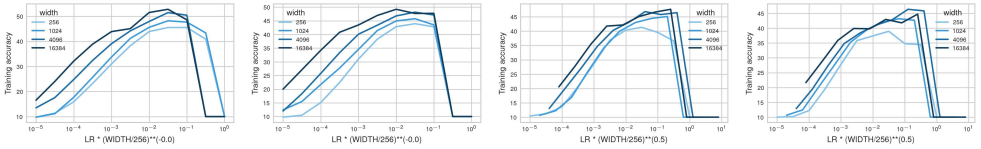


Figure F.19: **(Hidden-layer stability determines learning rate scaling in deep MLPs for SGD on CIFAR-10)** MLPs trained with SGD on CIFAR-10 with 4, 6, 8 and 10 layers (from left to right). First two x-axes are width-independent, last two scaled by $n^{1/2}$. While 4- and 6-layer MLPs self-stabilize sufficiently for approximate transfer under width-independent learning rate scaling, 8- and 10-layer MLPs have a clear max-stable learning rate scaling $n^{-1/2}$.

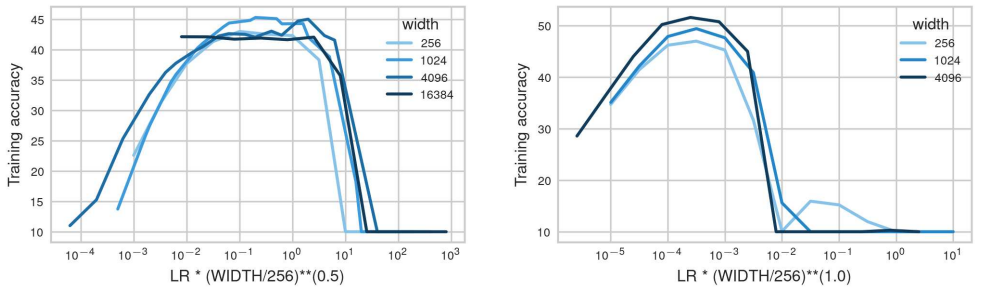


Figure F.20: **(Hidden-layer stability determines learning rate scaling in deep MLPs for SGD on CIFAR-10)** 8-layer MLPs trained with SGD (left) and Adam (right) on CIFAR-10 with MSE loss with Layernorm applied to the logits. The Layernorm has a similar stabilizing effect as CE loss and allows learning with logit blowup under $\eta_n = \Theta(n^{-1/2})$. For Adam, hidden and output layers learn width-independently with $\eta_n = \Theta(n^{-1})$.

F.7 MLPs with ADAM on CIFAR-10

All networks in Figure F.21 appear to transfer under large $\eta_n = \Theta(n^{-1/2})$ learning rate scaling. This scaling clearly induces activation blowup in the forward pass. A crucial difference to SGD is that activation blowup does not affect the updates in Adam since the gradient is normalized. For SGD, exploding gradients induce even larger explosion in the next forward pass, which in turn induces even larger explosion in the next backward pass. Hence, without activation stability, even the divergence exponent grows over time in SGD. For Adam, on the other hand, gradients are normalized, so that the forward pass always accumulates the same width-dependent exponent that is stabilized when passed through the softmax. Thus under sufficient numerical precision, from a stability point of view, Adam can even tolerate larger learning rates than the hidden-layer feature learning $\eta_n = \Theta(n^{-1})$, and the optimal learning rate may also be pushed toward input layer feature learning. Indeed, when fixing the first layer (Figure F.22), all MLPs transfer under $\eta_n = \Theta(n^{-1})$, which now achieves full width-independent effective updates. In this variant there are no conflicting objectives trading off hidden-layer and input-layer width-independence.

The fact that Adam with MSE loss (Figure F.23) can have large optimal learning rates $\eta_n = \Theta(n^{-1/2})$ indicates that the crucial effect of CE loss in SGD is stabilizing the gradients. Adam similarly limits the step size as the update scale is independent of χ_t. As for SGD, at large depth hidden-layer width-independence tends to dictate the optimal learning rate.

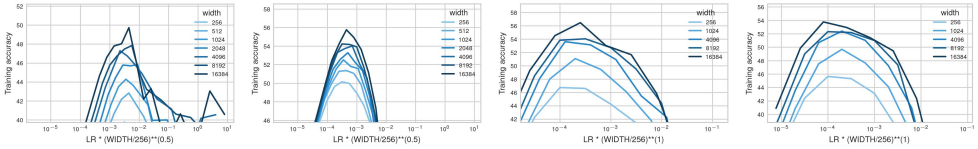


Figure F.21: **(Learning rate exponent $\eta_n = \Theta(n^{-1})$ for ADAM in deep MLPs on CIFAR-10)** MLPs trained with ADAM on CIFAR-10 with 2-layer random features, 2, 8, 10 layers (from left to right). The first 2 x-axes show approximate transfer under $n^{-1/2}$ learning rate scaling, the last 2 under n^{-1} learning rate scaling. As for SGD, in deeper nets hidden-layer width-independence dominates input-layer width-independence and induces optimal learning rate scaling $\eta_n \approx \Theta(n^{-1})$.

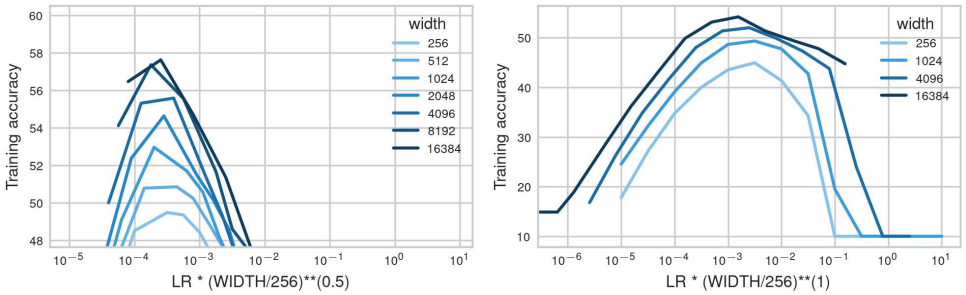


Figure F.22: **(Trade off between input- and hidden-layer width-independence)** 3-layer MLPs trained with ADAM on CIFAR-10 (left) and not training the first layer (right). 3-layer MLPs approximately transfer under $\eta_n = \Theta(n^{-1/2})$, being pushed toward input-layer feature learning. As there are no conflicting goals like preserving input layer feature learning, 3-layer MLPs with fixed input layer follow the width-independent exponent $\eta_n = \Theta(n^{-1})$ that yields hidden-and output-layer width-independent feature learning.

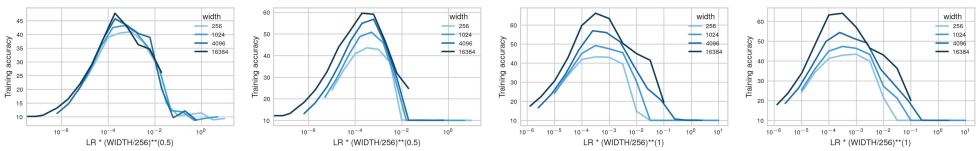


Figure F.23: **(Adam stabilizes the backward pass even under MSE loss)** MLPs trained with ADAM on CIFAR-10 under MSE loss with 2, 3, 6, 8 layers (from left to right). 2 and 3 layers show approximate transfer under $n^{-1/2}$ learning rate scaling, 6 and 8 layers show approximate transfer under n^{-1}.

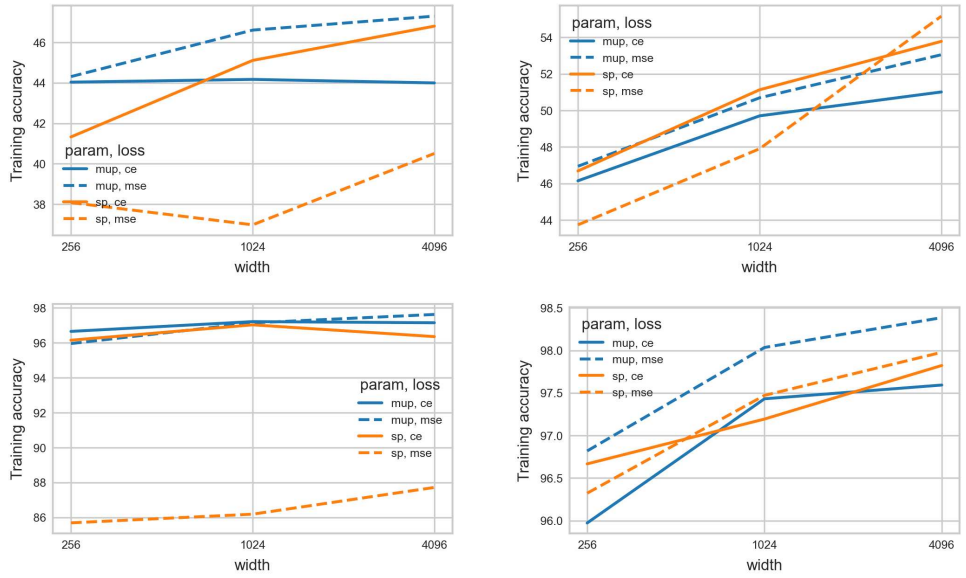


Figure F.24: **(Large performance difference between losses for SGD in SP)** Optimal training accuracy of 8-layer MLPs trained with SGD (left) and Adam (right) on CIFAR-10 (top) and MNIST (bottom) with MSE loss (dashed lines) and CE loss (solid lines) in μP (blue) and SP (orange). For SGD in SP, CE loss performs much better than MSE loss as large learning rates recover feature learning at large widths. The performance in μP depends much less on the loss function since features are always learned width-independently. In μP, MSE loss slightly outperforms CE loss. For ADAM, small learning rates $\eta_n = \Theta(n^{-1})$ in SP recover hidden-layer feature learning so that the difference between losses is much smaller.

F.8 Effective update parameterizations beyond μP

The logit updates can be decomposed into

$$f_t(\xi) - f_0(\xi) = W_0^{L+1}\Delta x_t^L(\xi) + \Delta W_t^{L+1} x_t^L(\xi),$$

for arbitrary inputs $\xi \in \mathbb{R}^{d_{in}}$ and $\Delta W_t^{L+1} = \sum_{t'=0}^{t-1} \chi_{t'} \cdot x_{t'}^L(\xi_{t'})$.

In this section, we consider vision and generated data sets in the regime $n \gg d_{\text{out}}$. First note that under large last-layer initialization $(W_0^{L+1})_{ij} \sim N(0, n^{-1})$ as in SP, fully width-independent training dynamics are impossible, since width-independent feature learning $\Delta x_t^L = \Theta(1)$ implies logit blowup through the term $W_0^{L+1} x_t^L = \Theta(n^{1/2})$ for both SGD and Adam. The fact that logit blowup does not prevent stable training under CE loss explains why we can achieve non-vanishing feature learning under SP last-layer initialization. When dropping the logit stability constraint, we can ask which is the optimal layerwise learning rate scaling under standard last-layer initialization. Following the μP desiderata, we still want to effectively update all layers, meaning a non-vanishing effect of the weight updates in each layer on the output function. With the correct choice of layerwise learning rates, we can still satisfy these desiderata for all scalings of last-layer initialization variance, which also implies that there is not a unique abc-equivalence class to fulfill these effective update desiderata when not requiring logit stability. We will see that SP full-align in Everett et al. (2024), which just uses the μP layerwise learning rates for SP initialization (which they promote as their overall best-performing parameterization without identifying stability under logit blowup as the key mechanism), fulfills these desiderata, except for vanishing last-layer update effect on the output function. We will introduce another variant with larger last-layer learning rate that recovers effective updates of all layers. For avoiding confusion with SP, meaning using a global learning rate, and with μP, meaning also achieving width-independence in the logits, we call this last variant *Maximal Update Parameterization under Standard Output-layer Initialization (MUSOLI)*.

For deriving the optimal layerwise learning rate exponents, first consider the scaling of hidden-layer pre-activation updates δh_l, $l \in [2, L]$, and input-layer pre-activation updates δh_1 (Yang and Hu, 2021, p. 51),

$$\delta h^l(\xi) = \Theta\Big(W_0^l \delta x_t^{l-1} + \eta_l \chi_{t-1} \frac{\partial f}{\partial h_{t-1}^l} \underbrace{(x_{t-1}^{l-1})^\top x_t^{l-1}(\xi))}_{n}\Big),$$

$$\delta h^1(\xi) = \Theta\Big(\eta_1 \chi_{t-1} \frac{\partial f}{\partial h_{t-1}^1} \underbrace{(\xi_{t-1})^\top \xi)}_{1}\Big),$$

where it holds that $\partial f / \partial h^l = \Theta(\partial f / \partial x^L) = W_t^{L+1} = \Theta(W_0^{L+1} - \eta_{L+1} \chi_t x^L)$ (at latest in the second step) (Yang and Hu, 2021, p. 52). Hence the correct l-th layer learning rate η_l for achieving a width-independent effect on the next layer's pre-activations needs to cancel out the backpropagated gradient scaling $\partial f / \partial h^l$ and for hidden layers additionally the LLN-like scaling from the inner product between activations. As we still require activation stability $x_T^L = \Theta(1)$, we have $\partial f / \partial x^L = \Theta(n^{-\min(b_{L+1}, c_{L+1})})$. While under standard μP, it holds that $\partial f / \partial x^L = \Theta(n^{-1})$, the changed gradient scaling must be counteracted by choosing hidden layer learning rate $\eta_l = \Theta(n^{\min(b_{L+1}, c_{L+1})-1})$, $l \in [2, L]$, and input layer learning rate $\eta_1 = \Theta(n^{\min(b_{L+1}, c_{L+1})})$. In words, under larger last-layer initialization or learning rate, the hidden and input layer learning rates should be scaled down by the same amount. Finally, SP-full-align achieves a width-independent effect of the last-layer weight updates on the logits. But as the width-independent feature updates $\Delta x^L = \Theta(1)$ induce logit blowup $W_0^{L+1} \Delta x_t^L = \Theta(n^{1/2})$, the effect of the last-layer weight updates on the softmax output is actually vanishing. For last-layer weight updates to affect the softmax output in the same scaling as the updates propagated forward, the last-layer learning rate needs to be $\eta_{L+1} = \Theta(n^{-b_{L+1}})$, hence $c_{L+1} = b_{L+1}$. Hence MUSOLI is defined as SP-full-align but setting $\eta_{L+1} = \Theta(n^{-b_{L+1}})$. This last-layer learning rate is larger than in μP or Everett et al. (2024) under standard last-layer initialization $b_{L+1} = 1/2$, but necessary for fulfilling the desideratum that the weight updates in all layers affect the output function non-vanishingly.

Figure F.34 shows that indeed all weight updates behave width-independently under μP with standard last-layer initialization $(W_0^{L+1})_{ij} \sim N(0, n^{-1})$. But the output logits are dominated by the activations propagated forward as $W_0^{L+1} \delta x_t^L = \Theta(n^{1/2})$, since δx_t^L and W_0^{L+1} are highly correlated. Consequently, the last-layer updates have vanishing effect on the output function, which induces width dependence. By additionally scaling up the last-layer learning rate $\eta_{L+1} = \Theta(n^{-1/2})$, the logit scaling exponent in the term $W_0^{L+1} \delta x_t^L = \Theta(n^{1/2})$ is matched in the last-layer update term $\Delta W_t^{L+1} x_t^L = \Theta(n^{1/2})$ so that $b_{L+1} = 1/2$ and $c_{L+1} = 1/2$ recovers a balanced influence of all layer updates in the softmax output.

Figure F.25 and Figure F.26 show that after single-pass SGD or Adam, for both SP-full-align and MUSOLI the optimal learning rate shrinks with width for both generated 2-class multi-index teacher data as well as MNIST. The optimal learning rate exponent is often closer to -0.5 as we consistently observe under MSE loss, preventing logit blowup. Figure F.27 shows the same for CIFAR-10. This behaviour persisting across 3 data sets suggests that neither SP-full-align nor MUSOLI can be expected to transfer the optimal learning rate in general. An interesting question for future work remains why logit divergence introduces a width-dependence in the optimal learning rate in these parameterizations.

As expected from parameterizations in the controlled divergence regime, Figure F.27 also shows that the maximal stable learning rate scales width-independently, since activation and gradient stability is preserved. Over the course of 20 epochs, the training dynamics under large learning rates in MLPs with at least 3 layers are stabilized and the optimal learning rate indeed scales width-independently under standard last-layer initialization. Hence width-dependence in parameterizations can induce optimal learning rate scaling that varies over the course of long training. But often the optimal learning rate scales like the maximal stable learning rate. In such cases our theory is predictive. Note that SP full-align and MUSOLI are much more robust to poor tuning of the learning rate than μP, both in terms of training and test accuracy (Figures F.28 and F.29). We leave a closer analysis of the multi-epoch setting to future work.

For ADAM, the gradient is normalized in the backward pass, so that input- and hidden-layer learning rates remain the same as in μP under large last-layer initialization. This is again equivalent to the SP full-align parameterization from Everett et al. (2024). The logit update term $W_0^{L+1}\Delta x_t^L = \Theta(n^{1/2})$ should again be balanced with a larger output layer learning rate $\eta_{L+1} = \Theta(n^{-1/2})$ if the weight updates of all layers should have a non-vanishing effect on the softmax output in the infinite-width limit (MUSOLI). Figure F.30 shows that nonlinear networks trained with Adam and large last-layer initialization already tend to transfer better under MUSOLI than under SP full-align after 1 epoch. Linear networks again have smaller optimal learning rate exponent, indicating that avoiding logit blowup improves over feature learning in this case, where feature learning does not even add expressivity. Generalization, learning rate transfer and learning rate sensitivity after 20 epochs tends to be similar in all 3 considered parameterizations in deep ReLU MLPs (Figure F.31), showing again that parameterizations with logit blowup are a viable alternative.

Especially in deep ReLU MLPs, the last-layer learning rate does not seem to have a big impact, and SP full-align and MUSOLI overall behave similarly for both SGD and Adam.

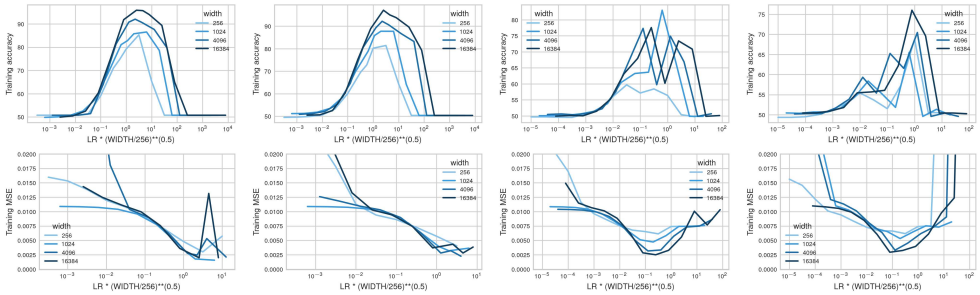


Figure F.25: **(Effective update variants do not transfer optimal learning rates on multi-index data)** Training accuracy of 8-layer MLPs trained for 1 epoch on multi-index teacher data under CE loss (top) and MSE loss (bottom) with SGD in SP-full-align, SGD in MUSOLI, Adam in SP-full-align and Adam in MUSOLI (from left to right). In all cases, logit blowup is avoided by optimal learning rates shrinking as $\eta_n = \Theta(n^{-1/2})$. Under CE loss the maximal stable learning rate remains width-independent, for SGD under MSE loss the maximal stable learning rate decays as $n^{-1/2}$, as necessary for stability.

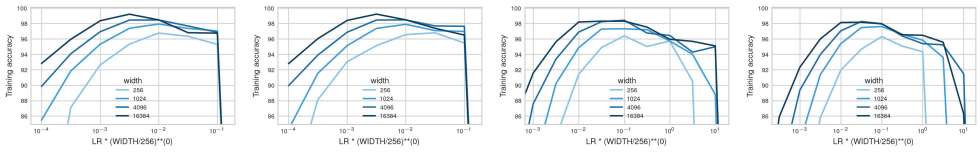


Figure F.26: **(Effective update variants do not transfer optimal learning rates on MNIST)** Training accuracy of 8-layer MLPs trained for 1 epoch on MNIST under CE loss with SGD in SP-full-align, SGD in MUSOLI, Adam in SP-full-align and Adam in MUSOLI (from left to right). In all cases, the optimal learning rate decays with width, while the maximal stable learning rate stays constant.

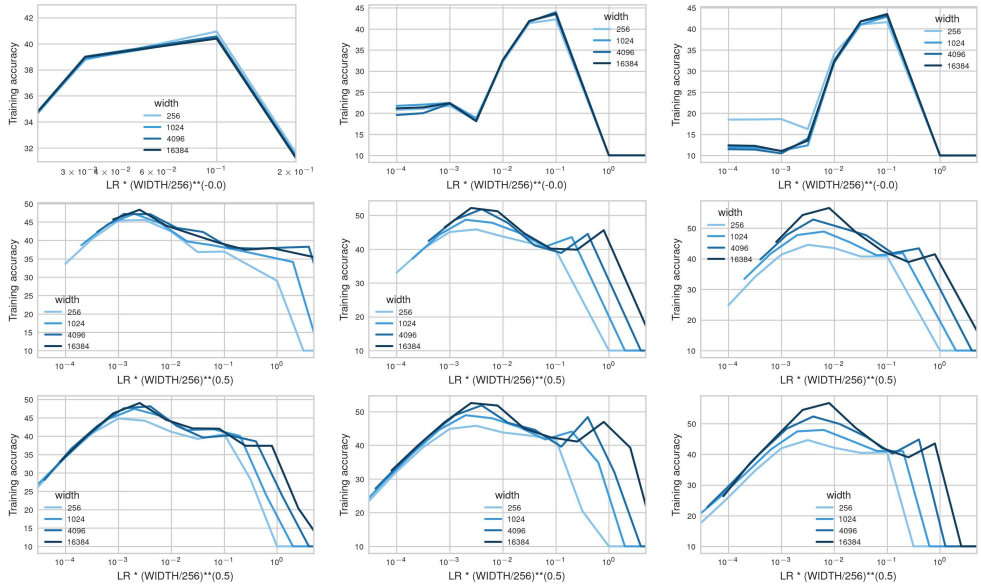


Figure F.27: **(Effective update variants for SGD on CIFAR-10)** MLPs with 2, 3 an 6 layers (from left to right) trained with SGD on CIFAR-10 in μP (top) versus SP full-align from Everett et al. (2024) (2nd row) versus SP full-align with larger last-layer learning rate (MUSOLI) (bottom row). While μP transfers with low variance as expected (left), μP with large standard last-layer initialization $b_{L+1} = 1/2$ and large last-layer learning rate $c_{L+1} = 1/2$ (right) have a non-trivial optimal learning rate scaling between $\Theta(n^{-1/2})$ and $\Theta(1)$, while the maximal stable learning rate scales width-independently.

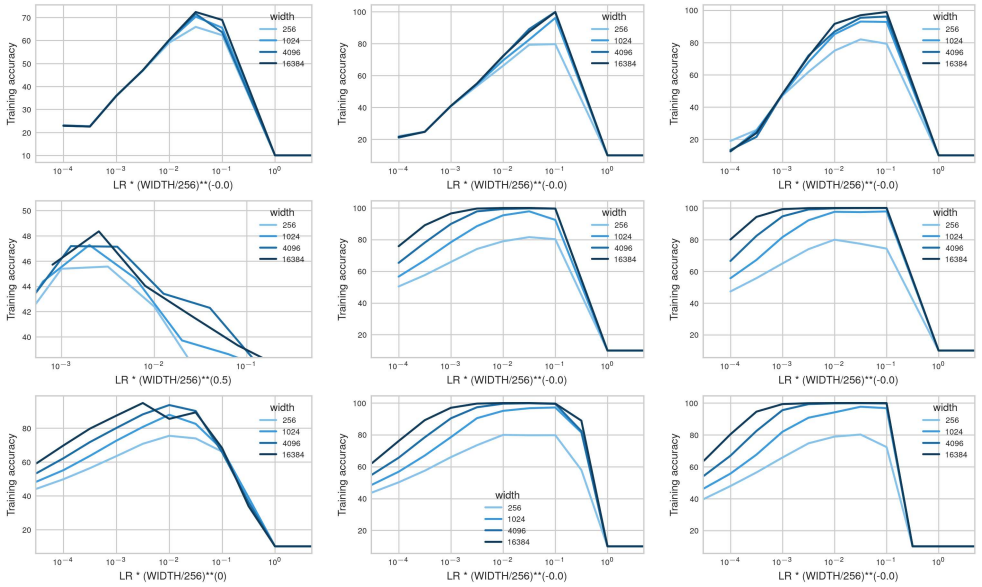


Figure F.28: **(Effective update variants for SGD on CIFAR-10 after convergence)** MLPs with 2, 3 an 6 layers (from left to right) trained with SGD in μP (top) versus SP full-align from Everett et al. (2024) (2nd row) versus SP full-align with larger last-layer learning rate (MUSOLI) (bottom row) as in Figure F.27 but trained for 20 epochs. After sufficiently long training the large learning rate dynamics stabilize in MUSOLI so that the optimum indeed scales width-independently. MUSOLI strictly dominates original μP in training accuracy, and robustness to badly tuned learning rate is strongly improved under SP last-layer initialization compared to original μP. In sufficiently deep MLPs, the larger last-layer learning rate barely matters, but in 2-layer nets SP-full align avoids output blowup and feature learning by transferring under $\eta_n = \Theta(n^{-1/2})$.

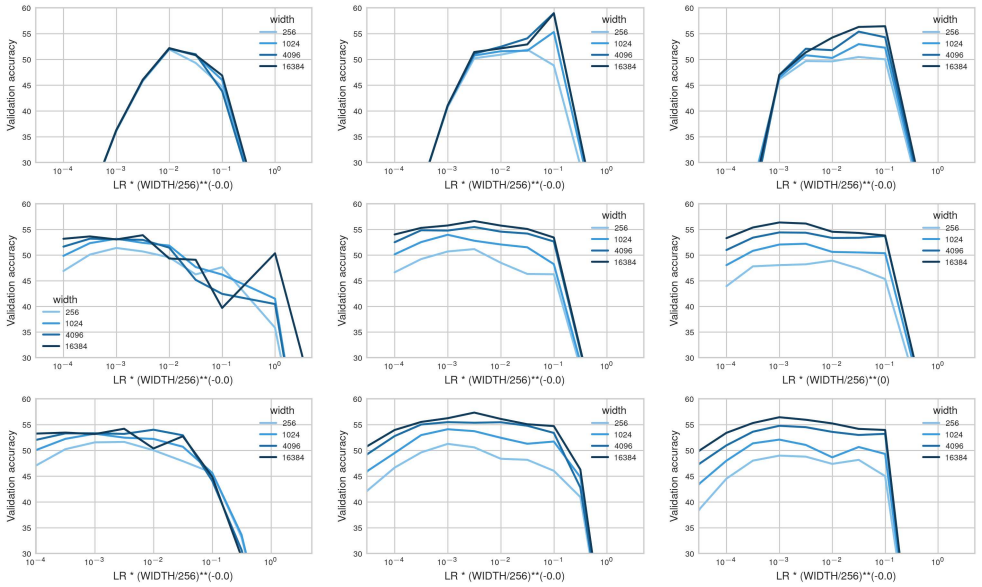


Figure F.29: **(Test accuracy of effective update variants for SGD on CIFAR-10 after convergence)** Test accuracy of 2-layer, 3-layer and 6-layer (from left to right) MLPs trained with SGD for 20 epochs on CIFAR-10 in μP (top) versus SP full-align from Everett et al. (2024) (2nd row) versus SP full-align with larger last-layer learning rate (MUSOLI) (bottom row). The validation-optimal learning rate scales width-independently in all cases. Observe that, while all variants generalize similarly well, the susceptibility to poorly tuned learning rates is much larger in μP than under parameterizations with large last-layer initialization.

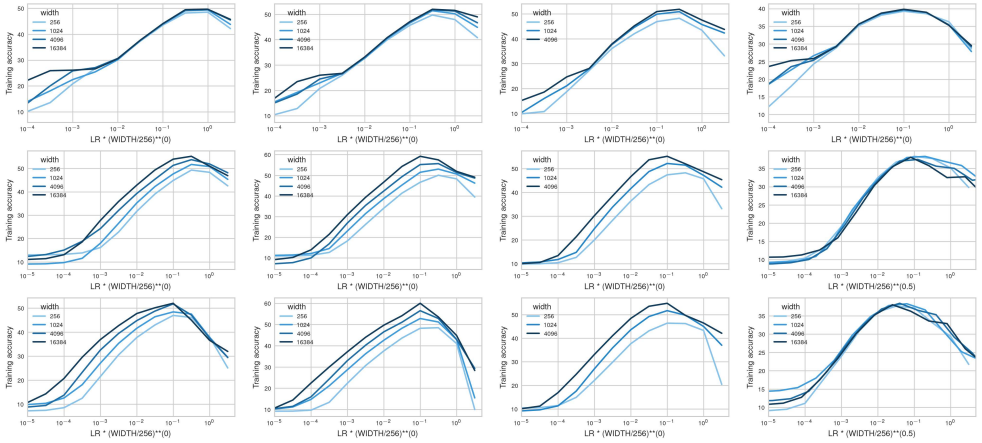


Figure F.30: **(Train accuracy of effective update variants for ADAM on CIFAR-10)** Train accuracy of 2-layer, 3-layer, 6-layer and 3-layer-linear MLPs (from left to right) trained with ADAM for 1 epochs on CIFAR-10 in μP (top row) versus SP full-align from Everett et al. (2024) (2nd row) versus MUSOLI (bottom row). The learning rate transfers irrespective of the architecture in μP. Large last-layer learning rate improves transfer in MUSOLI over SP full-align. The optimal learning rate scales as $\eta_n = \Theta(n^{-1/2})$ in both parameterizations with large last-layer initialization, as feature learning does not improve expressivity.

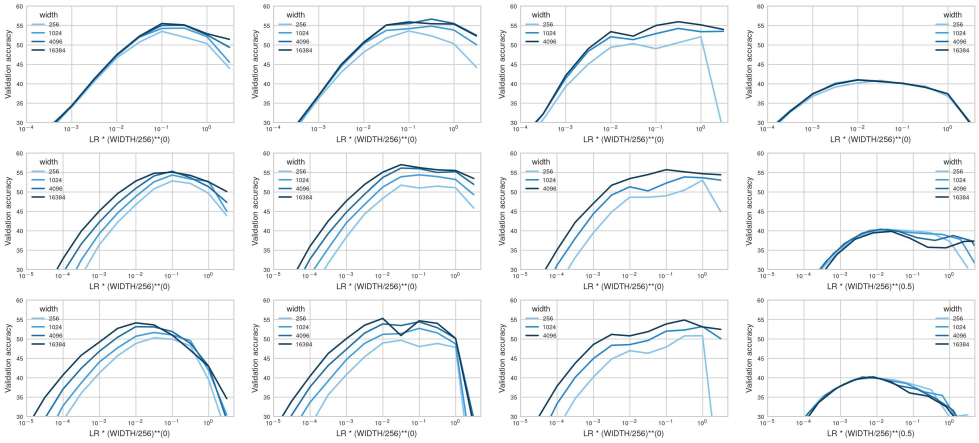


Figure F.31: **(Test accuracy of effective update parameterizations for ADAM on CIFAR-10 after convergence)** Test accuracy of 2-layer, 3-layer, 6-layer and 3-layer-linear MLPs (from left to right) trained with ADAM for 20 epochs on CIFAR-10 in μP (top row) versus SP full-align from Everett et al. (2024) (2nd row) versus MUSOLI (bottom row). The validation-optimal learning rate scales width-independently in all ReLU MLPs with at least 3 layers. 3-layer linear networks clearly transfer under $\eta_n = \Theta(n^{-1/2})$ in SP full-align and MUSOLI, as for sufficient width learning features does not add expressivity, and instead avoiding logit blowup dominates the learning rate scaling.

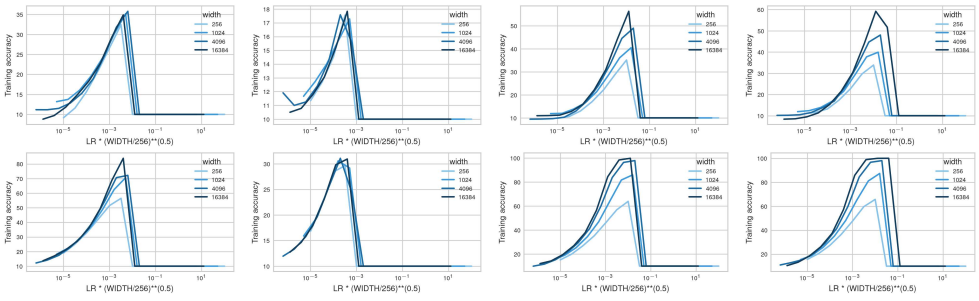


Figure F.32: **(Effective update variants with SGD under MSE loss avoid logit blowup)** Training accuracy of 2-layer, 3-layer linear, 6-layer and 8-layer MLPs (from left to right) trained with SGD for 1 epoch (top) and 20 epochs (bottom) on CIFAR-10 in SP full-align from Everett et al. (2024). Optimal learning rates shrinking as $\eta_n = \Theta(n^{-1/2})$ persists, avoiding logit blowup through $W_0^{L+1}\Delta x_t^L$. Only in 8-layer MLPs is the optimal learning rate saturating at the width-independent stability threshold.

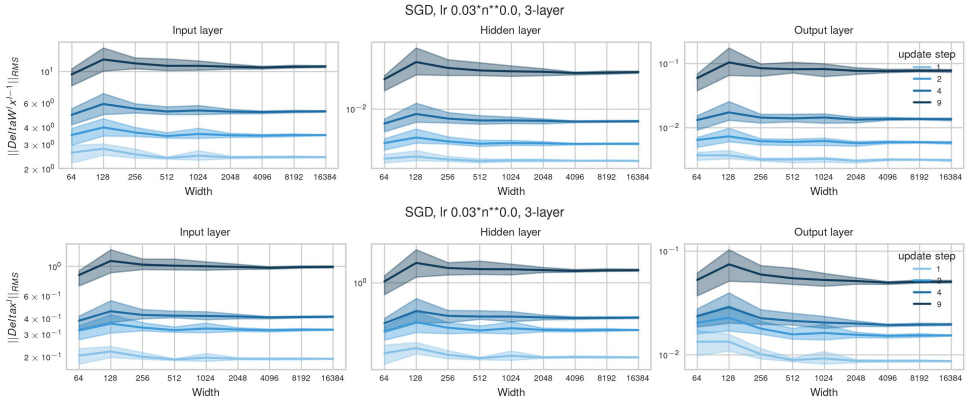


Figure F.33: **(Coordinate check for μP for SGD on CIFAR-10)** μP induces fully width-independent update dynamics.

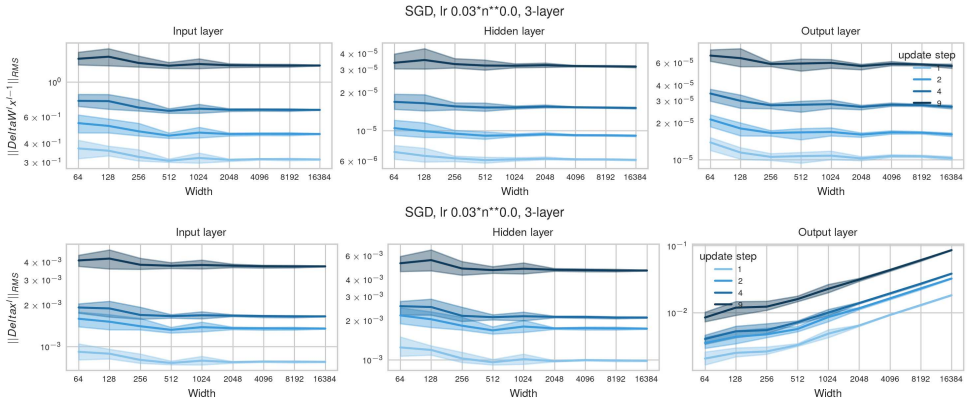


Figure F.34: **(Coordinate check for μP with standard initialization for SGD on CIFAR-10)** Effective updates $\|\Delta W^l x^{l-1}\|_{RMS}$ and activation updates $\|\Delta x^l\|_{RMS}$ as a function of width. The theoretically predicted scaling exponents hold: All layers update width-independently, but due to the large last-layer initialization, the activation updates correlated with W_0^{L+1} propagated forward induce output logits exploding as $W_0^{L+1}\delta x_t^L = \Theta(n^{1/2})$. This motivates increasing the last-layer learning rate to $\eta_{L+1} = \Theta(n^{-1/2})$ so that last-layer updates contribute with the same scaling. Note that in absolute terms, the updates are much smaller than under μP (Figure F.33).

NeurIPS Paper Checklist

1. **Claims**
   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?
   Answer: [Yes]
   Justification: In the abstract and introduction, we tried to summarize and contextualize our contributions while properly acknowledging related work to the best of our ability.
   Guidelines:
   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.
2. **Limitations**
   Question: Does the paper discuss the limitations of the work performed by the authors?
   Answer: [Yes]
   Justification: We clearly state the caveats of every reported result. We mostly discuss limitations and avenues for future work in Section 6.
   Guidelines:
   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.
3. **Theory assumptions and proofs**
   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?
   Answer: [Yes]
   Justification: We detail all assumptions and the proofs of our theoretical results in Appendix C.
   Guidelines:
   - The answer NA means that the paper does not include theoretical results.

- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**
   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?
   Answer: [Yes]
   Justification: We detail all necessary information in the main paper and Appendix D.
   Guidelines:
   - The answer NA means that the paper does not include experiments.
   - If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**
   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?
   Answer: [Yes]
   Justification: Upon acceptance, we will release open source code to reproduce all of our experiments on GitHub.
   Guidelines:
   - The answer NA means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.

- While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (https://nips.cc/public/guides/CodeSubmissionPolicy) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**
   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?
   Answer: [Yes]
   Justification: We detail all necessary information in the main paper and Appendix D.
   Guidelines:
   - The answer NA means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.
7. **Experiment statistical significance**
   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?
   Answer: [Yes]
   Justification: If not stated otherwise, error bars denote the region between the empirical 0.025- to 0.975-quantiles from 4 random seeds (affecting initial weights and data shuffling/generation). Due to computational constraints, we only train neural networks on several random seeds for a subset of the experiments.
   Guidelines:
   - The answer NA means that the paper does not include experiments.
   - The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).
   - It should be clear whether the error bar is the standard deviation or the standard error of the mean.
   - It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
   - For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
   - If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.
8. **Experiments compute resources**

Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?
Answer: [No]
Justification: Single training runs of 8-layer MLPs of width 16384 including tracking all relevant statistics as well as our 1.4B GPT model of width 4096 run within less than 24 hours on a single Nvidia A100 GPU. We typically trained MLPs up to width 4096 on a single Nvidia Geforce Rtx 2080 Ti within less than 24 hours.
Guidelines:

- The answer NA means that the paper does not include experiments.
- The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
- The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
- The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**
Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics https://neurips.cc/public/EthicsGuidelines?
Answer: [Yes]
Justification: The research in this paper conforms in every respect with the NeurIPS Code of Ethics.
Guidelines:
   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**
Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?
Answer: [Yes]
Justification: This paper aims to advance the understanding of standard deep learning practice. We do not foresee any societal impacts beyond the dual use considerations that apply to the whole field of machine learning research.
Guidelines:
   - The answer NA means that there is no societal impact of the work performed.
   - If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
   - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
   - The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
   - The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
   - If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**
    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?
    Answer: [NA]
    Justification: This paper studies pre-existing training procedures on established datasets. We do not release any data or high-risk models.
    Guidelines:
    - The answer NA means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.
12. **Licenses for existing assets**
    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?
    Answer: [Yes]
    Justification: We cite the standard CIFAR10 (Krizhevsky et al., 2009), MNIST (Deng, 2012) and DCLM-Baseline (Li et al., 2024) datasets following the standard practice. We also cite the Python assets PyTorch (Paszke et al., 2019) and LitGPT (Lightning AI, 2023) that we use as a basis for our experiments.
    Guidelines:
    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
    - If this information is not available online, the authors are encouraged to reach out to the asset's creators.
13. **New assets**
    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?
    Answer: [NA]
    Justification: The paper does not release new assets.
    Guidelines:
    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.
14. **Crowdsourcing and research with human subjects**

Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?
Answer: [NA]
Justification: The paper does not involve crowdsourcing nor research with human subjects.
Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**
Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?
Answer: [NA]
Justification: The paper does not involve crowdsourcing nor research with human subjects.
Guidelines:
    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.
16. **Declaration of LLM usage**
Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigorousness, or originality of the research, declaration is not required.
Answer: [NA]
Justification: LLMs were not used in a way that impacts the core methodology, scientific rigorousness, or originality of the research.
Guidelines:
    - The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy (https://neurips.cc/Conferences/2025/LLM) for what should or should not be described.

Part III

Discussion

Chapter 7

Discussion

In this thesis, we have contributed toward understanding how trained neural networks behave at increasing width. In Chapter 4, we have shown that overfitting of standard neural networks in NTP at extensive width is harmful in fixed data dimension, but that adding small-amplitude, high-frequency oscillations to the activation function can induce a spiky-smooth inductive bias that interpolates noisy training labels while achieving rate-optimal generalization. This shows that overfitting in large models is neither intrinsically helpful nor harmful for generalization in arbitrary fixed dimension. In Chapter 5, we have derived a layerwise scaling rule μP^2 of initialization variances, learning rates and perturbation radii for Sharpness Aware Minimization that induces width-independent training dynamics and allows the joint transfer of the learning rate and perturbation radius from small to large model scale, as opposed to μP without layerwise perturbation scaling. In Chapter 6, we have narrowed the gap between infinite-width theory and deep learning practice, by proving that qualitative properties such as training stability and feature learning of standard width scaling with optimal learning rate selection at each model scale can even be captured in the infinite width limit, when accounting for the stabilizing effect introduced by `torch.nn.CrossEntropyLoss`. In our experiments, we observe that the stability constraints that confine the controlled divergence regime, where training remains stable despite training divergence, are particularly predictive of the optimal learning rate scaling law as a function of width in deep, sensitive architectures such as Transformers.

We see the potential for exciting and important future work in two main directions: predictable scaling laws and generalization.

7.1 Predictable Scaling Laws

A physics approach to deep learning theory. While our exact understanding of neural network learning dynamics is still limited to toy architectures with restrictive distributional assumptions (Kunin et al., 2024; Zhang et al., 2025; Tsigler et al., 2025; Ren et al., 2025), the research community has made progress by identifying several phenomena that appear consistently across architectures, training procedures and data sets, such as a lack of robustness, learning of transferable representations, or learning rate optimality at the edge of stability. As in physics, a continual development of theoretical understanding on simplified settings paired with extensive empirical

evaluation in practical settings continues to deepen our collective understanding of the internal mechanisms that drive the success of deep learning. Similar to temperature in thermodynamic particle systems, can we find important summary statistics of neural networks that describe their current state and allow to predict how training evolves?

The observations by Kaplan et al. (2020) and Hoffmann et al. (2022) suggest that at least the dependence of the test risk on important scaling dimensions such as dataset size and model size approximately simplifies to separate scaling laws, which suggests that understanding these dependencies macroscopically may be feasible. As an important practical benefit, such scaling laws can be extrapolated for maintaining a compute-optimal trade-off between model and dataset size.

Width-dependent scaling. For width dependence, Tensor Program-based width-scaling arguments have shown surprising predictive accuracy even at moderate width and over the course of training (Vyas et al., 2024; Noci et al., 2024b; Chapter 5). In Chapter 6, we have shown that this even holds in parameterizations that do not induce fully width-independent training dynamics like SP with optimal learning rates, as opposed to the NTK model in the kernel regime. Similar observations have been made with respect to depth scaling (Bordelon et al., 2024c,b). This shows that, even in deep large-scale Transformer models, the scaling dependence with respect to important dimensions such as width and depth can simplify sufficiently to enable general, principled scaling prescriptions with impactful downstream consequences on generalization and predictability. For the example of NTP, activation updates vanish as a function of width. Such vanishing feature learning is often associated with suboptimal generalization. In μP, the SGD learning rate scales predictably as a function of width, so that it can be transferred from small to large scale (Yang et al., 2022). We have shown in Chapter 6 that this can even hold in SP where input-layer feature learning vanishes and logits diverge. This shows that the conditions that induce fast convergence of important hyperparameters to leading order are still poorly understood. It also remains unclear whether full width independence is always optimal, or whether logit divergence can sometimes induce improved performance or learning speed as training data is quickly memorized. Logit divergence may partially explain overconfidence in SP and suggests that networks in μP may be more calibrated. On the technical side, even though the NTK diverges in the controlled divergence regime, a rescaled NTK may still enable more detailed studies of the training dynamics in this infinite-width regime. Deeper understanding of the controlled divergence regime is paramount, as it captures the practically dominant large scale network training.

Stability and numerical considerations. When scaling sums, diverging or vanishing intermediate signals are unavoidable. Analyses that take numerical considerations into account in detail are essential for predictable and optimal practical performance at large scale (Blake et al., 2025). Not only `torch.nn.CrossEntropyLoss` but also normalization layers and Adam correct many misscaled signals in the forward and backward pass. Better understanding their interplay might further improve optimal learning rate predictability. Designing architecture and optimization components that guarantee automatic correct scaling could facilitate and robustify the wide-spread use of deep learning at all scales.

Joint scaling rules. Beyond width scaling, principled scaling rules with respect to training time and batch size constitute an important and challenging direction

to explore that may require more specific data- and architecture-dependent analyses. Ideally, joint scaling rules of all important scaling dimensions would allow finding a globally optimal trade-off between width, depth, dataset size and training time. Further improvements may be achieved by optimizing the order or mixture in which data is presented to the neural network.

7.2 Generalization

Classical notions of generalization. Generalization is often posed as the ultimate goal of machine learning. Traditionally this means accurate predictions on an unseen test set from the same underlying data distribution in classification or regression. In relation to our work in Chapter 4, the question remains open, whether there exist spiky-smooth inductive biases that improve generalization in practical feature learning architectures and on real world data. If so, which methods beyond adapting the activation function may enhance this behaviour? On the other hand, overfitting to noisy labels necessarily harms robustness. Simon et al. (2024) have partially answered in which cases overfitting is desirable over regularization for random feature regression, but addressing this question for feature learning models would yield closer correspondence to practical neural networks and inform the optimal choice of training procedure. What is the optimal amount of feature learning and which optimization procedure learns the optimal features? Are there inductive biases to prevent overfitting in large models even after multi-epoch training?

Stronger notions of generalization. With the increasing capabilities of AI systems, the research community is increasingly interested in more powerful generalization such as the identification of causal mechanisms or good performance on unseen tasks. The formal understanding of such generalization properties of complex neural networks remains much more elusive. Here, theory still has the opportunity to lay conceptual foundations for creating a useful language and enabling informed discussions about these higher goals. Similar conceptual groundwork, formal guarantees and provable limitations are missing for fuzzier but not less important goals related to trustworthiness.

Data structure. Ultimately the answer to most of the above questions is data-dependent. While there are initial attempts of capturing hierarchies (Dandi et al., 2025), symmetries (Gupta et al., 2024) and manifold structure (Goldt et al., 2020), much more effort should be invested in finding the right structural assumptions to accurately describe modern data modalities and tasks such as general-purpose chats that are interactive, multi-modal, may involve the use of tools and may cover arbitrary tasks that can be formulated in natural language. Höltgen and Williamson (2024) go as far as questioning the use of probability distributions in the context of machine learning research.

7.3 Concluding remarks

The overarching goal in this thesis was to develop theoretical understanding of artificial neural network training that is also relevant in practice, either by understanding limitations or by providing corrections that result in better generalization or more reliable scaling properties. We hope to inspire more researchers to pursue such

principled deep learning research that bridges theory and practice. In the best case, deep understanding may even enable to replace the large and expensive networks entirely by more efficient methods that inherit the inductive biases at limit behaviour with cheaper training and inference cost. With the increasing scale at which resources and effort are invested into machine learning research, our collective understanding of large neural networks will undoubtedly continue to grow in exciting and sometimes unexpected ways.

License Information

Chapter 8

Bibliography

Simona Abis and Laura Veldkamp. The changing economics of knowledge production. *The Review of Financial Studies*, 37(1):89–118, 2024. Cited on page 3.

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. Gpt-4 technical report. *arXiv:2303.08774*, 2023. Cited on page 3.

Elisabeth Agoritsas, Giulio Biroli, Pierfrancesco Urbani, and Francesco Zamponi. Out-of-equilibrium dynamical mean-field equations for the perceptron model. *Journal of Physics A: Mathematical and Theoretical*, 51(8):085002, 2018. Cited on page 17.

A Aizerman. Theoretical foundations of the potential function method in pattern recognition learning. *Automation and remote control*, 25:821–837, 1964. Cited on page 9.

Sam Altman. Planning for agi and beyond. *OpenAI website, see https://openai.com/index/planning-for-agi-and-beyond/*, 2023. Accessed: June 5th, 2025. Cited on page 3.

Maksym Andriushchenko and Nicolas Flammarion. Towards understanding sharpness-aware minimization. In *International Conference on Machine Learning (ICML)*, 2022. Cited on page 31.

Maksym Andriushchenko, Francesco Croce, Maximilian Müller, Matthias Hein, and Nicolas Flammarion. A modern look at the relationship between sharpness and generalization. In Andreas Krause, Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato, and Jonathan Scarlett, editors, *Proceedings of the 40th International Conference on Machine Learning (ICML)*, 2023a. Cited on page 17, 29, 34.

Maksym Andriushchenko, Aditya Vardhan Varre, Loucas Pillaud-Vivien, and Nicolas Flammarion. Sgd with large step sizes learns sparse features. In *International Conference on Machine Learning (ICML)*, 2023b. Cited on page 34.

Nachman Aronszajn. Theory of reproducing kernels. *Transactions of the American mathematical society*, 68(3):337–404, 1950. Cited on page 9.

Sanjeev Arora, Simon S. Du, Wei Hu, Zhiyuan Li, Russ R. Salakhutdinov, and Ruosong Wang. On exact computation with an infinitely wide neural net. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2019. Cited on page 15, 26, 27, 33.

Alexander Atanasov, Alexandru Meterez, James B Simon, and Cengiz Pehlevan. The optimization landscape of SGD across the feature learning strength. In *The Thirteenth International Conference on Learning Representations (ICLR)*, 2025. Cited on page 34.

Jean-Yves Audibert and Alexandre B. Tsybakov. Fast learning rates for plug-in classifiers. *The Annals of Statistics*, 35(2):608 – 633, 2007. Cited on page 27.

Gregor Bachmann, Sotiris Anagnostidis, and Thomas Hofmann. Scaling mlps: A tale of inductive bias. *Advances in Neural Information Processing Systems (NeurIPS)*, 36: 60821–60840, 2023. Cited on page 5.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. *arXiv:1409.0473*, 2014. Cited on page 4.

Yasaman Bahri, Ethan Dyer, Jared Kaplan, Jaehoon Lee, and Utkarsh Sharma. Explaining neural scaling laws. *Proceedings of the National Academy of Sciences*, 121 (27):e2311878121, 2024. Cited on page 5.

Isabel Barbera. Ai privacy risks and mitigations - large language models (llms). *European Data Protection Board, see https://www.edpb.europa.eu/system/files/2025-04/ai-privacy-risks-and-mitigations-in-llms.pdf*, 2025. Cited on page 3.

David GT Barrett and Benoit Dherin. Implicit gradient regularization. *International Conference on Learning Representations (ICLR)*, 2021. Cited on page 34.

Peter Bartlett. For valid generalization the size of the weights is more important than the size of the network. *Advances in Neural Information Processing Systems (NeurIPS)*, 9, 1996. Cited on page 20.

Peter L. Bartlett, Dylan J. Foster, and Matus Telgarsky. Spectrally-normalized margin bounds for neural networks. In *Advances in Neural Information Processing Systems (NeurIPS)*, NIPS'17, 2017. Cited on page 20, 26.

Peter L. Bartlett, Philip M. Long, Gábor Lugosi, and Alexander Tsigler. Benign overfitting in linear regression. *Proceedings of the National Academy of Sciences*, 117 (48):30063–30070, 2020. Cited on page 27.

Peter L. Bartlett, Andrea Montanari, and Alexander Rakhlin. Deep learning: a statistical viewpoint. *Acta Numerica*, 30:87–201, 2021. Cited on page 26.

Peter L. Bartlett, Philip M. Long, and Olivier Bousquet. The dynamics of sharpness-aware minimization: Bouncing across ravines and drifting towards wide minima. *Journal of Machine Learning Research*, 24(316):1–36, 2023. Cited on page 29, 34.

Daniel Barzilai and Ohad Shamir. Generalization in kernel regression under realistic assumptions. In *Forty-first International Conference on Machine Learning (ICML)*, 2024. Cited on page 10, 27.

Peter W Battaglia, Jessica B Hamrick, Victor Bapst, Alvaro Sanchez-Gonzalez, Vinicius Zambaldi, Mateusz Malinowski, Andrea Tacchetti, David Raposo, Adam Santoro, Ryan Faulkner, et al. Relational inductive biases, deep learning, and graph networks. *arXiv:1806.01261*, 2018. Cited on page 26.

Mikhail Belkin, Daniel Hsu, Siyuan Ma, and Soumik Mandal. Reconciling modern machine-learning practice and the classical bias–variance trade-off. *Proceedings of the National Academy of Sciences*, 116(32):15849–15854, 2019a. Cited on page 25, 26.

Mikhail Belkin, Alexander Rakhlin, and Alexandre B. Tsybakov. Does data interpolation contradict statistical optimality? In *International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2019b. Cited on page 28.

Yoshua Bengio. Reasoning through arguments against taking ai safety seriously. *Personal website, see https://yoshuabengio.org/2024/07/09/reasoning-through-arguments-against-taking-ai-safety-seriously/*, 2024. Accessed: January 6th, 2025. Cited on page 3.

Jeremy Bernstein and Laker Newhouse. Modular duality in deep learning. *arXiv:2410.21265*, 2024. Cited on page 20.

Alberto Bietti and Francis Bach. Deep equals shallow for ReLU networks in kernel regimes. In *International Conference on Learning Representations (ICLR)*, 2021. Cited on page 12.

Charlie Blake, Constantin Eichenberg, Josef Dean, Lukas Balles, Luke Yuri Prince, Björn Deiseroth, Andres Felipe Cruz-Salinas, Carlo Luschi, Samuel Weinbach, and Douglas Orr. u-μp: The unit-scaled maximal update parametrization. In *The Thirteenth International Conference on Learning Representations (ICLR)*, 2025. Cited on page 21, 254.

Cristian Bodnar, Wessel P. Bruinsma, Ana Lucic, Megan Stanley, Johannes Brandstetter, Patrick Garvan, Maik Riechert, Jonathan Weyn, Haiyu Dong, Anna Vaughan, Jayesh K. Gupta, Kit Tambiratnam, Alex Archibald, Elizabeth Heider, Max Welling, Richard E. Turner, and Paris Perdikaris. Aurora: A foundation model of the atmosphere. *arXiv:2405.13063*, 2024. Cited on page 3.

Blake Bordelon and Cengiz Pehlevan. Self-consistent dynamical field theory of kernel evolution in wide neural networks. *Advances in Neural Information Processing Systems (NeurIPS)*, 35:32240–32256, 2022. Cited on page 17.

Blake Bordelon and Cengiz Pehlevan. Deep linear network training dynamics from random initialization: Data, width, depth, and hyperparameter transfer. *arXiv:2502.02531*, 2025. Cited on page 16, 30.

Blake Bordelon, Alexander Atanasov, and Cengiz Pehlevan. A dynamical model of neural scaling laws. *arXiv:2402.01092*, 2024a. Cited on page 5, 6, 30.

Blake Bordelon, Hamza Tahir Chaudhry, and Cengiz Pehlevan. Infinite limits of multi-head transformer dynamics. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems (NeurIPS)*, 2024b. Cited on page 17, 30, 31, 254.

Blake Bordelon, Lorenzo Noci, Mufan Bill Li, Boris Hanin, and Cengiz Pehlevan. Depthwise hyperparameter transfer in residual networks: Dynamics and scaling limit. In *The Twelfth International Conference on Learning Representations (ICLR)*, 2024c. Cited on page 17, 30, 254.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in Neural Information Processing Systems (NeurIPS)*, 33:1877–1901, 2020. Cited on page 3.

Simon Buchholz. Kernel interpolation in sobolev spaces is not consistent in low dimensions. In *Conference on Learning Theory (COLT)*, 2022. Cited on page 28.

Yuhang Cai, Jingfeng Wu, Song Mei, Michael Lindsey, and Peter Bartlett. Large stepsize gradient descent for non-homogeneous two-layer networks: Margin improvement and fast optimization. *Advances in Neural Information Processing Systems (NeurIPS)*, 37:71306–71351, 2024. Cited on page 17.

Nicholas Carr. What happens when ai-generated lies are more compelling than the truth? *Behavioral Scientist, see https://behavioralscientist.org/what-happens-when-ai-generated-lies-are-more-compelling-than-the-truth/*, 2024. Cited on page 3.

Xiangning Chen, Cho-Jui Hsieh, and Boqing Gong. When vision transformers outperform resnets without pre-training or strong data augmentations. *arXiv:2106.01548*, 2021. Cited on page 29.

Lenaic Chizat and Francis Bach. On the global convergence of gradient descent for over-parameterized models using optimal transport. *Advances in Neural Information Processing Systems (NeurIPS)*, 31, 2018. Cited on page 16.

Lénaïc Chizat and Francis Bach. Implicit bias of gradient descent for wide two-layer neural networks trained with the logistic loss. In *Conference on Learning Theory (COLT)*, 2020. Cited on page 17.

Lénaïc Chizat, Edouard Oyallon, and Francis Bach. On lazy training in differentiable programming. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2019. Cited on page 16, 21, 34.

Jeremy Cohen, Simran Kaur, Yuanzhi Li, J Zico Kolter, and Ameet Talwalkar. Gradient descent on neural networks typically occurs at the edge of stability. In *International Conference on Learning Representations (ICLR)*, 2020. Cited on page 17.

Jeremy Cohen, Simran Kaur, Yuanzhi Li, J Zico Kolter, and Ameet Talwalkar. Gradient descent on neural networks typically occurs at the edge of stability. In *International Conference on Learning Representations (ICLR)*, 2021. Cited on page 34.

Jeremy M Cohen, Behrooz Ghorbani, Shankar Krishnan, Naman Agarwal, Sourabh Medapati, Michal Badura, Daniel Suo, David Cardoze, Zachary Nado, George E Dahl, et al. Adaptive gradient methods at the edge of stability. *arXiv:2207.14484*, 2022. Cited on page 34.

Alicia Curth, Alan Jeffares, and Mihaela van der Schaar. A u-turn on double descent: Rethinking parameter counting in statistical learning. *Advances in Neural Information Processing Systems (NeurIPS)*, 36, 2024. Cited on page 26.

Alex Damian, Eshaan Nichani, and Jason D. Lee. Self-stabilization: The implicit bias of gradient descent at the edge of stability. In *International Conference on Learning Representations (ICLR)*, 2023. Cited on page 17.

Yatin Dandi, Luca Pesce, Lenka Zdeborová, and Florent Krzakala. The computational advantage of depth: Learning high-dimensional hierarchical functions with gradient descent. *arXiv:2502.13961*, 2025. Cited on page 255.

Amit Daniely, Roy Frostig, and Yoram Singer. Toward deeper understanding of neural networks: The power of initialization and a dual view on expressivity. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2016. Cited on page 33.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, 2009. Cited on page 4.

Hannah Devlin. Ai 'could be as transformative as industrial revolution'. *The Guardian, see https://www.theguardian.com/technology/2023/may/03/ai-could-be-as-transformative-as-industrial-revolution-patrick-vallance*, 2023. Cited on page 3.

Luc Devroye, László Györfi, and Gábor Lugosi. *A probabilistic theory of pattern recognition*. Springer Science & Business Media, 2013. Cited on page 27.

Nolan Dey, Bin Claire Zhang, Lorenzo Noci, Mufan Li, Blake Bordelon, Shane Bergsma, Cengiz Pehlevan, Boris Hanin, and Joel Hestness. Don't be lazy: Completep enables compute-efficient deep transformers. *arXiv:2505.01618*, 2025. Cited on page 31.

Laurent Dinh, Razvan Pascanu, Samy Bengio, and Yoshua Bengio. Sharp minima can generalize for deep nets. In Doina Precup and Yee Whye Teh, editors, *Proceedings of the 34th International Conference on Machine Learning (ICML)*, 2017. Cited on page 34.

Konstantin Donhauser, Mingqi Wu, and Fanny Yang. How rotational invariance of common kernels prevents generalization in high dimensions. In *International Conference on Machine Learning (ICML)*, 2021. Cited on page 28.

Ethan Dyer and Guy Gur-Ari. Asymptotics of wide networks from feynman diagrams. *arXiv:1909.11304*, 2019. Cited on page 5.

Gintare Karolina Dziugaite and Daniel M Roy. Computing nonvacuous generalization bounds for deep (stochastic) neural networks with many more parameters than training data. *arXiv:1703.11008*, 2017. Cited on page 26.

Katie E Everett, Lechao Xiao, Mitchell Wortsman, Alexander A Alemi, Roman Novak, Peter J Liu, Izzeddin Gur, Jascha Sohl-Dickstein, Leslie Pack Kaelbling, Jaehoon Lee, and Jeffrey Pennington. Scaling exponents across parameterizations and optimizers. In *Proceedings of the 41st International Conference on Machine Learning (ICML)*, 2024. Cited on page 34, 35.

Pierre Foret, Ariel Kleiner, Hossein Mobahi, and Behnam Neyshabur. Sharpness-aware minimization for efficiently improving generalization. In *International Conference on Learning Representations (ICLR)*, 2021. Cited on page 29, 31.

Stanislav Fort, Gintare Karolina Dziugaite, Mansheej Paul, Sepideh Kharaghani, Daniel M Roy, and Surya Ganguli. Deep learning versus kernel learning: an empirical study of loss landscape geometry and the time evolution of the neural tangent kernel. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020. Cited on page 33.

Jerome H Friedman. On bias, variance, 0/1—loss, and the curse-of-dimensionality. *Data mining and knowledge discovery*, 1:55–77, 1997. Cited on page 27.

Behrooz Ghorbani, Song Mei, Theodor Misiakiewicz, and Andrea Montanari. Linearized two-layers neural networks in high dimension. *The Annals of Statistics*, 49(2): 1029 – 1054, 2021. Cited on page 28.

Xavier Glorot and Yoshua Bengio. Understanding the difficulty of training deep feedforward neural networks. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2010. Cited on page 8, 11, 30.

Sebastian Goldt, Marc Mézard, Florent Krzakala, and Lenka Zdeborová. Modeling the influence of data structure on learning in neural networks: The hidden manifold model. *Physical Review X*, 10(4):041044, 2020. Cited on page 255.

Eugene A. Golikov. Dynamically stable infinite-width limits of neural classifiers. *arXiv:2006.06574*, 2020. Cited on page 33.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, et al. The llama 3 herd of models. *arXiv:2407.21783*, 2024. Cited on page 3.

Albert Gu and Tri Dao. Mamba: Linear-time sequence modeling with selective state spaces. *iv:2312.00752*, 2023. Cited on page 16, 20, 30.

Sharut Gupta, Chenyu Wang, Yifei Wang, Tommi Jaakkola, and Stefanie Jegelka. In-context symmetries: Self-supervised learning through contextual world models. *Advances in Neural Information Processing Systems (NeurIPS)*, 37:104250–104280, 2024. Cited on page 255.

Vineet Gupta, Tomer Koren, and Yoram Singer. Shampoo: Preconditioned stochastic tensor optimization. In *International Conference on Machine Learning (ICML)*, 2018. Cited on page 4.

Boris Hanin and David Rolnick. How to start training: The effect of initialization and architecture. *Advances in Neural Information Processing Systems (NeurIPS)*, 31, 2018. Cited on page 11.

Boris Hanin and Alexander Zlokapa. Bayesian inference with deep weakly nonlinear networks. *arXiv:2405.16630*, 2024. Cited on page 30.

Moritz Hardt. The emerging science of machine learning benchmarks. Online at `https://mlbenchmarks.org`, 2025. Manuscript. Cited on page 4.

Trevor Hastie, Robert Tibshirani, and Jerome Friedman. *The elements of statistical learning: data mining, inference, and prediction. Volume 2*. Springer, 2009. Cited on page 25, 26.

Trevor Hastie, Andrea Montanari, Saharon Rosset, and Ryan J. Tibshirani. Surprises in high-dimensional ridgeless least squares interpolation. *The Annals of Statistics*, 50(2): 949 – 986, 2022. Cited on page 25, 26.

Soufiane Hayou and Greg Yang. Width and depth limits commute in residual networks. In *International Conference on Machine Learning (ICML)*, 2023. Cited on page 31.

Soufiane Hayou, Eugenio Clerico, Bobby He, George Deligiannidis, Arnaud Doucet, and Judith Rousseau. Stable resnet. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2021. Cited on page 30.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *IEEE international conference on computer vision (ICCV)*, 2015. Cited on page 4, 8, 11, 30, 32.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016. Cited on page 4.

Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997a. Cited on page 4.

Sepp Hochreiter and Jürgen Schmidhuber. Flat minima. *Neural Comput.*, 9(1):1–42, 1997b. Cited on page 34.

Jordan Hoffmann, Sebastian Borgeaud, Arthur Mensch, Elena Buchatskaya, Trevor Cai, Eliza Rutherford, Diego de Las Casas, Lisa Anne Hendricks, Johannes Welbl, Aidan Clark, et al. An empirical analysis of compute-optimal large language model training. *Advances in Neural Information Processing Systems (NeurIPS)*, 35:30016–30030, 2022. Cited on page 5, 254.

Noah Hollmann, Samuel Müller, Lennart Purucker, Arjun Krishnakumar, Max Körfer, Shi Bin Hoo, Robin Tibor Schirrmeister, and Frank Hutter. Accurate predictions on small data with a tabular foundation model. *Nature*, 2025. Cited on page 3.

Benedikt Höltgen and Robert C Williamson. Which distribution were you sampled from? towards a more tangible conception of data. *arXiv:2407.17395*, 2024. Cited on page 255.

Satoki Ishikawa and Ryo Karakida. On the parameterization of second-order optimization effective towards the infinite width. *International Conference on Learning Representations (ICLR)*, 2024. Cited on page 31.

Arthur Jacot, Franck Gabriel, and Clément Hongler. Neural Tangent Kernel: Convergence and generalization in neural networks. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2018. Cited on page 5, 14, 15, 33.

Alan Jeffares, Alicia Curth, and Mihaela van der Schaar. Deep learning through a telescoping lens: A simple model provides empirical insights on grokking, gradient boosting & beyond. *Advances in Neural Information Processing Systems (NeurIPS)*, 37: 123498–123533, 2024. Cited on page 15.

William Stanley Jevons. *The coal question; an inquiry concerning the progress of the nation and the probable exhaustion of our coal-mines*. Macmillan, 1866. Cited on page 5.

Keller Jordan, Yuchen Jin, Vlado Boza, You Jiacheng, Franz Cesista, Laker Newhouse, and Jeremy Bernstein. Muon: An optimizer for hidden layers in neural networks, 2024. Cited on page 4, 20.

John Jumper, Richard Evans, Alexander Pritzel, Tim Green, Michael Figurnov, Olaf Ronneberger, Kathryn Tunyasuvunakool, Russ Bates, Augustin Žídek, Anna Potapenko, et al. Highly accurate protein structure prediction with alphafold. *Nature*, 596(7873):583–589, 2021. Cited on page 3.

Jean Kaddour, Linqing Liu, Ricardo Silva, and Matt J Kusner. When do flat minima optimizers work? *Advances in Neural Information Processing Systems (NeurIPS)*, 35: 16577–16595, 2022. Cited on page 29.

Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. Scaling laws for neural language models. *arXiv:2001.08361*, 2020. Cited on page 5, 27, 29, 254.

Kenji Kawaguchi, Leslie Pack Kaelbling, and Yoshua Bengio. Generalization in deep learning. In *Mathematical Aspects of Deep Learning*. Cambridge University Press, 2022. Cited on page 26.

Nitish Shirish Keskar, Dheevatsa Mudigere, Jorge Nocedal, Mikhail Smelyanskiy, and Ping Tak Peter Tang. On large-batch training for deep learning: Generalization gap and sharp minima. *International Conference on Learning Representations (ICLR)*, 2017. Cited on page 34.

Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv:1412.6980*, 2014. Cited on page 4.

Zoe Kleinman and Chris Vallance. Ai 'godfather' geoffrey hinton warns of dangers as he quits google. *BBC, see https://www.bbc.com/news/world-us-canada-65452940*, 2023. Cited on page 3.

Dmitry Kobak, Jonathan Lomond, and Benoit Sanchez. The optimal ridge penalty for real-world high-dimensional data can be zero or negative due to the implicit ridge regularization. *Journal of Machine Learning Research*, 21(169):1–16, 2020. Cited on page 26.

Dmitrii Kochkov, Janni Yuval, Ian Langmore, Peter Norgaard, Jamie Smith, Griffin Mooers, Milan Klöwer, James Lottes, Stephan Rasp, Peter Düben, et al. Neural general circulation models for weather and climate. *Nature*, 632(8027):1060–1066, 2024. Cited on page 3.

Vladimir Koltchinskii and Olexandra Beznosova. Exponential convergence rates in classification. In *International Conference on Computational Learning Theory*, 2005. Cited on page 27.

Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2012. Cited on page 4.

Daniel Kunin, Allan Raventos, Clémentine Carla Juliette Dominé, Feng Chen, David Klindt, Andrew M Saxe, and Surya Ganguli. Get rich quick: exact solutions reveal how unbalanced initializations promote rapid feature learning. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems (NeurIPS)*, 2024. Cited on page 5, 21, 34, 253.

Jungmin Kwon, Jeongseop Kim, Hyunseo Park, and In Kwon Choi. Asam: Adaptive sharpness-aware minimization for scale-invariant learning of deep neural networks. In *International Conference on Machine Learning (ICML)*, 2021. Cited on page 29, 31.

Remi Lam, Alvaro Sanchez-Gonzalez, Matthew Willson, Peter Wirnsberger, Meire Fortunato, Ferran Alet, Suman Ravuri, Timo Ewalds, Zach Eaton-Rosen, Weihua Hu, et al. Learning skillful medium-range global weather forecasting. *Science*, 382 (6677):1416–1421, 2023. Cited on page 3.

Yann LeCun and Yoshua Bengio. Convolutional networks for images, speech, and time series. *The handbook of brain theory and neural networks*, 3361(10), 1995. Cited on page 4.

Yann LeCun, Bernhard Boser, John S Denker, Donnie Henderson, Richard E Howard, Wayne Hubbard, and Lawrence D Jackel. Backpropagation applied to handwritten zip code recognition. *Neural computation*, 1(4):541–551, 1989. Cited on page 4.

Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998. Cited on page 4.

Jaehoon Lee, Yasaman Bahri, Roman Novak, Samuel S. Schoenholz, Jeffrey Pennington, and Jascha Sohl-Dickstein. Deep neural networks as Gaussian processes. In *International Conference on Learning Representations (ICLR)*, 2018. Cited on page 11, 33.

Jaehoon Lee, Lechao Xiao, Samuel Schoenholz, Yasaman Bahri, Roman Novak, Jascha Sohl-Dickstein, and Jeffrey Pennington. Wide neural networks of any depth evolve as linear models under gradient descent. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2019. Cited on page 5, 12, 15, 33.

Jaehoon Lee, Samuel Schoenholz, Jeffrey Pennington, Ben Adlam, Lechao Xiao, Roman Novak, and Jascha Sohl-Dickstein. Finite versus infinite neural networks: an empirical study. *Advances in Neural Information Processing Systems (NeurIPS)*, 33: 15156–15172, 2020. Cited on page 33.

Aitor Lewkowycz, Yasaman Bahri, Ethan Dyer, Jascha Sohl-Dickstein, and Guy Gur-Ari. The large learning rate phase of deep learning: the catapult mechanism. *arXiv:2003.02218*, 2020. Cited on page 34, 35.

Mufan Li, Mihai Nica, and Dan Roy. The future is log-gaussian: Resnets and their infinite-depth-and-width limit at initialization. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2021. Cited on page 30.

Mufan Li, Mihai Nica, and Dan Roy. The neural covariance sde: Shaped infinite depth-and-width networks at initialization. *Advances in Neural Information Processing Systems (NeurIPS)*, 35:10795–10808, 2022. Cited on page 30.

Yuanzhi Li, Colin Wei, and Tengyu Ma. Towards explaining the regularization effect of initial large learning rate in training neural networks. *Advances in neural information processing systems (NeurIPS)*, 32, 2019. Cited on page 34.

Tengyuan Liang and Alexander Rakhlin. Just interpolate: Kernel "ridgeless" regression can generalize. *Annals of Statistics*, 48(3):1329–1347, 2020. Cited on page 26.

Tengyuan Liang, Tomaso Poggio, Alexander Rakhlin, and James Stokes. Fisher-rao metric, geometry, and complexity of neural networks. In *The 22nd international conference on artificial intelligence and statistics (AISTATS)*, 2019. Cited on page 26.

Tengyuan Liang, Alexander Rakhlin, and Xiyu Zhai. On the multiple descent of minimum-norm interpolants and restricted lower isometry of kernels. In *Conference on Learning Theory (COLT)*, 2020. Cited on page 28.

Marco Loog, Tom Viering, Alexander Mey, Jesse H Krijthe, and David MJ Tax. A brief prehistory of double descent. *Proceedings of the National Academy of Sciences*, 117(20): 10625–10626, 2020. Cited on page 25.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations (ICLR)*, 2019. Cited on page 4.

Neil Rohit Mallinar, James B Simon, Amirhesam Abedsoltan, Parthe Pandit, Misha Belkin, and Preetum Nakkiran. Benign, tempered, or catastrophic: Toward a refined taxonomy of overfitting. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2022. Cited on page 27.

Bernard Marr. Ai: Overhyped fantasy or truly the next industrial revolution? *Forbes, see https://www.forbes.com/sites/bernardmarr/2024/08/15/ai-overhyped-fantasy-or-truly-the-next-industrial-revolution/*, 2024. Cited on page 3.

Alexander G. de G. Matthews, Jiri Hron, Mark Rowland, Richard E. Turner, and Zoubin Ghahramani. Gaussian process behaviour in wide deep neural networks. In *International Conference on Learning Representations (ICLR)*, 2018. Cited on page 11, 33.

Andrew D. Mcrae, Santhosh Karnik, Mark Davenport, and Vidya K. Muthukumar. Harmless interpolation in regression and classification with structured features. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2022. Cited on page 26.

Song Mei, Andrea Montanari, and Phan-Minh Nguyen. A mean field view of the landscape of two-layer neural networks. *Proceedings of the National Academy of Sciences*, 115(33):E7665–E7671, 2018. Cited on page 14, 16.

Cade Metz. The godfather of ai'leaves google and warns of danger ahead. *The New York Times*, 1, 2023. Cited on page 3.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. *Advances in Neural Information Processing Systems (NeurIPS)*, 26, 2013. Cited on page 4.

Tom M Mitchell. The need for biases in learning generalizations. *Rutgers CS tech report CBM-TR-117*, 1980. Cited on page 26.

Andrea Montanari and Yiqiao Zhong. The interpolation phase transition in neural networks: Memorization and generalization under lazy training. *The Annals of Statistics*, 50(5):2816–2847, 2022. Cited on page 28.

Enea Monzio Compagnoni, Luca Biggio, Antonio Orvieto, Frank Norbert Proske, Hans Kersting, and Aurelien Lucchi. An SDE for modeling SAM: Theory and insights. In Andreas Krause, Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato, and Jonathan Scarlett, editors, *Proceedings of the 40th International Conference on Machine Learning (ICML)*, 2023. Cited on page 29, 31.

Vidya Muthukumar, Kailas Vodrahalli, Vignesh Subramanian, and Anant Sahai. Harmless interpolation of noisy data in regression. *IEEE Journal on Selected Areas in Information Theory*, 2020. Cited on page 26.

Vidya Muthukumar, Adhyyan Narang, Vignesh Subramanian, Mikhail Belkin, Daniel Hsu, and Anant Sahai. Classification vs regression in overparameterized regimes: Does the loss function matter? *JMLR*, 22(1):10104–10172, 2021. Cited on page 27.

Maximilian Müller, Tiffany Vlaar, David Rolnick, and Matthias Hein. Normalization layers are all that sharpness-aware minimization needs. *Advances in Neural Information Processing Systems (NeurIPS)*, 36, 2024. Cited on page 29, 31, 34.

Vinod Nair and Geoffrey E Hinton. Rectified linear units improve restricted boltzmann machines. In *Proceedings of the 27th international conference on machine learning (ICML-10)*, 2010. Cited on page 4.

Radford M. Neal. *Priors for Infinite Networks*. Springer New York, 1996. Cited on page 5, 11, 14, 33.

Yurii Nesterov et al. *Lectures on convex optimization*. Springer, 2018. Cited on page 34.

Behnam Neyshabur, Ryota Tomioka, and Nathan Srebro. Norm-based capacity control in neural networks. In *Proceedings of The 28th Conference on Learning Theory*, 2015. Cited on page 20.

Behnam Neyshabur, Srinadh Bhojanapalli, David Mcallester, and Nati Srebro. Exploring generalization in deep learning. In I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems (NeurIPS)*, 2017. Cited on page 26.

Phan-Minh Nguyen and Huy Tuan Pham. A rigorous framework for the mean field limit of multilayer neural networks. *Mathematical Statistics and Learning*, 6(3):201–357, 2023. Cited on page 16.

Zhaoyang Niu, Guoqiang Zhong, and Hui Yu. A review on the attention mechanism of deep learning. *Neurocomputing*, 452:48–62, 2021. Cited on page 4.

Lorenzo Noci, Sotiris Anagnostidis, Luca Biggio, Antonio Orvieto, Sidak Pal Singh, and Aurelien Lucchi. Signal propagation in transformers: Theoretical perspectives and the role of rank collapse. *Advances in Neural Information Processing Systems (NeurIPS)*, 35:27198–27211, 2022. Cited on page 31.

Lorenzo Noci, Chuning Li, Mufan Li, Bobby He, Thomas Hofmann, Chris J Maddison, and Dan Roy. The shaped transformer: Attention models in the infinite depth-and-width limit. *Advances in Neural Information Processing Systems (NeurIPS)*, 36, 2024a. Cited on page 31.

Lorenzo Noci, Alexandru Meterez, Thomas Hofmann, and Antonio Orvieto. Super consistency of neural network landscapes and learning rate transfer. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems (NeurIPS)*, 2024b. Cited on page 6, 16, 30, 254.

Roman Novak, Lechao Xiao, Yasaman Bahri, Jaehoon Lee, Greg Yang, Daniel A. Abolafia, Jeffrey Pennington, and Jascha Sohl-dickstein. Bayesian deep convolutional networks with many channels are gaussian processes. In *International Conference on Learning Representations (ICLR)*, 2019. Cited on page 33.

OLMo Team, Pete Walsh, Luca Soldaini, Dirk Groeneveld, Kyle Lo, Shane Arora, Akshita Bhagia, Yuling Gu, Shengyi Huang, Matt Jordan, et al. 2 olmo 2 furious. *arXiv:2501.00656*, 2024. Cited on page 3.

Manfred Opper, W Kinzel, J Kleinz, and R Nehl. On the ability of the optimal perceptron to generalise. *Journal of Physics A: Mathematical and General*, 23(11), 1990. Cited on page 25.

Guillermo Ortiz-Jiménez, Seyed-Mohsen Moosavi-Dezfooli, and Pascal Frossard. What can linearized neural networks actually say about generalization? *Advances in Neural Information Processing Systems (NeurIPS)*, 34:8998–9010, 2021. Cited on page 33.

Tomer Porian, Mitchell Wortsman, Jenia Jitsev, Ludwig Schmidt, and Yair Carmon. Resolving discrepancies in compute-optimal scaling of language models. *Advances in Neural Information Processing Systems (NeurIPS)*, 37:100535–100570, 2024. Cited on page 5.

Dan Qiao, Kaiqi Zhang, Esha Singh, Daniel Soudry, and Yu-Xiang Wang. Stable minima cannot overfit in univariate relu networks: Generalization by large step sizes. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems (NeurIPS)*, 2024. Cited on page 34.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International Conference on Machine Learning (ICML)*, 2021. Cited on page 3.

Ali Rahimi and Benjamin Recht. Random features for large-scale kernel machines. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2008. Cited on page 12.

Alexander Rakhlin and Xiyu Zhai. Consistency of interpolation with Laplace kernels is a high-dimensional phenomenon. In *Conference on Learning Theory (COLT)*, 2019. Cited on page 28.

Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. Hierarchical text-conditional image generation with clip latents. *arXiv:2204.06125*, 1 (2):3, 2022. Cited on page 3.

Yunwei Ren, Eshaan Nichani, Denny Wu, and Jason D Lee. Emergence and scaling laws in sgd learning of shallow neural networks. *arXiv:2504.19983*, 2025. Cited on page 5, 253.

Frank Rosenblatt. The perceptron: a probabilistic model for information storage and organization in the brain. *Psychological review*, 65(6):386, 1958. Cited on page 4.

David E Rumelhart, Geoffrey E Hinton, James L McClelland, et al. A general framework for parallel distributed processing. *Parallel distributed processing: Explorations in the microstructure of cognition*, 1(45-76):26, 1986a. Cited on page 4.

David E Rumelhart, Geoffrey E Hinton, and Ronald J Williams. Learning representations by back-propagating errors. *Nature*, 323(6088):533–536, 1986b. Cited on page 4.

Jürgen Schmidhuber. Learning to control fast-weight memories: An alternative to dynamic recurrent networks. *Neural Computation*, 4(1):131–139, 1992. Cited on page 4.

Bernhard Schölkopf and Alexander J. Smola. *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond*. MIT Press, 2001. Cited on page 10.

Ohad Shamir. The implicit bias of benign overfitting. In *Conference on Learning Theory (COLT)*, 2022. Cited on page 27.

James B Simon, Dhruva Karkada, Nikhil Ghosh, and Mikhail Belkin. More is better in modern machine learning: when infinite overparameterization is optimal and overfitting is obligatory. *International Conference on Learning Representations (ICLR)*, 2024. Cited on page 26, 255.

Justin Sirignano and Konstantinos Spiliopoulos. Mean field analysis of deep neural networks. *Mathematics of Operations Research*, 47(1):120–152, 2022. Cited on page 16.

Jascha Sohl-Dickstein, Roman Novak, Samuel S Schoenholz, and Jaehoon Lee. On the infinite width limit of neural networks with a standard parameterization. *arXiv:2001.07301*, 2020. Cited on page 32, 34.

Daniel Soudry, Elad Hoffer, Mor Shpigel Nacson, Suriya Gunasekar, and Nathan Srebro. The implicit bias of gradient descent on separable data. *Journal of Machine Learning Research*, 19(70):1–57, 2018. Cited on page 26.

Ingo Steinwart and Andreas Christmann. *Support Vector Machines*. Springer New York, 2008. Cited on page 10.

Trieu H Trinh, Yuhuai Wu, Quoc V Le, He He, and Thang Luong. Solving olympiad geometry without human demonstrations. *Nature*, 625(7995):476–482, 2024. Cited on page 3.

Alexander Tsigler and Peter L Bartlett. Benign overfitting in ridge regression. *Journal of Machine Learning Research*, 24(123):1–76, 2023. Cited on page 26, 27.

Alexander Tsigler, Luiz FO Chamon, Spencer Frei, and Peter L Bartlett. Benign overfitting and the geometry of the ridge regression solution in binary classification. *arXiv:2503.07966*, 2025. Cited on page 5, 253.

Yusuke Tsuzuku, Issei Sato, and Masashi Sugiyama. Normalized flat minima: Exploring scale invariant definition of flat minima for neural networks using pac-bayesian analysis. In *International Conference on Machine Learning (ICML)*, 2020. Cited on page 34.

F Vallet, J-G Cailton, and Ph Refregier. Linear and nonlinear extension of the pseudo-inverse solution for learning boolean functions. *Europhysics Letters*, 9(4):315, 1989. Cited on page 25, 26.

Leena Chennuru Vankadara, Jin Xu, Moritz Haas, and Volkan Cevher. On feature learning in structured state space models. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems (NeurIPS)*, 2024. Cited on page 16, 20, 30, 32.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in Neural Information Processing Systems (NeurIPS)*, 2017. Cited on page 3, 4.

Nikhil Vyas, Yamini Bansal, and Preetum Nakkiran. Limitations of the ntk for understanding generalization in deep learning. *arXiv:2206.10012*, 2022. Cited on page 33.

Nikhil Vyas, Alexander Atanasov, Blake Bordelon, Depen Morwani, Sabarish Sainathan, and Cengiz Pehlevan. Feature-learning networks are consistent across widths at realistic scales. *Advances in Neural Information Processing Systems (NeurIPS)*, 36, 2024. Cited on page 5, 6, 16, 30, 254.

Angelina Wang, Sayash Kapoor, Solon Barocas, and Arvind Narayanan. Against predictive optimization: On the legitimacy of decision-making algorithms that optimize predictive accuracy. *ACM Journal on Responsible Computing*, 1(1):1–45, 2024. Cited on page 3.

Kaiyue Wen, Tengyu Ma, and Zhiyuan Li. How sharpness-aware minimization minimizes sharpness? In *The Eleventh International Conference on Learning Representations (ICLR)*, 2023. Cited on page 29.

Kaiyue Wen, Zhiyuan Li, and Tengyu Ma. Sharpness minimization algorithms do not only minimize sharpness to achieve better generalization. *Advances in Neural Information Processing Systems (NeurIPS)*, 36, 2024. Cited on page 29.

Jonathan Wenger, Felix Dangel, and Agustinus Kristiadi. On the disconnect between theory and practice of overparametrized neural networks. *arXiv:2310.00137*, 2023. Cited on page 13, 30.

Blake Woodworth, Suriya Gunasekar, Jason D. Lee, Edward Moroshko, Pedro Savarese, Itay Golan, Daniel Soudry, and Nathan Srebro. Kernel and rich regimes in overparametrized models. In *Conference on Learning Theory (COLT)*, 2020. Cited on page 34.

Abraham J Wyner, Matthew Olson, Justin Bleich, and David Mease. Explaining the success of adaboost and random forests as interpolating classifiers. *Journal of Machine Learning Research*, 18(48):1–33, 2017. Cited on page 25, 28.

Greg Yang. Wide feedforward or recurrent neural networks of any architecture are gaussian processes. *Advances in Neural Information Processing Systems (NeurIPS)*, 32, 2019. Cited on page 11, 17.

Greg Yang. Tensor programs iii: Neural matrix laws. *arXiv:2009.10685*, 2021. Cited on page 11, 17.

Greg Yang and Edward J. Hu. Tensor programs iv: Feature learning in infinite-width neural networks. In *International Conference on Machine Learning (ICML)*, 2021. Cited on page 5, 6, 11, 13, 14, 16, 17, 32, 33, 34.

Greg Yang and Etai Littwin. Tensor programs ivb: Adaptive optimization in the infinite-width limit. *arXiv:2308.01814*, 2023. Cited on page 6, 11, 17, 34.

Greg Yang, Edward J Hu, Igor Babuschkin, Szymon Sidor, Xiaodong Liu, David Farhi, Nick Ryder, Jakub Pachocki, Weizhu Chen, and Jianfeng Gao. Tensor programs v: Tuning large neural networks via zero-shot hyperparameter transfer. *arXiv:2203.03466*, 2022. Cited on page 6, 16, 21, 29, 30, 254.

Greg Yang, James B. Simon, and Jeremy Bernstein. A spectral condition for feature learning. *arXiv:2310.17813*, 2023a. Cited on page 11, 19.

Greg Yang, Dingli Yu, Chen Zhu, and Soufiane Hayou. Tensor programs vi: Feature learning in infinite-depth neural networks. *arXiv:2310.02244*, 2023b. Cited on page 11, 17, 30.

Yuan Yao, Lorenzo Rosasco, and Andrea Caponnetto. On early stopping in gradient descent learning. *Constructive Approximation*, 26:289–315, 2007. Cited on page 16.

Lenka Zdeborová. Understanding deep learning is also a job for physicists. *Nature Physics*, 16(6):602–604, 2020. Cited on page 5, 26.

Chiyuan Zhang, Samy Bengio, Moritz Hardt, Benjamin Recht, and Oriol Vinyals. Understanding deep learning (still) requires rethinking generalization. *Communications of the ACM*, 64(3):107–115, 2021. Cited on page 25.

Ruiqi Zhang, Jingfeng Wu, Licong Lin, and Peter L Bartlett. Minimax optimal convergence of gradient descent in logistic regression via large and adaptive stepsizes. *arXiv:2504.04105*, 2025. Cited on page 5, 253.

Wenda Zhou, Victor Veitch, Morgane Austern, Ryan P. Adams, and Peter Orbanz. Non-vacuous generalization bounds at the imagenet scale: a PAC-bayesian compression approach. In *International Conference on Learning Representations (ICLR)*, 2019. Cited on page 26.