Inductive Bias in Machine Learning

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. Luca Silvester Rendsburg
aus Tübingen

Tübingen
2022

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

Tag der mündlichen Qualifikation	24.01.2023
Dekan:	Prof. Dr. Thilo Stehle
1. Berichterstatterin:	Prof. Dr. Ulrike von Luxburg
2. Berichterstatter:	Prof. Dr. Robert C. Williamson

Abstract

Inductive bias describes the preference for solutions that a machine learning algorithm holds before seeing any data. It is a necessary ingredient for the goal of machine learning, which is to generalize from a set of examples to unseen data points. Yet, the inductive bias of learning algorithms is often not specified explicitly in practice, which prevents a theoretical understanding and undermines trust in machine learning. This issue is most prominently visible in the contemporary case of deep learning, which is widely successful in applications but relies on many poorly understood techniques and heuristics. This thesis aims to uncover the hidden inductive biases of machine learning algorithms.

In the first part of the thesis, we uncover the implicit inductive bias of NetGAN, a complex graph generative model with seemingly no prior preferences. We find that the root of its generalization properties does not lie in the GAN architecture but in an inconspicuous low-rank approximation. We then use this insight to strip NetGAN of all unnecessary parts, including the GAN, and obtain a highly simplified reformulation.

Next, we present a generic algorithm that reverse-engineers hidden inductive bias in approximate Bayesian inference. While the inductive bias is completely described by the prior distribution in full Bayesian inference, real-world applications often resort to approximate techniques that can make uncontrollable errors. By reframing the problem in terms of incompatible conditional distributions, we arrive at a generic algorithm based on pseudo-Gibbs sampling that attributes the change in inductive bias to a change in the prior distribution.

The last part of the thesis concerns a common inductive bias in causal learning, the assumption of independent causal mechanisms. Under this assumption, we consider estimators for confounding strength, which governs the generalization ability from observational distribution to the underlying causal model. We show that an existing estimator is generally inconsistent and propose a consistent estimator based on tools from random matrix theory.

Zusammenfassung

Induktive Verzerrung beschreibt die Präferenz für Lösungen, welche ein Algorithmus für maschinelles Lernen hat, bevor er Daten sieht. Sie ist notwendiger Bestandteil für das Ziel des maschinellen Lernens, nämlich von einer Menge an Beispielen auf ungesehene Datenpunkte zu verallgemeinern. In der Praxis wird die induktive Verzerrung jedoch oft nicht explizit spezifiziert, was theoretisches Verständnis verhindert und das Vertrauen in maschinelles Lernen untergräbt. Am deutlichsten wird dieses Problem am zeitgenössischen Beispiel von deep learning, das zwar in vielen Anwendungen erfolgreich ist, aber auf einer Vielzahl schlecht verstandener Techniken und Heuristiken beruht. Ziel dieser Dissertation ist es, die versteckten induktiven Verzerrungen von Algorithmen des maschinellen Lernens aufzudecken.

Im ersten Teil der Dissertation decken wir die induktive Verzerrung von NetGAN auf, einem komplexen generativen Graphenmodell, das scheinbar keine Präferenzen hat. Wir stellen fest, dass die Ursache der Generalisierung nicht in der GAN-Architektur liegt, sondern in einer unscheinbaren Approximation mit niedrigem Rang. Wir nutzen diese Erkenntnis, um NetGAN von allen unnötigen Teilen, einschließlich des GAN, zu befreien und eine stark vereinfachte Reformulierung zu erhalten.

Als Nächstes präsentieren wir einen generischen Algorithmus, der die versteckte induktive Verzerrung in der approximativen Bayesschen Inferenz enthüllt. Während die induktive Verzerrung bei der Bayesschen Inferenz vollständig durch den Prior beschrieben wird, greifen reale Anwendungen oft auf approximative Techniken zurück, die unkontrollierbare Fehler machen können. Indem wir das Problem in Form von inkompatiblen bedingten Verteilungen reformulieren, kommen wir zu einem generischen Algorithmus, der auf Pseudo-Gibbs-Sampling basiert und die Änderung der induktiven Verzerrung auf eine Änderung des Priors zurückführt.

Der letzte Teil der Dissertation betrifft eine häufige induktive Verzerrung beim kausalen Lernen, die Annahme unabhängiger kausaler Mechanismen. Unter dieser Annahme betrachten wir Schätzer für die Stärke von Störfaktoren, die die Generalisierung von der Beobachtungsverteilung auf das zugrunde liegende kausale Modell bestimmt. Wir zeigen, dass ein bestehender Schätzer im Allgemeinen inkonsistent ist und präsentieren einen konsistenten Schätzer mit Werkzeugen aus der Theorie von Zufallsmatrizen.

Acknowledgements

I want to thank my supervisor Prof. Dr. Ulrike von Luxburg for her help and advice with this PhD. I also want to thank Damien Garreau for his supervision during my internship, which motivated me to start a PhD. Thanks to my fellow group members Sebastian Bordt, Leena Chennuru Vankadara, Moritz Haas, Solveig Klepper, and Michael Lohaus for engaging in scientific discussions and keeping me company during coffee breaks. I appreciate the support from Leila Masri and my thesis advisory committee Prof. Dr. Ulrike von Luxburg, Prof. Dr. Philipp Hennig, and Prof. Dr. Philipp Berens as part of the international Max Planck Research School for Intelligent Systems. Last but not least, I am deeply grateful to Hannah for her emotional support.

Contents

Chapter 1

Introduction

In the classical research cycle, the researcher starts by formulating a hypothesis based on their expertise. They then design an experiment that tests how well the hypothesis predicts real data. If there is a mismatch, the researcher carefully analyzes the data, updates the hypothesis, and repeats the cycle. This process repeatedly applies two opposing principles, deduction and induction. Deduction starts with a general hypothesis and uses logic to deduce exact conclusions for particular data points. For example, Newton's second law of motion Force = Mass $\times$ Acceleration can be rearranged to predict how much a particular object accelerates if we know its mass and the applied force. Reversely, induction starts with particular data points and aims to estimate a general hypothesis. For example, we can apply various forces to objects with various masses, observe their accelerations, and then use these data to derive a general rule that governs their relationship.

In today's information age, data are available in great abundance. While this facilitates the task of inductive inference, it also makes its manual implementation challenging, if not outright impossible. This is the case if there are too many data points to be parsed manually or if the underlying rule is too complex to be captured by a simple hypothesis. An example of the latter is digit recognition: people generally agree when asked to classify a specific hand-written digit, that is, they share the same underlying classification rule but find it hard to specify this rule explicitly.

Machine learning provides a remedy to these issues by automating the process of inductive inference. Generically speaking, a machine learning algorithm $\mathcal{A}$ is fed with a data set D and, without further intervention, returns a hypothesis H that tries to solve a given learning task. This inductive inference step is called the training phase. Once it is completed, the hypothesis can be used to make deductions. There are many different instances of this abstract framework. To name just a few, the data set types can include text in natural language processing, audio in speech recognition, images or video in computer vision, categorical data such as gender or ethnicity, or continuous data such as height or age. Learning tasks are commonly separated into the categories

of supervised and unsupervised learning. In supervised learning, each data point is associated with a label and the task is to learn a function that predicts those labels for given data points. Depending on the type of labels, the task is called classification for categorical labels and regression for continuous labels. In unsupervised learning, data points have no labels and the task is to discover patterns in the data. This includes clustering, in which the data set is partitioned into meaningful subgroups, ranking, in which the data points are ordered according to some criterion, or dimensionality reduction, in which the data are transformed into a simpler lower-dimensional representation without destroying their properties.

In the remainder of this paragraph, I introduce a specific formal learning problem based on which I will discuss the general concept of inductive bias. Assume we are given a set of labeled data points $D = \{(x_1, y_1), \dots, (x_n, y_n)\} \subset \mathbb{R}^d \times \mathbb{R}$ which are sampled independently from a distribution $\mathbb{P}_{X,Y}$. The task is to predict the (random) labels y of given data points x, a relationship which is fully described by the family of conditional distributions $\mathbb{P}_{Y|X}$. However, in practice, we often settle for the easier task of learning the most reasonable deterministic predictor $f\colon \mathbb{R}^d \to \mathbb{R}$ from a set of functions $f \in \mathcal{F}$. In this context, "most reasonable" is specified by a loss function $l\colon \mathbb{R} \times \mathbb{R} \to \mathbb{R}_{\geq 0}$. The loss $l(y', y)$ penalizes a prediction y' that differs from the true label y with the convention $l(y, y) = 0$ for exact predictions. Since the goal is to give good predictions on unseen data points, the *risk* of a predictor $f \in \mathcal{F}$ is defined as the average loss

$$\mathcal{R}(f) = \mathbb{E}_{(x,y)\sim\mathbb{P}_{X,Y}}[l(f(x), y)]$$

over unseen test points $(x, y) \sim \mathbb{P}_{X,Y}$ sampled from the same distribution as the training set. A predictor with minimal risk is called a *Bayes classifier* f^* with corresponding *Bayes risk* $\mathcal{R}^*$. Formally, they are given by

$$f^* \in \arg\min_{f\in\mathcal{F}} \mathcal{R}(f) \quad \text{and} \quad \mathcal{R}^* = \mathcal{R}(f^*)\,.$$

A learning algorithm successfully *generalizes* to unseen data if it returns a predictor whose risk is lower than that of random guessing.

Thesis structure This thesis is structured as follows. I first discuss inductive bias on a general level through the concepts of the no free lunch theorem (Section 1.1) and Occam's razor (Section 1.2). Next, I move to the more concrete example of inductive bias in deep learning for which I discuss the ways in which this bias is encoded explicitly (Section 1.3), why the success of deep learning relies on hidden, implicitly specified inductive bias (Section 1.4), and recent efforts to uncover this hidden bias (Section 1.5). Section 1.6 summarizes my original contributions, which also revolve around uncovering hidden inductive bias. They are presented in Chapters 2, 3, and 4. Chapter 5 concludes by discussing the necessity and the advantages of explicit inductive bias.

1.1 The no free lunch theorem: inductive bias is necessary

The inductive inference task requires generalizing information from a particular data set to unseen test points. This inverse problem is ill-posed since there are multiple (usually infinitely many) predictors which can explain the observed data. There is no unique candidate predictor and we have to make an arbitrary choice. For example, assume we observe the binary sequence 0, 1, 0, 0 and are asked to predict the next unseen element in $\{0, 1\}$. Is there any clever argument for why we should prefer one option over the other? The no free lunch theorem (Duda et al., 2000; Wolpert, 1994) tells us that this is not the case:

Theorem 1 (No free lunch, informal statement). *Uniformly averaged over all possible problem instances, all learning algorithms have the same generalization power.*

In other words, no algorithm is inherently superior to any other and there is no fundamental reason to prefer any elaborate, complex algorithm to random guessing. Algorithms can only differ in the way they distribute their generalization power over problem instances. For example, compare random guessing with the constant algorithm, which always returns some predictor $f_0 \in \mathcal{F}$ independently of the data. The no free lunch theorem tells us that, on average over all possible problem instances $f^* \in \mathcal{F}$, both algorithms have the same performance. But they strongly differ on particular instances: the constant algorithm is exact for $f^* = f_0$ but always misses for $f^* \neq f_0$, whereas random guessing has the same (low) performance on every problem instance. Does this now mean that learning is impossible and we could have just guessed instead of developing powerful algorithms such as deep neural networks? Fortunately, this is not the case, because the crux about learning lies in the clause "uniformly averaged over all possible problem instances". In real-world learning problems, we can often exclude some solutions or express preferences of some solutions over others, even before seeing any data. Formally, we can posit that the solution lies in some smaller subset $f^* \in \mathcal{F}_{\text{sub}} \subset \mathcal{F}$. This additional information allows us to obtain algorithms that perform above average on the relevant set $\mathcal{F}_{\text{sub}}$. This does not contradict the no free lunch theorem, which only enforces that these algorithms perform below average on the irrelevant set $\mathcal{F} \setminus \mathcal{F}_{\text{sub}}$. As a trivial example, assume we already know that the solution is f_0. As discussed above, the algorithm that constantly returns f_0 is not superior to random guessing over all solutions because it performs suboptimally on $\mathcal{F} \setminus \{f_0\}$, but *restricted on* $\{f_0\}$ it is optimal. We commonly refer to this a priori restriction of the solution space as *inductive bias*, because it biases the inductive inference task towards certain solutions, independently of any data. In the above sense, inductive bias is necessary for generalization and any statement of the form "algorithm A is superior to algorithm B" is ultimately a statement about their inductive biases. The word 'bias' is often negatively connoted, for example in statistics it describes the average discrepancy between a target parameter and an estimator. However here, inductive bias is not a nuisance that we try to get rid of, but a necessary ingredient for successful learning.

The bias-complexity tradeoff The success of learning depends on how well the inductive bias of an algorithm matches the problem instance. To measure this match and the related generalization power of an algorithm, consider the following error decomposition. For a learning problem with Bayes classifier $f^* \in \mathcal{F}$ and corresponding Bayes risk $\mathcal{R}^*$, the inductive bias of an algorithm is specified by learning from a restricted hypothesis class $\mathcal{F}_{\text{sub}} \subset \mathcal{F}$. Given a dataset D, the output of the algorithm is $\hat{f}(D) \in \mathcal{F}_{\text{sub}}$. Additionally, let $f^*_{\text{sub}} \in \mathcal{F}_{\text{sub}}$ be an optimal predictor in the restricted hypothesis class, defined similarly to the Bayes classifier as $f^*_{\text{sub}} \in \arg\min_{f \in \mathcal{F}_{\text{sub}}} \mathcal{R}(f)$. Then the difference between the risk of the learned predictor and the optimal risk can be decomposed as

$$\mathcal{R}(\hat{f}(D)) - \mathcal{R}^* = \underbrace{\mathcal{R}(\hat{f}(D)) - \mathcal{R}(f^*_{\text{sub}})}_{\text{estimation error}} + \underbrace{\mathcal{R}(f^*_{\text{sub}}) - \mathcal{R}^*}_{\text{approximation error}} .$$

The *approximation error* describes the mismatch between the inductive bias $\mathcal{F}_{\text{sub}}$ and the problem instance f^*, which is independent of the observed data D. The *estimation error* describes the hardness of the learning problem restricted on the hypothesis class $\mathcal{F}_{\text{sub}}$ and is a random quantity over the dataset D. A stronger inductive bias (that is, a less complex $\mathcal{F}_{\text{sub}}$) reduces the estimation error because it reduces the variability of the learning algorithm. At the same time, a stronger inductive bias can only increase the approximation error when good predictors are removed from the hypothesis class. We are therefore faced with a tradeoff in the complexity of the hypothesis class, called the *bias-complexity tradeoff*. This is similar to the bias-variance tradeoff under the mean-squared error. Predictors with bad generalization power are commonly distinguished into two categories based on the strength of the inductive bias. A weak inductive bias (large $\mathcal{F}_{\text{sub}}$) has low approximation error, but large estimation error. The learned predictor often fits the noise of the data set instead of extracting the underlying pattern. This is called *overfitting*. On the other hand, a strong but mismatched inductive bias has low estimation error, but large approximation error. The learned predictor is not complex enough to capture the underlying pattern of the data, because it is restricted to the wrong hypothesis class. This is called *underfitting*. Ideally, we impose as much inductive bias as possible, as long as we can guarantee that this bias matches the actual solution. This is easier said than done since our lack of prior knowledge about the solution lies at the core of the learning problem. The next section discusses what constitutes inductive bias and introduces a general guiding principle for choosing reasonable biases in real-world problems.

1.2 Occam's razor: a guiding principle for choosing the inductive bias

Before we come to a general principle for choosing the inductive bias, let us take a look at what else constitutes this bias besides a restriction of the hypothesis class $\mathcal{F}_{\text{sub}} \subset \mathcal{F}$

that was discussed in the previous section. This already starts with the data set: in theory it seems unambiguous to talk about Euclidean features $x \in \mathbb{R}^d$ with binary labels $y \in \{0, 1\}$, but in practice we have to choose which features to include and which to dismiss prior to any learning. For example if we want to predict what a customer wants to buy next, it seems reasonable to exclude seemingly irrelevant input features like 'hair color', but include others like 'age'. The same choice has to be made for the set of values that the label y can take. Missing practically relevant features and labels or including too many irrelevant ones can significantly harm the performance of the predictor. Other common practices include data pre-processing such as normalization or dimensionality reduction and data cleaning, which refers to correctly formatting the data and removing duplicates or incomplete data.

Another choice that fundamentally affects the inductive bias of an algorithm is the loss function. In practice, many different losses have been introduced to solve different tasks (Jadon, 2020; Wang et al., 2022). The loss is in fact related to two different kinds of ill-posed problems. The first problem arises on the distribution level when we want to learn a deterministic predictor instead of the full conditional distribution. Choosing a loss resolves this issue by defining the Bayes classifier. Richardson (2022) proposes to make this arbitrary choice not based on heuristics and instincts, but instead by choosing a certain model, which is arguably a more objective way to describe the underlying bias. The loss then arises naturally as an inconsistency of this model. The second ill-posed problem that the loss can address is that of finite samples, which was discussed in the previous section. Since the loss that defines the Bayes classifier is known, it seems reasonable to use the same loss during training. A corresponding general framework is *empirical risk minimization* (Vapnik, 1999), where the predictor $\hat{f}$ based on a data set $D = \{(x_1, y_1), \dots, (x_n, y_n)\}$ is given by

$$\hat{f}(D) \in \arg\min_{f \in \mathcal{F}} \mathcal{R}_D(f)\,, \quad \text{where } \mathcal{R}_D(f) = \frac{1}{n}\sum_{i=1}^{n} l(f(x), y) \text{ denotes the } \textit{train risk}.$$

The idea is that the train risk based on the observed data points serves as a proxy for the test risk based on unseen test points, which is the actual target of learning. More generally, the loss describes how compatible a predictor is with the observed data. This general principle also includes other approaches that make distributional assumptions about the data. For example, maximum likelihood estimation based on a parametric family of distributions $\{p_w(y|x) \mid w \in \mathcal{W}\}$ can be interpreted as empirical risk minimization by rewriting

$$\hat{w} \in \arg\max_{w \in \mathcal{W}} \prod_{i=1}^{n} p_w(y_i|x_i) = \arg\min_{w \in \mathcal{W}} \frac{1}{n}\sum_{i=1}^{n} \underbrace{-\log p_w(y_i|x_i)}_{\text{loss of } w \text{ at } (x_i, y_i)}\,.$$

To summarize, we can impose inductive bias by choosing the data set, the loss function or a similar compatibility score between predictor and observed data, and by restricting the hypothesis class. However, this inductive bias is often still not strong enough and

results into overfitting. In extreme cases it is not even strong enough to uniquely specify the output of the learning algorithm. For example in underdetermined linear regression, there are infinitely many solutions that minimize the empirical risk. Therefore, we now turn to the general principle of Occam's razor.

Occam's razor William of Ockham, an English scholastic philosopher and theologian of the 14th century, is credited with the statement

"Entities should not be multiplied beyond necessity."

This philosophical guiding principle for problem-solving is commonly referred to as *Occam's razor*. It states that when presented with competing explanations, the simplest one should be preferred. Competing explanations is precisely the issue that we face in machine learning: after specifying some way to assess the compatibility between a predictor and the observed data, for example by a loss function, we are usually left with infinitely many predictors that fit the data well. Some of those predictors may generalize, but most do not and the question is how to distinguish them. In fact, a good fit to the training data itself is a vacuous property, because any possible predictor can be changed to memorize the training data to achieve a perfect fit. Only fitting the data therefore often leads to overfitting. It is the combination with other principles that enables generalization, and Occam's razor is such a principle that tells us to choose simple predictors. We have already encountered a direct way of implementing this principle into a learning algorithm, namely restricting the hypothesis class to some subset $\mathcal{F}_{\text{sub}} \subset \mathcal{F}$ of simple functions. For example, linear regression can only return linear functions, which are arguably simple. This hard restriction is a special instance of the more general *complexity measures*. A complexity measure $\Omega\colon \mathcal{F} \to \mathbb{R}_{\geq 0}$ penalizes every predictor for being complex (that is, not simple), prior to seeing any data. This complexity measure can be used to break ties between predictors that fit the data equally well and therefore adhere to Occam's razor of choosing the simplest explanation. While this sounds good, the problem is that Occam's razor does not tell us what 'simple' means. Another important caveat is that, under the light of the no free lunch theorem, this principle is not guaranteed to produce good predictors. Only the combination of being biased towards simple functions and the true explanation being simple in the same sense leads to good predictors. In the remainder of this section, I discuss some common implementations of Occam's razor for various machine learning tasks and highlight one natural choice for complexity measures, the Bayesian Occam factor.

Common implementations of Occam's razor I only mention a small selection of examples here since it is impossible to exhaustively cover the large variety in which algorithms can impose assumptions. Two frameworks that organically incorporate the idea of a complexity measure are empirical risk minimization and Bayesian inference. Empirical risk minimization simply adds the complexity measure Ω as a regularizer to

the train risk. The new objective function thus becomes $\mathcal{R}_D(f) + \lambda \cdot \Omega(f)$, where the factor $\lambda \geq 0$ is used to trade off data fit with complexity of a predictor. This approach is called *regularized risk minimization*. A similar framework is Bayesian inference, in which the family of conditional distributions from maximum likelihood estimation is augmented by a prior distribution over the solutions. The prior distribution acts as a complexity measure because it, as the name suggests, describes our a priori preference for solutions before seeing any data. Beyond frameworks, a central, but strong simplifying assumption of many machine learning algorithms is that of independent and identically distributed training data. Assuming that training and test data are independent samples from the same distribution $\mathbb{P}_{X,Y}$ greatly restricts the space of all possible joint distributions. Generalizations of this assumption include time series data and distribution shift problems. Another common assumption is that a function is simple if it is smooth. As such, many learning algorithms have a preference for smooth functions by restricting the hypothesis class directly, for example to linear functions, or by penalizing non-smooth functions in the form of a complexity measure. The notion of smoothness itself depends on the arbitrary choice of metric or a corresponding similarity measure. Other hard restrictions of the hypothesis class include, for example, a low rank assumption for matrices, or the manifold hypothesis. The latter assumes that high-dimensional data lie on an (unknown) manifold of significantly lower dimension.

The Bayesian razor and the Occam factor The Occam factor is a particularly natural complexity measure that arises from a Bayesian problem formulation. This paragraph follows the discussion in MacKay (2002) at the example of two hypotheses H_1, H_2 (for example H_1 : Gaussian distributions vs. H_2 : all continuous distributions) with corresponding solutions $h_i \in H_i$. Choosing a hypothesis simply based on the best fit h_i^{ML} to the observed data D, that is, $\arg\max_{H_i} P(D|h_i^{ML}, H_i)$, would lead to overfitting, because more complex hypotheses allow for a better fit. The Bayesian approach to this problem is to introduce prior distributions: one on the solution level $P(h_i|H_i)$ to fit the data under a fixed hypothesis, and one on the hypothesis level $P(H_i)$. Applying Bayes' theorem to the latter yields the posterior uncertainty over hypotheses given the data $P(H_i|D) \propto_{H_i} P(H_i)P(D|H_i)$. The prior over hypotheses allows us to impose inductive bias directly, but this approach yields a complexity measure even if we stay agnostic with $P(H_1) = P(H_2)$. This yields $P(H_i|D) \propto_{H_i} P(D|H_i)$, where the second term is called the *evidence* in Bayesian inference and describes how likely it is to observe the data D under the given hypothesis H_i. Contrary to the best fit, the evidence naturally incorporates a notion of Occam's razor and prefers the simpler hypothesis. This is because the more complex hypotheses spread their mass over more solutions, which leads to smaller evidence, even if there are particular solutions that fit the data well. To understand this idea in more detail, we can further decompose the evidence $P(D|H_i) = \int_{h_i} P(D|h_i, H_i)P(h_i|H_i)\,\mathrm{d}h_i$ by replacing the integrand with a flat curve at the best fit h_i^{ML} over an appropriate volume $\sigma_{h_i|D}$ and further replacing the

prior with a uniform distribution over another volume σ_{h_i}. This yields

$$\underbrace{P(D|H_i)}_{\text{Evidence}} \approx \underbrace{P(D|h_i^{ML}, H_i)}_{\text{Best likelihood fit}} \times \underbrace{\frac{\sigma_{h_i|D}}{\sigma_{h_i}}}_{\text{Occam factor}}$$

The Occam factor describes the ratio of posterior to prior uncertainty and is ≤ 1. This factor gets small for complex models with large prior support, because the model zeroes in on the much smaller subset of solutions that fit the data, that is, $\sigma_{h_i|D} \ll \sigma_{h_i}$. For simpler models, there is less reduction in uncertainty, which results in larger factors. In this decomposition, we can therefore clearly see that a hypothesis is rewarded for its ability to fit the data, but also punished for its complexity by the Occam factor.

1.3 Explicit inductive bias in deep learning

In the next three sections of the introduction, I discuss inductive bias at the example of deep learning. I start with explicit ways in which inductive bias is encoded in this section, then move to implicit inductive bias in Section 1.4, and conclude with attempts of reverse-engineering this implicit bias in Section 1.5. Uncovering implicit bias is also the goal of my work in Chapters 2 and 3; my contributions are reviewed in Section 1.6.

Deep learning has come a long way from Rosenblatt's first perceptron (Rosenblatt, 1961) to powerful architectures such as transformers (Vaswani et al., 2017) and is now state of the art in many machine learning tasks (LeCun et al., 2015). The main way in which deep learning explicitly imposes inductive bias is through a structural prior in the form of an architecture, which restricts the hypothesis class of representable functions. The structural prior commonly follows the principle of compositionality, in which complex objects are made up of simple objects and their interactions (Battaglia et al., 2018). An additional hierarchical structure allows to process information on different levels of granularity. This resembles the way in which we humans understand the world (McClelland and Rumelhart, 1981; Navon, 1977). For example, we do not simply memorize every valid sentence in an unstructured manner. Instead, we only memorize the significantly smaller set of valid words and some rules to combine them. We also maintain the hierarchy that letters are part of words, words are part of sentences, and sentences are parts of texts. In deep learning architectures, compositionality is realized by individual neurons that interact with each other through a pre-defined set of edges. Neurons are grouped into layers, which can be stacked hierarchically. As such, the types of interactions are fixed as part of the structural prior, but their strengths are learnable. An important structural prior in deep learning architectures is given by invariances:

Inductive bias through invariances Formally, a neural network is a function $f_\theta \colon \mathcal{X} \to \mathcal{Y}$ from some input space $\mathcal{X}$ to some output space $\mathcal{Y}$ with learnable weights

$\theta \in \Theta$. We say that a function f_θ is *invariant* under a group G that acts on the input space $\mathcal{X}$ if $f_\theta(g \circ x) = f_\theta(x)$ for all $g \in G, x \in \mathcal{X}$. Such an invariance can strongly restrict the function f_θ, because it equivalently states that f_θ is constant on every set $\{g \circ x \mid g \in G\}$ for $x \in \mathcal{X}$. Network architectures that enforce invariances a priori restrict the hypothesis class of learnable functions and as such impose inductive bias. For example, assume we want to learn a function that labels an image as 'cat' or 'dog'. If we hold the prior belief that the label of an image does not depend on the position of the object, we are formally saying that the true classifier is invariant to translations. This already dismisses a vast amount of potential solutions. We can then impose this bias by choosing a network architecture that only contains functions which respect this invariance. A related concept to invariance is *equivariance* with similar implications. Formally, if the group G also acts on the output space $\mathcal{Y}$, a function f_θ is equivariant under a group G if $f_\theta(g \circ x) = g \circ f_\theta(x)$ for all $g \in G, x \in \mathcal{X}$. This means that a shifted input produces a similarly shifted output. For example in image segmentation, we expect that moving the object in an image should simultaneously move its outline.

In deep learning, different invariances can be enforced by certain types of architectures. This is achieved by weight sharing and restricted connections between neurons. Translation invariance is central to the field of computer vision for learning from image or video data. It is realized by convolutional layers, one of the basic building blocks in convolutional neural networks (Fukushima and Miyake, 1982; LeCun et al., 1989, 2010). After the large success of convolutions on Euclidean data, they are now also translated to non-Euclidean domains such as manifold or graph data (Bronstein et al., 2017). Temporal invariance in the form of the Markov property is enforced by recurrent neural networks, which process a sequence of inputs (Elman, 1990; Lipton, 2015). In the case of unstructured data such as a graph, which consists of a set of nodes and edges, a common desideratum is permutation invariance. This means that the function does not depend on the arbitrary ordering of the nodes and is a central property of graph neural networks (Gori et al., 2005; Wu et al., 2021). It has been empirically validated that structural priors alone already describe a useful inductive bias for real-world problems. For example, Lempitsky et al. (2018) show that untrained, randomly initialized convolutional networks capture low-level image statistics of natural images.

Beyond the invariances described above, there are many others that we might assume for real-world problems, but for which no obvious architecture exists. In such cases, a common workaround is data augmentation (Shorten and Khoshgoftaar, 2019). Here, each data point x in the training set is augmented by a set of new points $\{g_1 \circ x, \ldots, g_j \circ x\}$ for transformations $g_1, \ldots, g_j \in G$, all of which get the same label as x. Instead of strictly enforcing the invariance $f_\theta(g \circ x) = f(x)$ for all $g \in G$, this approach merely promotes it. In return, it is directly applicable to any kind of transformation that can be simulated. Common basic geometric transformations include flipping, cropping, rotations, kernel filters (sharpen and blur), color space manipulations, and noise injections. But also more elusive transformations such as the 'style' of an image can be implemented with the help of a separate style transfer network (Gatys et al., 2015).

Despite all the above structural priors in deep learning architectures, it is not clear that they alone impose sufficiently strong inductive biases to explain the generalization power. In fact, while past approaches had stronger restrictions in the form of hand-crafted features, modern approaches rely on end-to-end architectures with minimal restrictions. Why does the remarkable flexibility of architectures that can approximate any function (Hornik et al., 1989) not immediately lead to overfitting? A possible answer is that deep learning does not only impose inductive bias *explicitly* through its architecture, but also *implicitly* through other aspects of the learning procedure. A discussion of implicit biases in deep learning follows in the next section.

1.4 Implicit inductive bias in deep learning

Deep learning models are often trained to zero training error and even continue training after. Yet, they successfully generalize to unseen test points instead of overfitting to the data. In an influential work, Zhang et al. (2017, 2019) put a spotlight on this disconnect between theory and practice by showing that neural networks can easily fit pure noise. Belkin et al. (2018) reports similar results for kernel machines, which can be viewed as a theoretically more tangible special instance of neural networks. These results imply that common capacity measures such as the Rademacher complexity or VC dimension are large, which makes their associated generalization bounds vacuous. Classical learning theory can therefore not explain why neural networks generalize. Other common regularization techniques can improve generalization, but are also not its cause. This includes both direct techniques such as weight decay or dropout (Srivastava et al., 2014) and more indirect techniques such as early stopping or batch and layer normalization (Ba et al., 2016; Ioffe and Szegedy, 2015). There needs to be another fundamental reason why neural networks generalize (Belkin, 2021).

Inductive bias of overparameterized systems Overparameterization is a key ingredient for explaining why deep learning generalizes. Consider the example of linear regression under the square loss with d dimensions and n examples, for which the design matrix is full rank. For $d \leq n$ the system is overdetermined and has a unique solution, which means that the choice of hypothesis class (linear functions) and loss uniquely determine the algorithm. For $d < n$, the system is underdetermined and has infinitely many solutions that minimize the training loss. An algorithm is therefore not well-defined by hypothesis class and loss alone. It has to additionally choose which of these minimizing solutions it returns. This is precisely the situation in deep learning, where vastly overparameterized architectures are trained to interpolate the data. Deep learning procedures commonly do not specify *explicitly* which of the many possible minimizers they return. Instead, the choice is governed implicitly by the architecture, all employed learning techniques, and the optimization algorithm. I therefore refer to the collection of these choices as *implicit* inductive bias in deep learning.

Classical learning theory is concerned with the underparameterized setting. Here, the training error simply decreases with model complexity, whereas the test error follows a U-shape. This results from the typical bias-variance tradeoff, where too simple models underfit due to large bias, too complex models overfit due to large variance, and a sweet spot exists in between. Belkin et al. (2019) extend this curve to the overparameterized setting in order to explain the puzzling generalization behavior of modern complex systems: while the training error stays zero when further increasing the model complexity beyond the interpolation threshold, the test error can actually decrease again. This phenomenon is referred to as *double descent*. The second descent happens only if the additional inductive bias for choosing a minimizer is aligned with the true solution. To continue the example of linear regression where there are multiple interpolators for $d > n$, a common choice is the min-norm interpolator. This describes the additional inductive bias of preferring smooth solutions, where smoothness is measured by the corresponding norm. At the interpolation threshold $d = n$, there is only one non-smooth interpolator with bad generalization. But as d grows larger than n, the set of interpolators grows and includes smoother options, which allows the additional inductive bias to kick in. In high dimensions, smoothness and interpolation become more compatible and the test risk descends for a second time. This was demonstrated by Belkin et al. (2018) empirically for random Fourier feature models whose smoothness bias is described by the norm of a corresponding kernel, deep neural networks optimized with gradient descent, and several other overparameterized models.

Benign overfitting. The apparent contradiction of choosing interpolators with good generalization properties is compactly described by the oxymoron *benign overfitting*. After observing this phenomenon empirically in deep learning and other models such as AdaBoost and random forests (Wyner et al., 2017), researchers have tried to reproduce benign overfitting theoretically in simpler models with special focus on the min-norm interpolator. Hastie et al. (2022) compute the risk of ridge regression solutions and the min-norm interpolator in the proportional asymptotic regime, which also produces a double descent curve. Muthukumar et al. (2020) give corresponding non-asymptotic results and Bartlett et al. (2020) considers regression over general Hilbert spaces. To understand under which conditions overfitting can be benign, Shamir (2022) considers the min-norm interpolator and the max-margin classifier for different losses and Bubeck and Sellke (2021) show that strong overparameterization is necessary for the existence of smooth interpolators. A complementary line of work addresses the question of whether overfitting can be not only benign, but also necessary for generalization. Feldman (2020) postulates a heavy-tailed model in which memorization of low frequency examples is necessary for generalization, which is empirically validated for deep learning tasks by Feldman and Zhang (2020). Similarly, Brown et al. (2021) construct examples of next-symbol prediction problems and multiclass classification for which memorization is necessary in an information-theoretic sense. Kobak et al. (2020) show that the optimal ridge regularization can be zero and Cheng et al. (2022) investigate the cost of not interpolating.

To summarize the situation from an inductive bias perspective, interpolation per se does not fully specify a learning algorithm in overparameterized hypothesis classes such as neural networks. The algorithm and its generalization ability depend on an additional choice, which is often made only implicitly. The next section aims to understand this implicit choice in deep learning and reformulate it in an explicit way.

1.5 Reverse-engineering inductive bias in deep learning

Ultimately, a deep learning model trained until interpolation is given a data set D and returns a predictor $\hat{f}(D)$ from the set of all interpolating predictors $\mathcal{F}_{\text{int}} \subset \mathcal{F}$. But for overparameterized models, $\mathcal{F}_{\text{int}}$ often contains more than one predictor, so which is chosen by the algorithm? Ideally, we would like to understand the algorithm's choice as being guided by a complexity measure $\Omega(f)$ that describes its inductive bias. That is, the goal is to find a function $\Omega\colon \mathcal{F} \to \mathbb{R}_{\geq 0}$ for which the algorithm satisfies

$$\hat{f}(D) = \underset{f \in \mathcal{F}_{\text{int}}}{\arg\min}\, \Omega(f)\,.$$

A well-known example of implicit regularization is gradient descent on linear functions under the square loss. There exist multiple solutions if the system is underdetermined, but gradient descent (with appropriate initialization and step size) converges to the specific solution with minimal norm. This means that the corresponding implicit complexity measure is given by the ℓ_2 norm, which motivates why the literature is often concerned with the min-norm interpolator. To arrive at such a result, one needs to analyze the specific update rule of the optimization procedure. In this simple case the update rule preserves the property of lying in the row space of the design matrix, and the min-norm interpolator is the unique solution with this property. In classification, the chosen loss functions often do not have a unique root (such as the square loss), but instead monotonically decrease to 0 in the limit (such as the logistic loss). This means that no finite minimizer exists and optimizers necessarily diverge to reach 0 training loss. However, Soudry et al. (2018) show that the direction of the linear predictor converges to that of the max-margin solution for separable data, which is extended to non-separable data by Ji and Telgarsky (2019).

The implicit bias of optimization does not only depend on the chosen optimization procedure and hypothesis class, but also on its parameterization. This is illustrated at the example of matrix factorization in underdetermined matrix regression. As discussed above, gradient descent on the space of matrices $X \in \mathbb{R}^{n \times n}$ directly leads to the implicit ℓ_2 regularizer $\|X\|_2$. The reparameterization $X = UU^T$ with $U \in \mathbb{R}^{n \times d}$ for $d \geq n$ does not change the hypothesis class, but it does change the update rule for gradient descent on U and therefore its inductive bias. Gunasekar et al. (2017) shows that this can change the implicit regularization to the nuclear norm in certain settings and conjectures that this holds in general. Arora et al. (2019) extends this result to deep matrix factorizations, but refutes the conjecture in favor of implicit rank regular-

ization, which is confirmed by Li et al. (2021). Gunasekar et al. (2018a) investigate the implicit regularization of linear methods for different losses, different variants of gradient descent, and different hyperparameters such as momentum and step size.

Optimization bias in neural networks To understand what complexity measure drives the generalization in actual neural networks, Neyshabur et al. (2017) and Jiang et al. (2020) conduct empirical studies in which they compare generalization performance with several candidate measures for trained networks, but find no definite explanation. One of these candidate measures is the sharpness of the minimum, for which Keskar et al. (2017) empirically observe that stochastic gradient descent with larger batch size leads to sharper minima. To make definite, provable statements about the implicit complexity measure, strong assumptions are often necessary. These include simplified optimization procedures, for example infinitesimal step size because the gradient flow is more amenable to analysis, or linear activations in neural networks, which simply reparameterize linear predictors. In this sense, the previously discussed results on (deep) matrix factorization can be viewed as results on fully connected neural networks with linear activations. The training behavior of neural networks is hereby broadly categorized into the *kernel regime* and the *rich regime*. The kernel regime is characterized by an invariance of the optimization trajectory during training. In this regime, training the neural network behaves like training a kernel machine with respect to the neural tangent kernel (Jacot et al., 2018), which depends on the random initialization. The implicit complexity measure is therefore given by the ℓ_2 norm of the corresponding reproducing kernel Hilbert space. In contrast, many neural networks have been shown to produce more sparsity-inducing solutions such as implicit rank regularization, which cannot be described by a kernel machine. This regime is described as the rich regime. The transition between kernel and rich regime can be governed by various hyperparameter choices for the deep learning architectures, for example the layer width (Du et al., 2019) or the initialization scale (Chizat et al., 2019). In linear diagonal neural networks, Woodworth et al. (2020) show for regression that the transition is controlled by an interaction of initialization scale, width, and training error. Moroshko et al. (2020) extend these results to classification. Besides fully connected and diagonal linear neural networks, Yun et al. (2021) give results for a generalized tensor formulation that also includes other architectures such as linear convolutional networks (Gunasekar et al., 2018b; Jagadeesan et al., 2022). Azulay et al. (2021) describe a general technique that encompasses many previous results by recasting the gradient flow dynamics as infinitesimal mirror descent.

Negative results In some settings it is not only hard to find an implicitly minimized complexity measure, but provably impossible. This suggests that the framework of regularized risk minimization might be too restrictive to capture the inductive bias of some algorithms. For matrix factorization, there exist learning problems for which every (quasi-)norm diverges, hence no such norm can be implicitly minimized (Razin

and Cohen, 2020). For stochastic gradient descent in convex optimization problems, there exist learning problems for which no reasonable complexity measure explains the generalization (Dauber et al., 2020). Even more dramatically, Vardi and Shamir (2021) show that, already in the simple case of a single ReLU neuron, the implicit regularization of gradient flow cannot be described by any explicit function of the model parameters. They further show that the only generally valid inductive bias in this setting is a balancedness condition (Du et al., 2018), which states that the layer norms remain invariant throughout optimization. However, it is generally unclear how this restriction on the parameter level translates to restrictions on the corresponding functions.

1.6 Thesis contributions

In the introduction, I described that a crucial part of the inductive bias in deep learning is specified only implicitly. The discussed works attempt to make this bias explicit by showing that gradient-based optimization implicitly minimizes some complexity measure, which describes the inductive bias. My work has the same general goal of uncovering implicit inductive bias and was created in collaborations, see Section 1.6.1. Instead of analyzing bias from optimization as above, we investigate the bias of a specific algorithm that uses deep learning as an intermediate step. Next, we present a generic algorithm to uncover the inductive bias that is incurred by approximations in Bayesian inference. Specifically, the next three chapters are structured as follows.

In Chapter 2, we search for the hidden inductive bias of the particular graph generative model NetGAN (Bojchevski et al., 2018). A graph generative model takes a graph as input and is supposed to generate new graphs that "look like" the input graph. While most classical models focus on few hand-selected graph statistics and are too simplistic to reproduce real-world graphs, NetGAN recently emerged as an attractive alternative: by training a GAN to learn the random walk distribution of the input graph, the algorithm is able to reproduce a large number of important network patterns simultaneously, without explicitly specifying any of them. We investigate the implicit bias of NetGAN. We find that the root of its generalization properties does not lie in the GAN architecture, but in an inconspicuous low-rank approximation of the logits random walk transition matrix. Step by step we can strip NetGAN of all unnecessary parts, including the GAN, and obtain a highly simplified reformulation that achieves comparable generalization results, but is orders of magnitudes faster and easier to adapt. Being much simpler on the conceptual side, we reveal the implicit inductive bias of the algorithm—an important step towards increasing the interpretability, transparency and acceptance of machine learning systems.

In Chapter 3, we investigate the inductive bias of approximate Bayesian inference. This is unnecessary in full Bayesian inference, where the bias is directly described by the prior distribution. However, full Bayesian posteriors are rarely analytically tractable,

which is why real-world Bayesian inference heavily relies on approximate techniques. Approximations generally differ from the true posterior and require diagnostic tools to assess whether the inference can still be trusted. We investigate a new approach to diagnosing approximate inference: the approximation mismatch is attributed to a change in the inductive bias by treating the approximations as exact and reverse-engineering the corresponding prior. We show that the problem is more complicated than it appears to be at first glance, because the solution generally depends on the observation. By reframing the problem in terms of incompatible conditional distributions we arrive at a natural solution: the *Gibbs prior*. The resulting diagnostic is based on pseudo-Gibbs sampling, which is widely applicable and easy to implement. We illustrate how the Gibbs prior can be used to discover the inductive bias in a controlled Gaussian setting and for a variety of Bayesian models and approximations.

The last work in Chapter 4 concerns the inductive bias of causal learning. The introduction discussed the inductive bias necessary to solve the inverse problem of generalizing from finite samples to properties of the underlying distribution. Causal learning faces the additional inverse problem of generalizing from the observational distribution to the underlying causal model. This second step also requires inductive bias in the form of additional assumptions. A common assumption and the focus of this thesis is the independence of causal mechanisms (ICM). In our paper "Interpolation and Regularization for Causal Learning", which is not part of this thesis, we investigate whether benign overfitting (see Section 1.4) also occurs for causal learning under the ICM. We find that this is indeed possible and that the behavior is governed by the confounding strength. The work presented in Chapter 4 is a technical follow-up on the above paper that analyzes estimators for confounding strength from observational data under the ICM. We find that existing estimators are generally biased and propose a consistent estimator based on tools from random matrix theory.

1.6.1 Publications

This thesis is based on the following publications.

Chapter 2: Rendsburg, L., Heidrich, H., von Luxburg, U. (2020) NetGAN without GAN: From random walks to low-rank approximations. In *International Conference on Machine Learning (ICML)*.

Chapter 3: Rendsburg, L., Kristiadi, A., Hennig, P., von Luxburg, U. (2022) Discovering inductive bias with Gibbs priors: A diagnostic tool for approximate Bayesian inference. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*.

Chapter 4: Rendsburg, L., Vankadara, L. C., Ghoshdastidar, D., von Luxburg, U. (2022) A consistent estimator for confounding strength. *arXiv preprint arXiv:2211.01903 (under review).*

During my PhD, I co-authored three other papers that do not appear in this thesis.

The first work compares centrality measures that are based only on ordinal triplet comparisons. The project started during my internship prior to the PhD, during which it was then completed.

Rendsburg, L., Garreau, D. (2021) Comparison-based centrality measures. In *International Journal of Data Science and Analytics.*

I was one of the two main contributors to the following paper on benign overfitting in causal learning.

Vankadara, L. C., Rendsburg, L., von Luxburg, U., Ghoshdastidar, D. (2022) Interpolation and Regularization for Causal Learning. In *Neural Information Processing Systems (NeurIPS).*

The following work translates the graph-theoretic concept of tangles to a practical clustering algorithm for machine learning. In the first version, Solveig Klepper and I have been the main contributors. Solveig Klepper then took the lead for the latest version.

Klepper, S., Elbracht, C., Fioravanti, D., Kneip, J., Rendsburg, L., Teegen, M., von Luxburg, U. (2022) Clustering with Tangles: Algorithmic Framework and Theoretical Guarantees. *arXiv preprint arXiv:2006.14444 (under review).*

Chapter 2

NetGAN without GAN: From Random Walks to Low-Rank Approximation

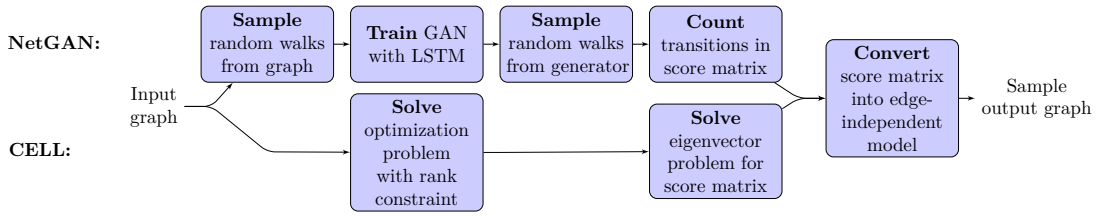


Figure 2.1: Pipelines for NetGAN (upper path) and our proposed method CELL (lower path). CELL is a condensed version of NetGAN that bypasses the expensive sampling steps and replaces the GAN with an optimization problem.

A graph generative model is a mechanism to achieve the following task: for a given input graph (or a set of input graphs), generate new graphs that have a similar structure as the input graph. The mechanism is supposed to slightly perturb the graph, but should not change its characteristic structure (such as the community structure, the characteristic path lengths, etc). Being able to create perturbed copies of a graph is useful in many different scenarios, for example: comparing a small sample of brain networks for Alzheimer patients (just one, in the extreme case) to a large population of healthy subjects, making robustness statements about a climate network by running a sensitivity analysis on perturbed copies, or performing a generic bootstrap analysis.

A recent graph generative model that has received a lot of attention is NetGAN (Bojchevski et al., 2018). First, it samples a set of random walks from the input graph to train a GAN (Goodfellow et al., 2014), whose generator learns to produce node-sequences that resemble random walks over the input graph. Generated graphs are

then obtained as reconstructions based on these sequences. The inherent assumption of this approach is that random walks describe graphs in a reasonably holistic way: local statistics such as motifs are observable in the individual random walks, while global statistics such as cluster structure and diameter are encoded in the distribution over random walk sequences. As opposed to other approaches, NetGAN does not make any explicit model assumptions; rather, it is supposed to implicitly learn many local and global graph statistics simultaneously by reproducing random walk statistics. However, if the goal is to generalize (perturb) the input graph, there has to be an implicit bias as to which type of generalization (perturbation) is preferred (no free lunch). The goal of our work is to characterize this bias of NetGAN, which we will achieve by reformulating it in terms of a distance function between graphs. This formulation provides insights on the influence of design choices and model parameters, such as the length of the random walks. Scrutinizing the NetGAN architecture, we observe that many of its components can be considerably simplified. Step by step we strip all the unnecessary parts until we are left with the only crucial ingredient, a low-rank approximation of the logits random walk transition matrix. Our main contributions are:

- **Reformulation of NetGAN.** We reformulate NetGAN as a low-rank approximation with respect to the Kullback-Leibler divergence between transition matrices, which requires neither a GAN nor any sampling.
- **Huge speedup.** Our algorithm retains the generalization performance of NetGAN, but runs in seconds instead of hours. See Table 2.1 for a comparison of training times.
- **Transparency.** Our algorithm is conceptually much simpler than NetGAN. This opens the possibility to analyze it theoretically, and allows for application-specific adaptions.

Table 2.1: Training time (in seconds) for NetGAN and our proposed method CELL on a variety of networks. NetGAN requires a GPU for training, while CELL runs on a CPU.

Data set (Nodes/ Edges)	NetGAN	CELL
CORA-ML (2,810/ 7,981)	7,478	21
Citeseer (2,110/ 3,668)	4,654	10
PolBlogs (1,222/ 16,779)	55,276	15
RT-GOP (4,687/ 5,529)	14,800	23
Web-EDU (3,031/ 6,474)	11,000	16

2.1 Background: NetGAN

2.1.1 Graph and random walk notation

Like NetGAN, we consider an unweighted, undirected, and connected graph $G = (V, E)$ with nodes $V = [N] = \{1, \dots, N\}$, edges $E \subseteq V \times V$, and number of edges $e(G) = |E|$. It has adjacency matrix $A \in \{0,1\}^{N \times N}$, degree vector $d \in \mathbb{R}^N$, degree matrix $D = \mathrm{diag}(d) \in \mathbb{R}^{N \times N}$, and the transition matrix for unbiased random walks on G is given by $P = D^{-1}A \in \mathbb{R}^{N \times N}$. We additionally assume G to be non-bipartite so that the random walk described by P has a unique stationary distribution $\pi \in \mathbb{R}^N$. A single random walk of length T is an ordered tuple $R = (v_0, \dots, v_T) \in V^{T+1}$, and a set of n random walks is denoted by $\mathcal{R} = \{R_1, \dots, R_n\}$. The score matrix $S(\mathcal{R}) \in \mathbb{R}^{N \times N}$ counts the transitions in $\mathcal{R}$, that is, $S_{v,w}$ equals the total number of times random walks in $\mathcal{R}$ transition from v to w. If clear from the context, we drop the dependency on $\mathcal{R}$ and write S instead of $S(\mathcal{R})$. An edge-independent random graph model, sometimes also called inhomogeneous Erdős-Rényi model, is a symmetric matrix $A^\dagger \in [0,1]^{N \times N}$ of edge probabilities. Graphs on the same vertices $[N]$ are sampled from this model by drawing $e(G)$ edges $\{v, w\}$ with probability $A^\dagger_{v,w}$ independently and without replacement. We use bold symbols, if we consider an object as a random variable (e. g. $\boldsymbol{R}$ instead of R).

2.1.2 NetGAN

In this section, we give a high-level overview of the NetGAN algorithm; for more details, we refer the reader to Bojchevski et al. (2018). NetGAN is a graph generative model: given a single input graph G, it returns graphs G' on the same set of nodes by proceeding in two main steps. First, it learns the distribution over random walks drawn from the input graph in the *learning step*. It then reconstructs the graph based on "synthetic" random walks sampled from this learned distribution in the *reconstruction step*. See Figure 2.1 for a schematic overview.

Learning step. Given an input graph G, NetGAN samples a large set $\mathcal{R}$ of random walks of fixed length T with randomly chosen start nodes. These random walks form the training set for a GAN: the generator tries to produce node sequences of length T that resemble the observed random walks in $\mathcal{R}$, while the discriminator tries to distinguish real from generated sequences. Both generator and discriminator use the Long short-term memory architecture (LSTM) (Hochreiter and Schmidhuber, 1997), and they are trained with the Wasserstein loss (Arjovsky et al., 2017). Training finishes once an early stopping criterion is met, after which the generator is used to sample synthetic random walks.

During and after training, the generator constructs each synthetic random walk $(v_0, \dots, v_T)$ in a step-by-step procedure. First, random noise z is used to initialize the memory state m_0 of the LSTM architecture and the start node v_0 of the sequence. A

function f_θ with learnable parameters θ then repeatedly updates the two values: given the current memory state m_t and node v_t, it outputs the next memory state m_{t+1} and the distribution p_{t+1} over the next node v_{t+1} in form of logits. The next node v_{t+1} is then obtained as a sample from this distribution. In equations, this update is described by

$$\begin{aligned}(m_{t+1}, p_{t+1}) &= f_\theta(m_t, v_t)\,, \\ v_{t+1} &\sim \mathrm{Cat}\left(\sigma(p_{t+1})\right)\,,\end{aligned} \tag{2.1}$$

where Cat denotes the categorical distribution and σ the softmax function, which converts the logits into a probability distribution on $[N]$. This procedure is repeated until the sequence has the desired length T.

Reconstruction step. After training is finished, NetGAN uses the generator to generate a large set of n synthetic random walks. Their transitions are counted in a joint score matrix S, which is then converted into an edge-independent random graph model $A^\dagger$ by symmetrizing and then normalizing it, that is,

$$A^\dagger_{k,l} = \frac{\max\{S_{k,l}, S_{l,k}\}}{\sum_{k',l'=1}^{N} \max\{S_{k',l'}, S_{l',k'}\}}\,. \tag{2.2}$$

To obtain the new graph G', NetGAN samples $e(G)$ edges independently and without replacement from $A^\dagger$ while preventing self-loops and isolated nodes.

2.2 What causes the generalization?

In this section, we identify those parts of NetGAN that we believe to be absolutely necessary to achieve the two goals of producing new graphs that (i) resemble the input graph by mimicking its graph statistics, but (ii) also generalize the input graph by sharing only a certain amount of its edges. The complicated GAN- and LSTM-based architecture used by NetGAN disguises its underlying bias and makes a direct analysis difficult. Therefore, we examine all the individual steps of NetGAN, not in terms of *how* they work, but *what* they aim to achieve.

The random walks? The intuition of NetGAN is that graphs with a similar random walk distribution also share many of their topological properties. In fact, as we observe in Section 2.4.4, learning the transition matrix of random walks by counting their transitions is sufficient for perfectly reconstructing the input graph. This excludes the possibility that by reducing graphs to their random walk statistics, we introduce an irreversible systematic bias.

The GAN? The role of the GAN is to learn the random walk distribution of the input graph. We prove in Section 2.4.4 that if the GAN perfectly learns the random walk distribution, NetGAN will simply reproduce the input graph instead of generalizing it. However, the results reported by Bojchevski et al. (2018) show that even if NetGAN is trained for a long time, it produces graphs that are considerably different from the input

graph as measured by edge overlap. Consequently, there has be be another mechanism that prevents the GAN from memorizing the input graph.

The LSTM? As the authors of NetGAN pointed out themselves, the LSTM architecture, which is supposed to capture long-term dependencies, seems to be an odd choice for learning Markov sequences that by construction do not have any such dependencies. It is possible that this architecture choice injects noise into the learning process, which prevents memorization of the input graph. Yet, this type of noise seems to be rather uncontrolled, and we consider it unlikely that this aspect of the LSTM cannot be replaced by a simpler, more direct mechanism.

Computational trick: low-rank approximation. What is left? In our opinion, the only component that explains why NetGAN successfully generalizes graphs is a computational trick: the LSTM is not operating on the high-dimensional space $\mathbb{R}^N$ directly. In order to reduce computational complexity, it uses learnable down- and up-projections $W_{\text{down}} \in \mathbb{R}^{N\times H}$ and $W_{\text{up}} \in \mathbb{R}^{H\times N}$ with $H \ll N$. As we derive in Section 2.2.1, these projections force the update rule of a node and memory state pair (v_t, m_t) with v_t as one-hot vector to be of the form

$$\begin{aligned} p_{t+1} &= {v_t}^\top W(m_t)\,, \\ v_{t+1} &\sim \operatorname{Cat}\left(\sigma(p_{t+1})\right)\,, \end{aligned} \tag{2.3}$$

where $W(m_t) \in \mathbb{R}^{N\times N}$ depends on m_t and has rank at most H. Because $W(m_t)$ is the transition matrix after applying σ, we refer to it as the *logit transition matrix*. NetGAN forces this matrix to have low rank, which leads us the following conjecture:

Conjecture: The key ingredient of NetGAN is to learn the random walk distribution by performing a low-rank approximation of the logit transition matrix.

To validate this conjecture, we derive a simplified method that applies this low-rank approximation directly and demonstrate its comparable performance in experiments.

2.2.1 Replacing the GAN

In this section, we inspect the high-level structure of NetGAN. The learnable projection matrices are given by $W_{\text{down}} \in \mathbb{R}^{N\times H}$ and $W_{\text{up}} \in \mathbb{R}^{H\times N}$ with $H \ll N$. Given the current node v_t as a one-hot vector and suppressing the next memory state m_{t+1} in notation, the generator f_θ can be written as

$$p_{t+1} = f_\theta\left(m_t, v_t\right) = g_\theta\left(m_t, {v_t}^\top W_{\text{down}}\right) W_{\text{up}}\,, \tag{2.4}$$

where $g_\theta \colon \mathbb{R}^H \to \mathbb{R}^H$ is the part of f_θ that operates on the low-dimensional space. We collect the row vectors $g_\theta\left(m_t, {v_t}^\top W_{\text{down}}\right)$ in a matrix $\widetilde{W}_{\text{down}}(m_t) \in \mathbb{R}^{N\times H}$ and define the product $W(m_t) \coloneqq \widetilde{W}_{\text{down}}(m_t) W_{\text{up}} \in \mathbb{R}^{N\times N}$ to obtain

$${p_{t+1}} = {v_t}^\top \widetilde{W}_{\text{down}}(m_t) W_{\text{up}} = {v_t}^\top W(m_t)\,. \tag{2.5}$$

Therefore, $W(m_t)$ simply serves as logit transition matrix for the random walks. Because of the factorization that defines $W(m_t)$, its rank is at most H.

To derive how exactly we can replace the GAN with a low-rank approximation, we first simplify the update rule in Eq. (2.5) by dropping the LSTM and with it the dependency on the memory state m_t; this is justified by the Markov property of unbiased random walks. What remains is a matrix W, whose learnable parameters are intertwined with the low-dimensional part g_θ of the generator:

$$p_{t+1} = {v_t}^\top W = g_\theta \left({v_t}^\top W_{\text{down}} \right) W_{\text{up}} \,. \tag{2.6}$$

Motivated by the assumption that the identity function $\text{Id}\colon \mathbb{R}^H \to \mathbb{R}^H$ can be represented as g_θ, we drop the structural restriction imposed by g_θ, leaving us with $W = W_{\text{down}} W_{\text{up}}$ and update rule

$$p_{t+1} = {v_t}^\top W_{\text{down}} W_{\text{up}} \,. \tag{2.7}$$

The new update of node v_t is thereby realized by sampling from the categorical distribution of the corresponding row $\sigma(W_{v_t})$, that is, $v_{t+1} \sim \text{Cat}(\sigma(W_{v_t}))$. In this form, training the GAN is equivalent to learning the random walk transition matrix directly from the parametric family $\mathcal{P} = \{\sigma_{\text{rows}}(W) \in \mathbb{R}^{N \times N} \;:\; W \in \mathbb{R}^{N \times N}, \text{rank}(W) \leq H\}$, where σ_{rows} denotes the function that applies σ to each row of a matrix. We then proceed by learning the transition matrix from this parametric family directly with the maximum likelihood approach.

2.3 Stripping NetGAN

We now gradually simplify NetGAN by stripping it of all unnecessary components in Section 2.3.1. Additionally, we observe in Section 2.3.2 that sampling random walks can be circumvented with a limit argument. This leads to our new, highly simplified method called Cross-Entropy Low-rank Logits (CELL), see Section 2.3.3 for a summary and Figure 2.1 for a schematic outline.

2.3.1 Low-rank approximation replaces the GAN

Motivated by the above conjecture, we now prune the update rule in Eq. (2.3) until we arrive at a rank-constrained optimization problem. Justified by the Markov property of unbiased random walks, we first drop the LSTM and the memory state m_t. In Section 2.2.1, we derived that the GAN learns the random walk distribution by choosing its transition matrix directly from the parametric family $\mathcal{P} = \{\sigma_{\text{rows}}(W) \in \mathbb{R}^{N \times N} : W \in \mathbb{R}^{N \times N}, \text{rank}(W) \leq H\}$, where σ_{rows} denotes the function that applies the softmax σ to each row of a matrix. The training set for this problem consists only of the transitions

of random walks in $\mathcal{R}$, and the noise random variable z plays the subordinate role of choosing the first node. This parametric family formulation defeats the purpose of using a GAN at all, which is why instead we revert to the classical maximum likelihood approach (or, equivalently, the cross-entropy loss) on $\mathcal{P}$: using the notation $(k,l) \in \mathcal{R}$ to denote all transitions (with multiple counting) of random walks in $\mathcal{R}$, the resulting problem is given by

$$\begin{aligned} \min_{W \in \mathbb{R}^{N \times N}} & -\sum_{(k,l) \in \mathcal{R}} \log \sigma_{\text{rows}}(W)_{k,l} \,, \\ \text{s. t.} \quad & \operatorname{rank}(W) \leq H \,. \end{aligned} \tag{2.8}$$

In short: instead of learning the random walk distribution by training a GAN, we approximate its transition matrix directly by solving a rank-constrained optimization problem.

2.3.2 Bypassing random walk sampling

There is another aspect of NetGAN that is somewhat puzzling: even to learn a graph of moderate size, for example the graph CORA-ML with about 3,000 vertices and 8,000 edges, NetGAN needs to sample 7,500,000 random walks of length 15 from the input graph, which are worth 112,500,000 edges. In other words, we see every edge of the input graph about 14,000 times on average — with which any edge-frequency statistic would be very close to its expected value. The same order of magnitude applies to the sampling of random walks from the generator in the reconstruction step. With that observation, a natural question is whether we can circumvent the random walk sampling, and the answer is yes. Since the random walks are only used in form of the score matrix that contains the frequency of node transitions, and this matrix converges for a large number of random walks, we can substitute the actual score matrix with its limit value. The remainder of this section formalizes this idea in Eq. (2.11) and applies it to NetGAN at both sampling steps.

Convergence of the score matrix S. First, we consider a single random walk $\boldsymbol{R} = (\boldsymbol{v_0}, \ldots, \boldsymbol{v_T})$ of length T as a random variable, whose distribution depends on the distribution $q_0 \in \mathbb{R}^N$ of the first node $\boldsymbol{v_0}$ and the transition matrix P. For $t \in \{1, \ldots, T\}$, let $Q_t \in \mathbb{R}^{N \times N}$ denote the distribution of the t-th transition $(\boldsymbol{v_{t-1}}, \boldsymbol{v_t})$ in $\boldsymbol{R}$. Its marginal $\boldsymbol{v_{t-1}}$ is distributed as $q_{t-1} \in \mathbb{R}^N$ and its conditional $\boldsymbol{v_t}|\boldsymbol{v_{t-1}}$ is distributed as P, which yields the matrix decomposition

$$Q_t = \operatorname{diag}(q_{t-1}) P \,. \tag{2.9}$$

From this perspective, counting the transitions of a single random walk R in a score matrix $S(R) \in \mathbb{R}^{N \times N}$ can be expressed as $S(R) = \sum_{t=1}^{T} \hat{Q}_t(R)$, where $\hat{Q}_t(R)$ is the empirical version of Q_t based on one sample. The score matrix $S = S(R_1, \ldots, R_n)$ based on n random walks $R_1, \ldots, R_n$ decomposes into $S = \sum_{j=1}^{n} S(R_j)$, and with the above considerations we have $S = \sum_{j=1}^{n} \sum_{t=1}^{T} \hat{Q}_t(R_j)$. By the Glivenko-Cantelli theorem for

empirical distributions, we can compute the limit of S/n for $n \to \infty$ as

$$\frac{S}{n} = \sum_{t=1}^{T} \frac{1}{n} \sum_{j=1}^{n} \hat{Q}_t(R_j) \xrightarrow[n\to\infty]{\text{a. s.}} \sum_{t=1}^{T} Q_t \,. \tag{2.10}$$

Using Eq. (2.9), the normalized right-hand side is given by $\sum_{t=1}^{T} Q_t/T = \operatorname{diag}(\rho_T)P$, where $\rho_T = \sum_{t=1}^{T} q_{t-1}/T$. In that sense, using a large number of random walks reduces to node weights ρ_T. Since the underlying graph is by assumption connected and non-bipartite, the stationary distribution π of P exists and is unique, that is, $\pi = \lim_{t\to\infty} q_t$. Hence the Cesàro mean ρ_T also converges to π as $T \to \infty$. Or, in other words: for any initial distribution q_0, the node weights induced by sufficiently long random walks are given by π. In conjunction with Eq. (2.10), we obtain the limit of the normalized score matrix

$$\frac{S}{nT} \xrightarrow[n,T\to\infty]{\text{a. s.}} \operatorname{diag}(\pi)P \,. \tag{2.11}$$

Note that we take two limits to approximate S. We take the first limit with respect to the amount of random walks n, because NetGAN samples many random walks. The second limit with respect to the length T dilutes the influence of the initial distribution ($\rho_1 = q_0$) in favor of the stationary distribution ($\lim_{T\to\infty} \rho_T = \pi$). This is appropriate because most real-world networks have small diameter, and the length $T = 15$ used in NetGAN already ensures that ρ_T is close to its limit distribution. Furthermore, the authors of NetGAN already observed that taking longer random walks increases performance. Finally, in Section 2.4.5 we will see that the information encoded in the start distribution of the random walk can be more directly incorporated by the node weights.

Replacing random walks from the input graph. The objective in Eq. (2.8) sums over all node transitions in $\mathcal{R}$. We count the transitions in a corresponding score matrix $S = S(\mathcal{R})$ to rewrite the objective function as

$$-\sum_{k,l=1}^{N} S_{k,l} \log \sigma_{\text{rows}}(W)_{k,l} \,. \tag{2.12}$$

Normalizing S does not change the minimum, and allows us to approximate it with the limit $\operatorname{diag}(\pi)P$ in Eq. (2.11). Since we consider unbiased random walks according to $P = D^{-1}A$, the stationary distribution π is proportional to the degrees d, hence $\operatorname{diag}(\pi)P \;\propto\; \operatorname{diag}(d)D^{-1}A = A$. This means that we observe every edge in every direction with the same frequency, see Lovász et al. (1993) for a survey on random walks on graphs. We use this new weighting A to define our final objective function

$$F(W) = -\sum_{k,l=1}^{N} A_{k,l} \log \sigma_{\text{rows}}(W)_{k,l} \tag{2.13}$$

and our final objective

$$\begin{aligned} \min_{W \in \mathbb{R}^{N \times N}} \quad & F(W)\,, \\ \text{s.t.} \quad & \operatorname{rank}(W) \leq H\,, \end{aligned} \tag{2.14}$$

whose solution is denoted as W^*. Note that the sum in Eq. (2.13) grows only as $\mathcal{O}(e(G))$, and we can enforce the rank-constraint in Eq. (2.14) with the factorization $W = W_{\text{down}} W_{\text{up}}$, where $W_{\text{down}} \in \mathbb{R}^{N \times H}, W_{\text{up}} \in \mathbb{R}^{H \times N}$, resulting in $\mathcal{O}(NH)$ trainable parameters and a non-convex optimization problem.

Replacing random walks from the generator. In principle, we could use the synthetic transition matrix $P^* = \sigma_{\text{rows}}(W^*)$ defined with the solution W^* of Eq. (2.14) in place of the generator: we produce synthetic random walks of length T, with transition matrix P^*, and with the same distribution over the first node as in the training set, and then count their transitions in a score matrix. But since the score matrix is needed only up to proportionality for the edge-independent model, we can use the limit in Eq. (2.11) instead, which replaces sampling random walks with solving the eigenvector problem ${\pi^*}^\top P^* = {\pi^*}^\top$. That is, we skip sampling random walks and simply set $S = \operatorname{diag}(\pi^*) P^*$.

2.3.3 Our algorithm: Cross-Entropy Low-rank Logits (CELL)

In the previous section, we have shown how to (i) replace the LSTM and GAN architecture with a low-rank approximation of the logit transition matrix with respect to the cross-entropy loss, (ii) replace sampling random walks from the input graph with using its adjacency matrix directly, and (iii) replace sampling random walks from the generator with solving an eigenvector problem. The result of this analysis is our simplified algorithm Cross-Entropy Low-rank Logits (CELL), summarized in Algorithm 1. It takes the adjacency matrix A of a graph G as input and returns a symmetric matrix $A^\dagger$ of edge probabilities, from which new graphs G' can be sampled. For solving optimization problem (2.14), we factorize $W = W_{\text{down}} W_{\text{up}}$ with $W_{\text{down}} \in \mathbb{R}^{N \times H}$ and $W_{\text{up}} \in \mathbb{R}^{H \times N}$ to satisfy the rank constraint, and optimize with Adam (Kingma and Ba, 2014). Training continues until a stopping criterion is met, for which we pause at regular intervals and generate new graphs to evaluate the stopping criterion. In this work, we consider the criterion of reaching a predefined edge overlap of generated graphs and input graph, see Section 2.5.1.

2.4 Conceptual analysis

Our simple reformulation of NetGAN now opens the possibility to formally analyze the inductive bias associated with its components and allows for user-specific adaptations.

[1] Code available at https://github.com/hheidrich/CELL

Algorithm 1 Cross-Entropy Low-rank Logits (CELL)[1]

input adjacency matrix $A \in \{0,1\}^{N\times N}$, rank $H \ll N$
output matrix of edge probabilities $A^\dagger \in [0,1]^{N\times N}$

1: Solve optimization problem (2.14) for W^*
2: Compute transition matrix: $P^* \leftarrow \sigma_{\text{rows}}(W^*)$
3: Solve eigenvalue problem ${\pi^*}^\top P^* = {\pi^*}^\top$ for π^*
4: Compute score matrix: $S \leftarrow \text{diag}(\pi^*)P^*$
5: Convert score matrix S to edge-independent model $A^\dagger$:
$S^\dagger \leftarrow \max\{S, S^\top\}$; $A^\dagger \leftarrow S^\dagger/\text{sum}(S^\dagger)$

return $A^\dagger$

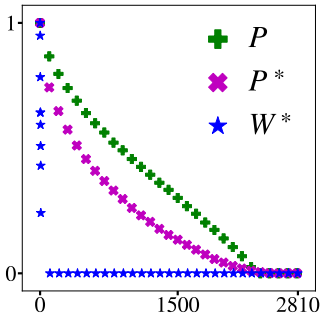


Figure 2.2: Portion of absolute eigenvalues (sorted and rescaled) for CORA-ML with CELL trained to 50% edge overlap for $H = 9$.

2.4.1 Inductive bias of NetGAN

Our analysis has shown that the graphs produced by NetGAN come from the class of graphs whose logit transition matrix has a low rank. Note that this does *not* imply that the transition matrix itself has low rank. Even if W^* is trained to have low rank, the corresponding synthetic transition matrix $P^* = \sigma_{\text{rows}}(W^*)$ can have full rank, as is visualized by the eigenvalues in Figure 2.2. Additionally, our experiments in Section 2.5.2 suggest that approximating the transition matrix with a low rank matrix and the Frobenius norm as loss function does not achieve good generalization performance. However, minimizing the cross-entropy loss for approximation instead yields generalization performance comparable to the one of NetGAN and CELL. Since the cross-entropy corresponds to the Kullback-Leibler (KL) divergence (see Section 2.4.2), this suggests that **using the KL divergence as distance measure for approximating transition matrices is the reason for the good generalization performance. On a high level, NetGAN generalizes a graph by choosing new graphs, whose transition matrix is similar in terms of KL-divergence, from a restricted set of graphs.** Whether this restriction is realized by a low-rank assumption on the logits or on the transition matrices itself is not essential, although the former is computationally more feasible.

2.4.2 Information-theoretic representation of objective function F

When considering distributions in this section, we let any matrix with positive entries refer to the uniquely determined distribution that is obtained after normalization. We can reformulate our objective F, defined in Eq. (2.13), in terms of information-theoretic quantities to determine its minimum irrespective of the rank constraint. To do so, we consider node transitions as a random variable $(\boldsymbol{v}, \boldsymbol{w})$ on $[N] \times [N]$. As derived for Eq. (2.13) in case of node transitions on the input graph, $(\boldsymbol{v}, \boldsymbol{w})$ is distributed according to the adjacency matrix A, wherefore the corresponding conditional distribution of $\boldsymbol{w}|\boldsymbol{v}$ is given by P. The synthetic transition matrix $\sigma_{\text{rows}}(W)$ represents another conditional distribution for $\boldsymbol{w}|\boldsymbol{v}$. From this perspective, we can reformulate F as

$$\begin{aligned} F(W) &= -\sum_{v,w=1}^{N} A_{v,w} \log \sigma_{\text{rows}}(W)_{v,w} \\ &\propto -\mathbb{E}_{(v,w)\sim A} \left[\log \sigma_{\text{rows}}(W)_{v,w}\right] \\ &= -\mathbb{E}_{(v,w)\sim A} \left[\log A_{v,w}\right] + \mathbb{E}_{(v,w)\sim A} \left[\log \left(\frac{A_{v,w}}{\sigma_{\text{rows}}(W)_{v,w}}\right)\right] \\ &= \mathrm{H}_A(\boldsymbol{w}|\boldsymbol{v}) + \mathrm{KL}\left(A(\boldsymbol{w}|\boldsymbol{v}) \;||\; \sigma_{\text{rows}}(W)(\boldsymbol{w}|\boldsymbol{v})\right) . \end{aligned}$$

The first term on the right-hand side is the conditional entropy of the true underlying node transition distribution A and does not depend on W. The second is the conditional relative entropy between the true node transition distribution A, whose conditional is given by P, and the learned conditional $\sigma_{\text{rows}}(W)$. This shows that F is minimized by any W satisfying $\sigma_{\text{rows}}(W) = P$, which makes the low-rank constraint necessary for generalization in the learning step.

2.4.3 Bias of the optimization objective and resulting hard examples

We optimize the objective function in Eq. (2.13) to learn the random walk distribution in form of its transition matrix. By inspecting the objective, we can understand how this is achieved: the synthetic transition matrix $\sigma_{\text{rows}}(W)$ is rewarded *directly* for putting mass on edges of the input graph ($A_{k,l} = 1$). But because the total mass is limited ($\sum_l \sigma_{\text{rows}}(W)_{k,l} = 1$), it is only penalized *indirectly* for wasting mass on non-edges ($A_{k,l} = 0$). In particular, there is no distinction between different non-edges. This hints towards **poor performance for graphs with strong restrictions on the set of edges we deem realistic**, because there is no notion of "bad" edges that could prevent their generation; a possible remedy to this problem is extending the objective function with such a notion.

We illustrate this effect with the example of ε-neighborhood graphs in Figure 2.3. Here, we want to avoid the generation of edges between nodes with large distance in the Euclidean space, which is not taken into account by NetGAN and CELL. However, a simple adaptation of our method, denoted as "Local CELL", can prevent long edges

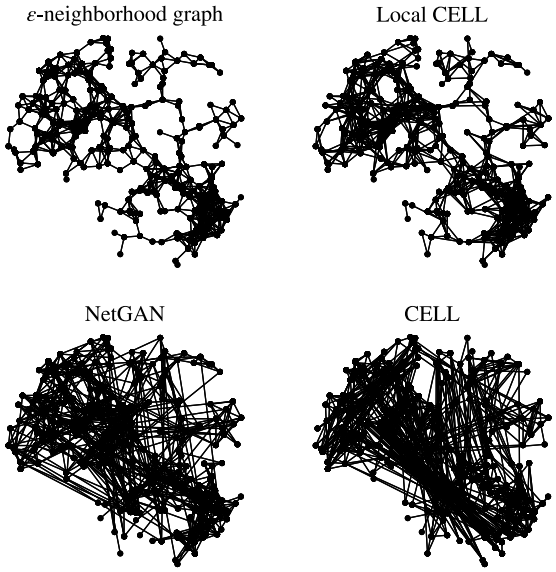


Figure 2.3: Comparison of an ε-neighborhood graph (top left) with graphs generated by Local CELL, a version of our method biased towards short edges, NetGAN, and our method CELL. Only Local CELL does not generate edges between distant points.

without loss of generalization performance. The corresponding experiment is provided in Section 2.6.4.

2.4.4 No bias in the reconstruction step

A natural question is whether our method might be able to generalize even without the rank constraint in the learning step. Or, phrased differently, whether the reconstruction step introduces a generalization bias. This is not the case. We derived in Section 2.4.2 that without any rank constraint, we exactly recover the input transition matrix as $P^* = P$. Because of $\operatorname{diag}(\pi)P \propto A$, this also holds for the score matrix

$$S = \operatorname{diag}(\pi^*)P^* \propto A\,. \tag{2.15}$$

Since A is already symmetric, the edge-independent model is given by $A^\dagger \propto A$. This model is equivalent to uniformly sampling edges from the input graph G, and sampling $e(G)$ edges from this model without replacement means sampling all of them. Hence it simply returns the input graph with zero variance. Therefore, **reconstructing the graph with an edge-independent model does not contribute to generalization.** Another interpretation of this observation is that random walks are sufficient to learn a graph in principle.

2.4.5 Influence of the random walk parameters

For NetGAN it is still unclear how the length T and the start distribution for the first node q_0 of the random walks influence the generated graphs. We derived in Section 2.3.2 that it does not exploit any complicated patterns in the random walk paths, but simply counts the transitions, which comes down to a weighting of the nodes. When translating NetGAN to our approach, we observe that **the random walk length controls how much influence the start distribution has on the node weights:** instead of taking the limit $T \to \infty$ in the derivation of Section 2.3.2, we could have completed the analysis with the node weights $\rho = \sum_{t=1}^{T} q_{t-1}/T$ to arrive at the parametrized objective function

$$F_\rho(W) = -\sum_{k,l=1}^{N} \frac{\rho_k}{d_k} A_{k,l} \log \sigma_{\text{rows}}(W)_{k,l} \,. \tag{2.16}$$

This allows for further interpretation and adaption:

Random walks of length one are sufficient. The random walk parameters T and q_0 are relevant for Eq. (2.16) only because they determine the node weights ρ. On the other hand, all possible node weights ω can be realized by choosing $q_0 = \omega$ and $T = 1$. This implies for NetGAN that **only using random walks of length one imposes no restriction**, if the distribution of the start node is considered as a hyperparameter instead.

An example that is now readily explained is the setting in Jalilifard et al. (2019). They observe empirically that using short random walks in NetGAN reduces the performance, and propose to counteract by choosing the start distribution as the density function described in Zhou et al. (2009). Within our framework, this is explained by the node weights: for short random walks, they are close to the uniform distribution (the start distribution of NetGAN), which overemphasizes nodes with low degree and results in bad performance. Choosing the start distribution closer to the stationary distribution instead has the same effect on the node weights as using long random walks.

Node weights ρ as a hyperparameter. Instead of indirectly setting the node weights ρ through the random walk parameters q_0 and T as is done in NetGAN, we can treat ρ as a hyperparameter directly to incorporate beliefs about the graph. Weighting nodes according to the stationary distribution assigns equal weight to all *edges* in Eq. (2.16). In general, increasing the weight of a node encourages generated graphs to include its adjacent edges. This enables us, for example, to "protect" a certain set of nodes in the sense of preserving their neighborhoods in the generated graphs by increasing their weight.

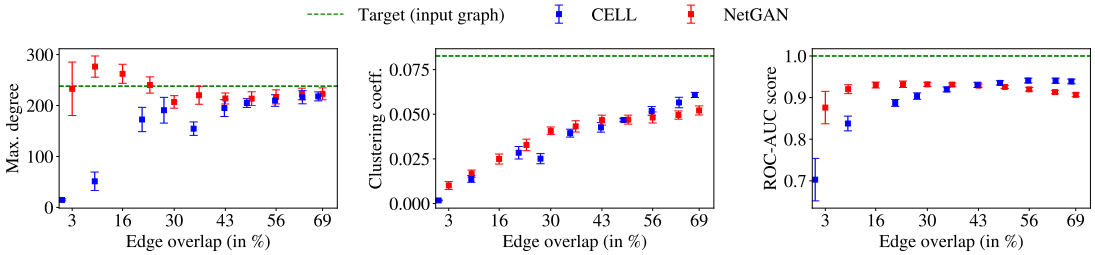


Figure 2.4: Mean and standard deviation for five trials on CORA-ML, plotted against edge overlap with the input graph. Aside from different initializations, NetGAN and CELL display similar behavior. Additional experiments are provided in Section 2.6.6.

2.5 Experiments

The purpose of this section is to (i) verify that CELL has performance comparable to NetGAN while being much faster, and (ii) demonstrate the importance of the cross-entropy loss and benefit of the logit-transformation by comparing with other low-rank approximation baselines.

2.5.1 Setup of the experiments

Data sets and preprocessing. We experiment on a variety of graph data sets: the citation networks CORA-ML (McCallum et al., 2000) and CITESEER (Sen et al., 2008), the political blogs network POLBLOGS (Adamic and Glance, 2005), the retweet network RT-GOP, and the web graph WEB-EDU (Gleich et al., 2004). All graphs except for CORA-ML are taken from Rossi and Ahmed (2015). For CORA-ML, we use the same preprocessed version as Bojchevski et al. (2018), an overview of the data sets is given in Table 2.3. We preprocess the graphs by removing loops, edge weights, and edge directions. We then restrict them to their largest connected component to ensure that they are connected. For evaluating the link prediction performance during and after training, we split each graph into training-, validation-, and test-set by taking out 10% of the edges for validaton and another 5% for testing, while ensuring that the remaining graph stays connected. The validation set is only used for the VAL-criterion, an alternative stopping criterion based on link prediction performance that is described in Section 2.6.3.

Baselines. We compare our model CELL to NetGAN (Bojchevski et al., 2018) and a number of non-parametric baselines: the configuration model, which simply rewires some randomly chosen edges (Molloy and Reed, 1995), and low-rank approximations of the adjacency matrix (LR-Adj), the random walk transition matrix (LR-Trans), the symmetric normalized Laplacian (LR-Lap), and the modularity matrix (LR-Mod), in a similar framework as described by Baldesi et al. (2018). To investigate the contribution

Table 2.2: Graph statistics and link prediction performance on CORA-ML for generated graphs from NetGAN, our method CELL, and baselines, averaged over five trials. Statistics that are matched by model design for the configuration model are indicated as $*$, and cases that are not applicable as $-$. CELL produces statistics comparable to NetGAN, but is orders of magnitudes faster. This experiment is repeated for all other data sets in Section 2.6.5.

Graph	Max. degree	Assort-ativity	Triangle count	Square count	Power law exp.	Cluster-ing coeff.	Charac. path len.	ROC-AUC score	Time (in s)
CORA-ML	238	-0.076	2,802	14,268	1.86	8.26e-2	5.63	1	–
Conf. model	$*$	-0.053	623	3111	$*$	1.96e-2	4.43	–	1
LR-Adj	121	-0.042	444	1,128	1.72	2.78e-2	5.17	0.561	32
LR-Trans	139	-0.058	558	1,617	1.77	2.94e-2	5.07	0.709	33
LR-Lap	167	-0.084	691	1942	1.79	2.79e-2	4.76	0.800	38
LR-Mod	122	-0.043	437	1,135	1.72	2.75e-2	5.17	0.557	48
LR-CE	193	-0.068	1,388	6,284	1.79	5.68e-2	5.37	0.950	73
NetGAN	219	-0.071	1,461	5,555	1.80	5.23e-2	5.13	0.950	7,478
CELL	204	-0.070	1,396	6,880	1.82	5.07e-2	5.26	0.938	21

Table 2.3: Data sets used. Nodes and edges refer to the largest connected component.

Name	Nodes	Edges
CORA-ML	2,810	7,981
Citeseer	2,110	3,668
PolBlogs	1,222	16,779
RT-GOP	4,687	5,529
Web-EDU	3,031	6,474

of the logit transformation for CELL, we additionally consider a low-rank approximation of the transition matrix with respect to the cross-entropy loss instead of using the Frobenius norm (LR-CE). The original paper by Bojchevski et al. (2018) also compared to a number of parametric baselines, which have the purpose of explicitly fitting some hand-selected graph parameters, but fail to reproduce others. For brevity we do not report the results of these parametric baselines.

Setup and evaluation metrics. To make the results comparable, we train CELL and NetGAN until the same stopping criterion of 52% edge overlap with the input graph is satisfied. This is done by pausing the training at regular intervals, generating a single graph, and calculating the ratio of shared edges to input edges. While NetGAN is trained on a GPU, only a CPU is required for training CELL.

Our first evaluation metric is a set of common graph statistics for input and generated graphs, whose purpose is to measure the extent to which the newly generated graphs reproduce network patterns of the input graph. Since memorizing the input graph trivially reproduces all of its graph statistics, we additionally evaluate the generalization properties in a link prediction task. To do so, we use the edges in the test set and an equal amount of randomly chosen non-edges from the original graph. After training, these are presented to the generative models, which try to classify them as existent or non-existent in the original graph on the basis of the score matrix (or, equivalently, the edge-independent model $A^{\dagger}$). This matrix is produced by all considered models except for the configuration model. A high value in the score matrix suggests the existence of the corresponding edge, while a low value suggests that the edge did not exist in the original graph. The performance is measured by the ROC-AUC score (Area Under Curve for Receiver Operating Characteristic curve), applied to the score matrix evaluated at the edges in question.

2.5.2 Evaluation

CELL vs. NetGAN. The results for graphs generated on CORA-ML are presented in Table 2.2. Compared to the other baselines, **CELL generates graphs with statistics close to those of NetGAN** and has similar link prediction performance; some of their small differences might be attributed to the noise of the LSTM used by NetGAN, and to the different optimization procedures. The latter can be observed in Figure 2.4, which shows the evolution of generated graph statistics during training: NetGAN starts off with a different initialization, but as training continues, the generated graph statistics get close to the target well before memorizing the input graph. Further confirmation of this behavior is given in Sections 2.6.6 and 2.6.7. However, the most striking difference is the training time, for which our method is **orders of magnitudes faster**, see Table 2.1.

CELL vs. baselines. Almost all **baselines fail to reproduce most of the graph statistics**, while CELL is reasonably close to all of them. Only LR-CE, the version of our method without the logit space, has performance very similar to CELL. This hints towards the **importance of the cross-entropy loss** rather than the logit space for successfully generalizing a graph. However, using the logit space still has the advantage of requiring only a small rank ($H = 9$ for CELL as compared to $H = 950$ for LR-CE), which results in less trainable parameters and shorter training time.

2.6 Experimental details and additional experiments

2.6.1 Graph statistics

Definition of various graph statistics used in this work. Part of the table is extracted from Bojchevski et al. (2018).

Table 2.4: Graph statistics for a graph $G = (V, E)$ with $N = |V|$ nodes and $m = |E|$ edges.

Graph statistic	Computation	Description
Assortativity	$\frac{\text{cov}(X,Y)}{\sigma_X \sigma_Y}$	Pearson correlation of degrees of connected nodes, where the (x_i, y_i) pairs are the degrees of connected nodes.
Power law exponent	$1 + n \left(\sum_{v \in V} \log \frac{d(v)}{d_{\min}} \right)^{-1}$	Exponent of the power law distribution, where d_{MIN} denotes the minimum degree in a network.
Relative edge distribution entropy	$-\frac{1}{\log N} \sum_{v \in V} \frac{d(v)}{2m} \log \frac{d(v)}{2m}$	Normalized entropy of the degree distribution, 1 means uniform, 0 means a single node is connected to all others.
Gini coefficient	$\frac{2 \sum_{i=1}^{N} i \hat{d}_i}{N \sum_{i=1}^{N} \hat{d}_i} - \frac{N+1}{N}$	Common measure for inequality in a distribution, where $\hat{d}$ is the sorted list of degrees in the graph.
Characteristic path length	$\frac{1}{N(N-1)} \sum_{u \neq v} d(u, v)$	Average shortest path length, where $d(u, v)$ is the shortest path length between nodes u and v.
Spectral gap	$\lambda_1(L)$	Smallest non-zero eigenvalue λ_1 of the graph Laplacian $L = D - A$.
Motif count	–	Number of copies of H contained in G as a subgraph. Considered motifs are wedges, triangles, and squares.

2.6.2 Baselines

- **Configuration model.** We randomly sample a fraction of the edges in the input graph (fraction stated in brackets), and then rewire the remaining edges by severing them and randomly matching the stubs. This yields a graph with the same degree distribution as in the input graph. Because the resulting graph is not simple in general, we then remove all loops and multiple edges (with high probability, there are only few of them).

- **Low-rank approximations with respect to Frobenius norm.** A class of graph generative models similar in spirit to our method is the Spectral Graph Forge framework, which is based on performing low-rank approximations of matrices derived from the input adjacency matrix A. The pipeline consists of the following steps:

  1. Transform A into any derived matrix $M = M(A)$.
  2. Perform a low-rank approximation of M to obtain $\tilde{M}$.
  3. Back-transform $\tilde{M}$ to $\tilde{A}$ by applying the inverse of the transformation.
  4. Obtain edge-independent model $A^\dagger$ by making $\tilde{A}$ symmetric and then normalizing it.
  5. Sample the new adjacency matrix $A' \sim A^\dagger$.

  We apply this framework to the adjacency matrix A (no transformation), the random walk transition matrix $P = D^{-1}A$, the symmetric normalized Laplacian $L^{\text{sym}} = I - D^{-\frac{1}{2}}A^{-\frac{1}{2}}$, and the modularity matrix $B = A - dd^\top/(2e(G))$. The rank of the approximation is chosen so that the desired edge overlap with the input graph is reached: on CORA-ML, we use rank 1600 for $M \in \{A, P, B\}$ and rank 2520 for $M = L^{\text{sym}}$. For the transition matrix, we choose the same back-transformation $\tilde{A} = \text{diag}(\tilde{\pi})\tilde{P}$ as in our method instead of $\tilde{A} = D\tilde{P}$, which uses the degree matrix of the input graph. Given $\tilde{A}$, we proceed like NetGAN and our method with $S = \tilde{A}$. For link prediction, we also use the score matrix.

- **Low-rank approximation with respect to cross-entropy loss.** This baseline is a version of our method CELL, but without the logit space. That is, we solve the optimization problem

$$\begin{aligned} \min_{\tilde{P} \in \mathbb{R}^{N \times N}} & -\sum_{k,l=1}^{N} A_{k,l} \log \tilde{P}_{k,l} \,, \\ \text{s.t.} \quad & \text{rank}(\tilde{P}) \leq H \quad \text{and} \quad \tilde{P} \in \mathcal{P} \,, \end{aligned} \tag{2.17}$$

  where $\mathcal{P}$ is the set of stochastic matrices on $\mathbb{R}^{N \times N}$. Similar to how we proceed in CELL, we enforce this constraint with the parametrization

$$\tilde{P} = D(e^C e^D)^{-1} e^C e^D \,, \tag{2.18}$$

where $C \in \mathbb{R}^{N\times H}, D \in \mathbb{R}^{H\times N}$, the exponential function is taken element-wise, and $D(e^C e^D)$ denotes the diagonal matrix of row sums for $e^C e^D$. We then optimize the objective in Eq. (2.17) over C and D with Adam.

2.6.3 Stopping criteria

In addition to the rank constraint, CELL and NetGAN both use an early stopping criterion for learning the random walk distribution.

EO-criterion. The EO-criterion is the stopping criterion used in this work and by NetGAN, and generates graphs with a predefined edge overlap with the input graph (e. g. 50%). To employ it, training is stopped during regular intervals, the edge-independent model is constructed, and a single graph G' is generated. If the fraction of edges $e(G \cap G')/e(G)$ is smaller than the predefined threshold, training is continued, otherwise it is stopped.

VAL-criterion. The VAL-criterion is a stopping criterion proposed by NetGAN and represents an alternative to the edge overlap (EO) criterion used in this work. It is employed by evaluating the link prediction performance on the validation set during training of the optimization problem, and then stopping the training as soon as the link prediction performance does not improve for a predefined amount of training iterations. For evaluation after training, the test set is used instead of the validation set.

2.6.4 Example of a bias of NetGAN and CELL: ε-neighborhood graphs

This section demonstrates the concepts discussed in Section 2.4.3 on ε-neighborhood graphs. These graphs arise by choosing the nodes as points in a metric space, and connecting those pairs of points by an unweighted, undirected edge whose distance is smaller than a constant $\varepsilon > 0$ (see Figure 2.3 for an illustration). Given an ε-graph as input, there is one major property that a graph generative model should keep intact: edges should occur between points with a small distance in the underlying space, but not for points with a large distance. Figure 2.3 shows that NetGAN and CELL do not comply with this desired tendency: both algorithms generate long edges. We can easily counteract this mismatch in inductive bias for CELL by extending our loss function with an additional term that penalizes long edges. Because the underlying distances of the metric space are not directly represented in the ε-neighborhood graph anymore, we use its shortest path distances $\mathcal{D} \in \mathbb{R}^{N\times N}$ as a proxy (which is sensible as one can prove that the shortest path distances in ε-graphs converge to the underlying metric distances (Orlitsky, 2005; Tenenbaum et al., 2000)). We refer to the resulting method

as "Local CELL", whose loss function is given by

$$F(W) = -\sum_{k,l=1}^{N} A_{k,l} \log \sigma_{\text{rows}}(W)_{k,l} - \sum_{k,l=1}^{N} A_{k,l} \left[\mathcal{D}_{k,l} \leq k\right] \log \sigma_{\text{rows}}(W)_{k,l}\,, \tag{2.19}$$

where $[\mathcal{A}] = 1$, if statement $\mathcal{A}$ is true, and 0 otherwise. A comparison of NetGAN, CELL, and Local CELL is given in Table 2.5, see also Figure 2.3 for an illustration. As expected, NetGAN and CELL generate graphs with long edges, resulting in a large average edge length and small characteristic path length. Local CELL on the other hand is significantly closer to the input graph in this regard without a loss in performance for other statistics. It even improves on some other statistics, because the objective function is more appropriate for this type of graph.

Table 2.5: Statistics of ε-neighborhood graph and generated graphs from three generative models, averaged over five trials. For the generated graphs, "avg. edge len." is computed only from generated edges that are not present in the input graph.

Graph	Charac. path len.	Avg. edge len.	Spectral gap	Assort-ativity	Power law exp.	Triangle count	Wedge count
ε-neighborhood	8.97	0.10	2.04e-3	0.75	1.51	2,319	11,557
NetGAN	3.54	0.41	4.49e-2	0.09	1.50	895	10,638
CELL	4.01	0.50	3.57e-2	0.38	1.50	1,144	11,030
Local CELL	5.92	0.21	5.13e-3	0.47	1.51	1,088	11,377

2.6.5 Additional baseline experiments

Graph statistics and link prediction performance on all data sets described in Section 2.5 for generated graphs from NetGAN, our method CELL, and baselines, averaged over five trials. Statistics that are matched by model design for the configuration model are indicated as $*$, and cases that are not applicable as $-$. For a visualization and interpretation of the results, see Section 2.6.7.

Table 2.6: CORA-ML (2,810 nodes, 7,981 edges).

Graph	Max. degree	Assort-ativity	Triangle count	Square count	Power law exp.	Cluster-ing coeff.	Charac. path len.	ROC-AUC score	Time (in s)
CORA-ML	238	-0.076	2,802	14,268	1.86	8.26e-2	5.63	1	–
Conf. model	∗	-0.053	623	3111	∗	1.96e-2	4.43	–	1
LR-Adj	121	-0.042	444	1,128	1.72	2.78e-2	5.17	0.561	32
LR-Trans	139	-0.058	558	1,617	1.77	2.94e-2	5.07	0.709	33
LR-Lap	167	-0.084	691	1942	1.79	2.79e-2	4.76	0.800	38
LR-Mod	122	-0.043	437	1,135	1.72	2.75e-2	5.17	0.557	48
LR-CE	193	-0.068	1,388	6,284	1.79	5.68e-2	5.37	0.950	73
NetGAN	219	-0.071	1,461	5,555	1.80	5.23e-2	5.13	0.950	7,478
CELL	204	-0.070	1,396	6,880	1.82	5.07e-2	5.26	0.938	21

Table 2.7: Citeseer (2,110 nodes, 3,668 edges).

Graph	Max. degree	Assort-ativity	Triangle count	Square count	Power law exp.	Cluster-ing coeff.	Charac. path len.	ROC-AUC score	Time (in s)
Citeseer	72	-0.015	483	1,866	2.24	8.70e-2	10.68	1	–
Conf. model	∗	-0.014	108	282	∗	1.95e-2	6.33	–	1
LR-Adj	34	4.75e-2	89	188	2.09	2.62e-2	8.17	0.608	12
LR-Trans	36	-0.022	119	364	2.15	3.20e-2	8.58	0.825	8
LR-Lap	48	0.019	108	161	2.18	2.45e-2	7.82	0.362	12
LR-Mod	32	5.33e-4	87	162	2.09	2.67e-2	8.17	0.603	108
LR-CE	47	-0.076	138	549	2.13	3.50e-2	8.57	0.903	19
NetGAN	52	-0.074	361	478	2.15	8.50e-2	9.03	0.951	4,654
CELL	44	-0.093	106	318	2.17	2.54e-2	7.36	0.858	10

Table 2.8: PolBlogs (1,222 nodes, 16,779 edges).

Graph	Max. degree	Assort-ativity	Triangle count	Square count	Power law exp.	Cluster-ing coeff.	Charac. path len.	ROC-AUC score	Time (in s)
PolBlogs	298	-0.222	60,873	2,631,731	1.44	0.189	2.82	1	–
Conf. model	∗	-0.140	31,364	1,263,826	∗	0.118	2.72	–	1
LR-Adj	171	-0.022	15,497	430,846	1.36	0.082	2.66	0.63	1
LR-Trans	200	-0.114	27,428	918,543	1.40	0.114	2.73	0.861	1
LR-Lap	234	-0.214	19,593	511,781	1.36	0.086	2.55	0.745	2
LR-Mod	170	-0.028	15,528	433,669	1.36	0.082	2.66	0.624	16
LR-CE	248	-0.226	34,942	1,303,305	1.40	0.126	2.66	0.943	17
NetGAN	261	-0.244	37,849	1,438,174	1.41	0.132	2.70	0.950	55,276
CELL	268	-0.243	49,366	2,043,407	1.43	0.160	2.78	0.949	15

Table 2.9: RT-GOP (4,687 nodes, 5,529 edges).

Graph	Max. degree	Assort-ativity	Triangle count	Square count	Power law exp.	Cluster-ing coeff.	Charac. path len.	ROC-AUC score	Time (in s)
RT-GOP	270	-0.135	0	2	4.29	0	14.01	1	–
Conf. model	∗	-0.092	56	241	∗	1.89e-3	5.68	–	1
LR-Adj	239	-0.117	0	87	3.74	0	12.05	0.559	7
LR-Trans	328	-0.111	0	139	4.53	0	6.18	0.676	19
LR-Lap	162	-0.070	5	1	3.09	5.15e-4	14.61	0.466	164
LR-Mod	192	-0.101	0	34	3.41	0	21.10	0.550	84
LR-CE	233	-0.122	0	29	3.74	0	20.08	0.874	129
NetGAN	221	-0.112	14	15	3.64	6.74e-4	16.33	0.738	14,800
CELL	253	-0.142	0	6	4.11	0	16.90	0.704	23

Table 2.10: Web-EDU (3,031 nodes, 6,547 edges).

Graph	Max. degree	Assort-ativity	Triangle count	Square count	Power law exp.	Cluster-ing coeff.	Charac. path len.	ROC-AUC score	Time (in s)
Web-EDU	99	-0.183	4491	35,423	2.11	0.167	4.56	1	–
Conf. model	∗	-0.109	873	4,913	∗	0.032	4.59	–	1
LR-Adj	58	-0.114	1,096	3,932	1.97	0.081	6.66	0.579	18
LR-Trans	166	-0.034	2,692	20,424	2.13	0.120	5.13	0.862	12
LR-Lap	103	-0.098	514	1,637	2.01	0.028	5.25	0.360	30
LR-Mod	58	-0.121	989	3,292	1.97	0.075	6.44	0.595	97
LR-CE	74	-0.136	1,194	4,330	1.99	0.080	6.41	0.994	72
NetGAN	92	-0.174	1,244	3,022	2.02	0.064	5.51	0.992	11,000
CELL	63	-0.234	1,176	4,710	2.03	0.069	6.67	0.977	16

2.6.6 Evolution of graph statistics during training

To compare the generated graphs of CELL and NetGAN for different edge overlaps, we fix all hyperparameters and stop training at regular intervals to compute the graph statistics of the generated graphs. Since we fix the ranks for the low-rank constraint, the generated graphs will not converge to 100% edge overlap. But this area of high edge overlap is of little interest anyway, because the shared edges alone force the generated graphs to reproduce many graph statistics. Note that in order to reduce computational complexity, NetGAN uses less random walks to compute statistics during training (for example to evaluate the stopping criteria): instead of sampling many random walks from the generator, it keeps track of the random walks generated in the last 1,000 iterations during training to build the score matrix. For our method CELL there is no such distinction, we complete our pipeline as described in Section 2.5.

Aside from few exceptions, for example the triangle count and the related clustering coefficient on CITESEER, we observe the same behavior as described in Section 2.5.2: after a short initialization phase, CELL and NetGAN display comparable behavior.

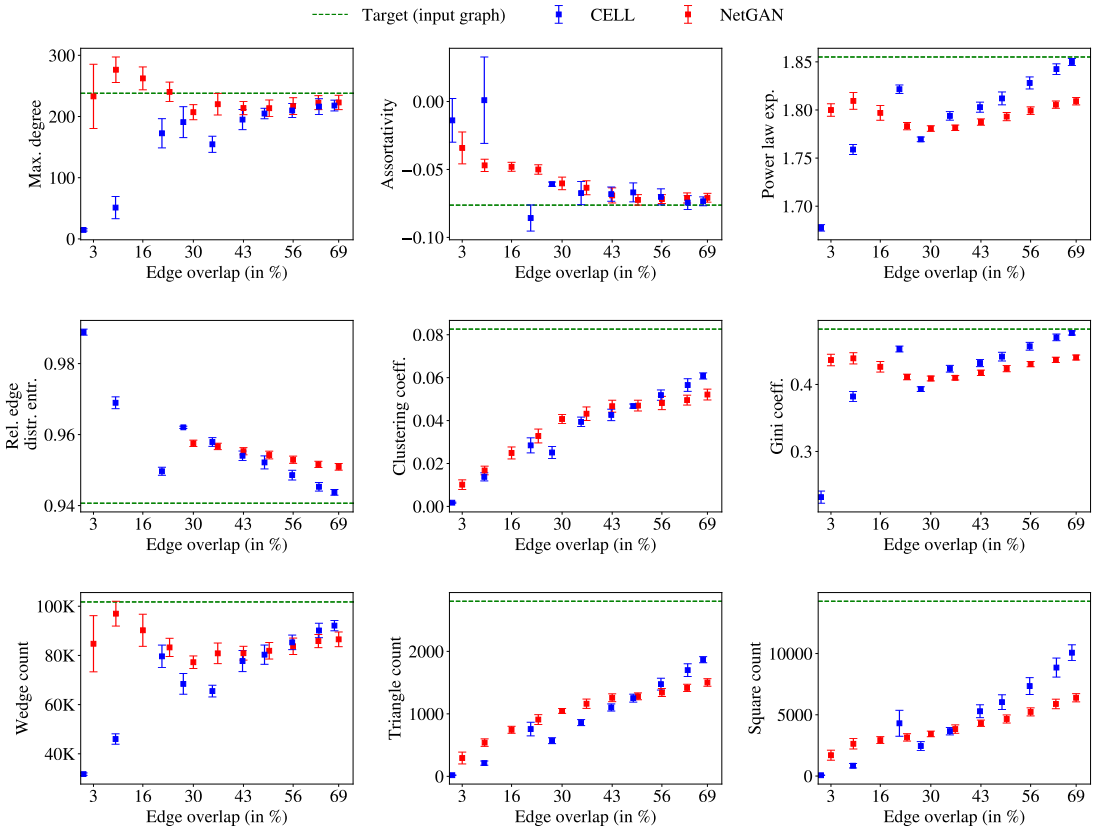


Figure 2.5: Graph statistics during training for NetGAN and CELL on CORA-ML, plotted against edge overlap.

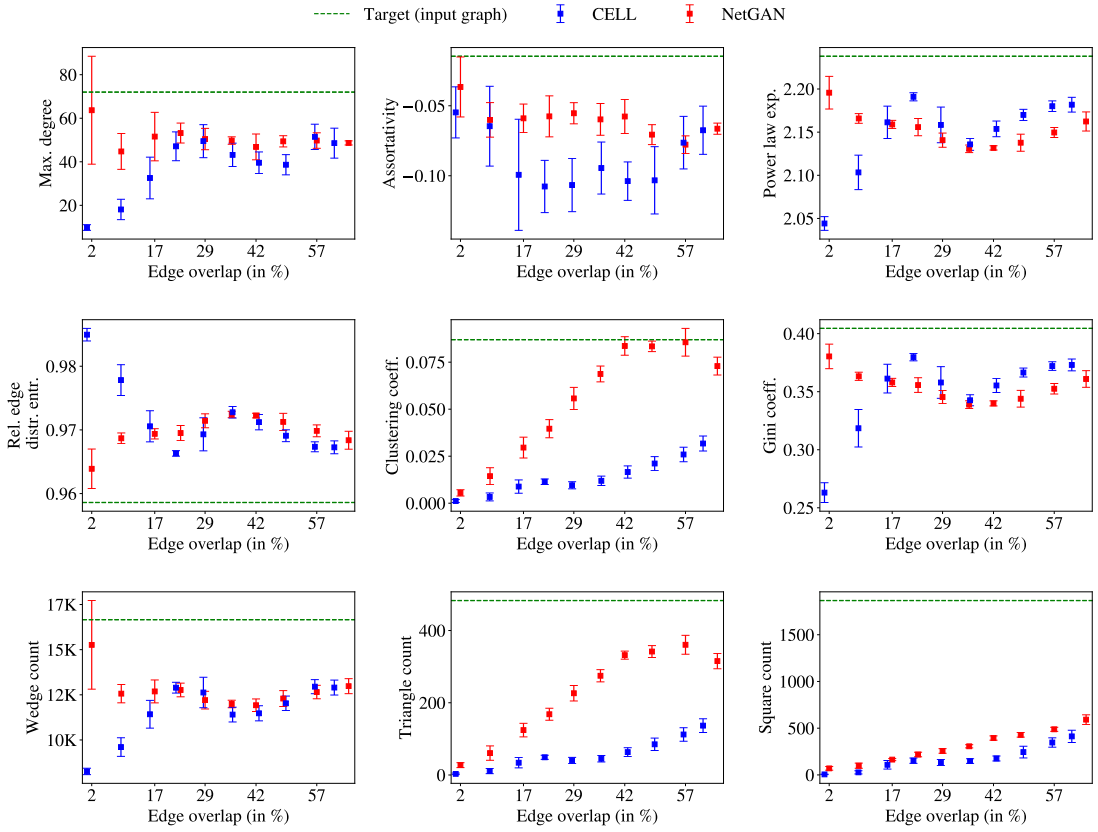


Figure 2.6: Graph statistics during training for NetGAN and CELL on CITESEER, plotted against edge overlap.

2.6.7 Comparison of relative errors

For a graph statistic s on the input graph, let $s_{\mathrm{M}} = (s_{\mathrm{M}}^{(1)}, \ldots, s_{\mathrm{M}}^{(K)})$ denote the estimates of model M for s in K trials. The average relative error is then defined by $s_{\mathrm{rel}}(\mathrm{M}) = 1/K \sum_{k=1}^{K} |s - s_{\mathrm{M}}^{(k)}|/|s|$. In Figure 2.7, we depict the relative errors for NetGAN, CELL, and baselines on a variety of data sets and graph statistics. Small relative errors indicate good performance in the sense that the generated graphs are close to the input graph. Over all data sets, a general trend can be observed: on most instances, the three models NetGAN, CELL, and LR-CE behave similarly and better as compared to the other baselines. Occasional deviations of this behavior might be attributed to the different optimization procedures and the early stopping. For some networks, for example CITESEER, NetGAN seems to outperform CELL, but for others like POLBLOGS, CELL performs better; this reflects that their different optimization procedures might or might not contribute to the goal of learning the network at hand.

2.6.8 Hyperparameters

For all our considered models except the configuration model and NetGAN, we choose the rank parameter H such that the generated graphs achieve the predefined edge overlap with the input graph. For example on CORA-ML with 2,810 nodes, we choose $H = 1600$ for LR-Adj, LR-Trans, and LR-Mod, $H = 2520$ for LR-Lap, $H = 950$ for LR-CE, and only $H = 9$ for our method CELL. In general, a higher rank increases the ability of the model to generate graphs with a high edge overlap. For NetGAN, we only consider unbiased random walks ($p = q = 1$) with batch size 128 and length 16. The dimensions H_g and H_d for the low-rank projection for generator and discriminator are both 128. Both generator and discriminator have a single hidden layer with 40 hidden units for the generator and 30 hidden units for the discriminator. The temperature τ is annealed from $\tau = 5$ to $\tau = 0.5$ with a multiplicative decay of $1 - 10^{-5}$ every step.

We optimize the methods LR-CE, NetGAN and CELL using Adam. For LR-CE and CELL, we use a learning rate of 0.1 and weight decay of 10^{-7}, and for NetGAN the learning rate is 0.0003 with L_2-regularization of 10^{-7} for the generator and $5 \cdot 10^{-5}$ for the discriminator. The Wasserstein gradient penalty is set to 10.

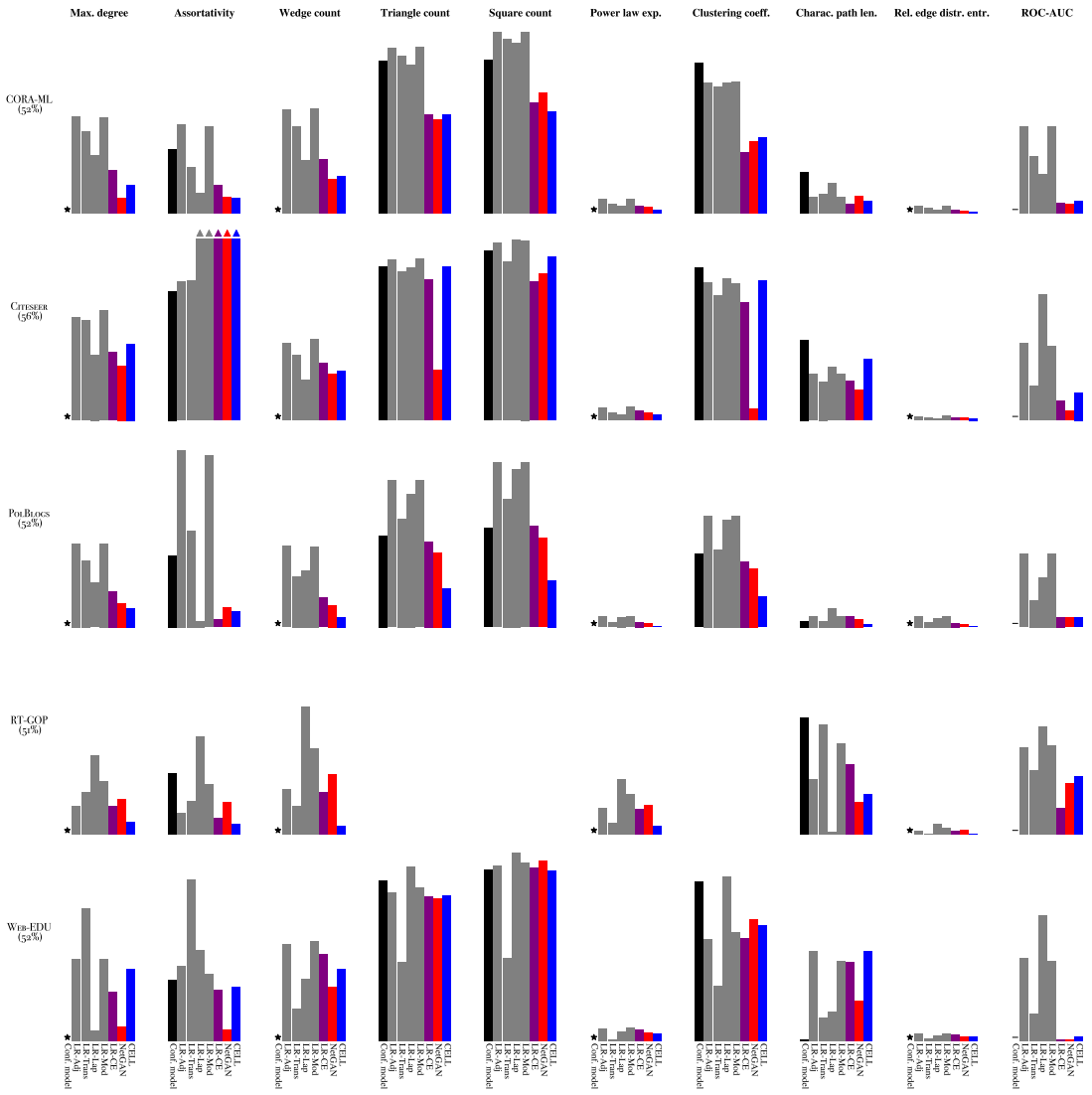


Figure 2.7: Relative errors of NetGAN, CELL and baselines, trained until the EO-stopping criterion given in brackets, and averaged over five trials. Rows represent the input graphs, columns represent the graph statistics. The y-axis ranges from 0 to 1 in every cell; values that exceed 1 are capped and indicated with an arrow. For the Conf. model, statistics that are matched exactly (0 relative error) are indicated by $\star$, and the non-existent ROC-AUC score is indicated by a $-$. The three statistics triangle count, square count and clustering coefficient for the extremely sparse network RT-GOP are omitted, because their value is zero and the relative errors are not defined (triangle count, clustering coefficient), or it is too small to be produce a meaningful relative error (square count is 2). For the actual graph statistics of generated and input graphs, see Section 2.6.5.

2.7 Discussion and future work

We derived a condensed version of NetGAN by identifying its essential steps and performing them directly. We verified experimentally that it retains the generalization performance of NetGAN, but is much faster. Additionally, our simple formulation of the algorithm makes it more accessible for analysis and application-specific extensions.

Analysis. In essence, we revealed the initial random-walk-based approach to be a low-rank approximation of the random walk transition matrix in the logit space. More naive low-rank approximations of matrices related to the input graph do not achieve competitive performance when approximating with respect to the Frobenius norm, but do so for the cross-entropy loss — a curious fact that we plan to investigate in future work. Based on our new, simplified methods we could analyze the inductive biases of the different components and the role of the parameters of NetGAN. For example, we discover that length and choice of start node of the random walks amount to a weighting of the nodes, which controls their importance in the graph generation process. Based on our better understanding of the bias, we can construct examples which both NetGAN and our algorithm cannot treat in a satisfactory manner.

Extensions. We demonstrated that our method is easily extendable by manipulating the loss function. An additional loss term can prevent the generation of edges we deem undesired, and node weights can emphasize user-specified nodes. Because learning step and reconstruction step are independent, each of them could be replaced by a different procedure. For example, instead of sampling from an edge-independent model, a more general method would sample independent paths to further emphasize locality.

Conclusion. Beyond the particular case of NetGAN, our work is part of a more high-level agenda. Machine learning is used in diverse applications, often not by machine learning experts, and the outcome of algorithms might have considerable impact in science and society. In such a context it is particularly important that our community actively attempts to understand the inherent inductive biases, strengths, and also the weaknesses of algorithms. Finding examples where an algorithm works is important — but maybe even more important is to understand under which circumstances the algorithm produces misleading results. For graph generative models, this might concern medical studies on brain graphs or geoscience studies on climate graphs. We should work hard to make our algorithms as transparent and interpretable as possible. This work is a small step in that direction.

Chapter 3

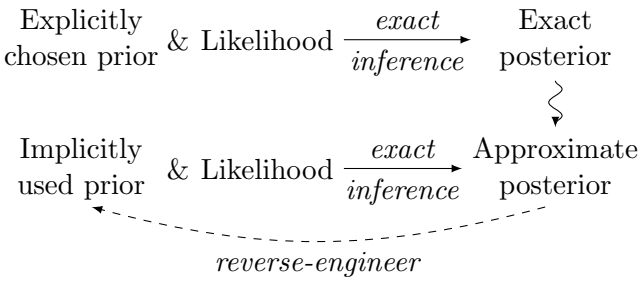
Discovering Inductive Bias with Gibbs Priors: A Diagnostic Tool for Approximate Bayesian Inference

Bayesian inference is based on the posterior distribution $p(\theta|y)$ over latent variables θ given an observation y. Bayes' theorem gives an explicit formula for computing the posterior, but is often infeasible in practice because the latent space is too large to work with, the appearing integrals are intractable, or the likelihood function cannot be evaluated. In these cases, practitioners revert to approximating the posterior instead. This approach comprises a cornucopia of methods, which can be divided into two groups. The first group consists of deterministic approximation methods that compute a feasible approximating distribution [1] $q(\theta|y)$ to the exact posterior $p(\theta|y)$ and includes methods such as variational inference (Blei et al., 2017; Hinton and van Camp, 1993; Hoffman et al., 2013; Jordan et al., 1999; Kucukelbir et al., 2017; Ranganath et al., 2014), Laplace approximations (Daxberger et al., 2021; MacKay, 1992; Rue et al., 2009, 2017; Spiegelhalter and Lauritzen, 1990), and expectation propagation (Minka, 2001). The second group consists of stochastic sampling methods that generate samples from (an approximation to) the posterior and includes methods such as Markov chain Monte Carlo (Bardenet et al., 2017; Casella and George, 1992; Hoffman and Gelman, 2014) and approximate Bayesian computation (Beaumont, 2019; Diggle and Gratton, 1984; Sisson et al., 2018). For a general introduction to approximate methods in Bayesian inference see Bishop (2006). While approximate methods make Bayesian inference feasible, they come at the cost of a distortion in the posterior. The resulting approximate inference can deviate significantly from exact Bayesian inference. This calls for diagnostic tools

[1] While standard notation for the approximation is $q(\theta)$, it will be useful in the context of this work to think of it as a conditional distribution.

to assess whether the result can still be trusted. Most existing diagnostics suffer from one or more of the following weaknesses: they are specific to a particular setting, they require evaluating the density of the approximation, which is unavailable for sampling-based methods, or they are restricted to the marginal distributions of a multivariate posterior. An overview of diagnostic tools is given in Section 3.1.

Existing diagnostics describe the difference to exact Bayesian inference by assessing the mismatch between approximation and true posterior. In contrast, we investigate a new perspective for diagnostic tools: we describe the approximate inference directly by attributing this mismatch to a change in the inductive bias. In a fully Bayesian setting, the inductive bias is specified explicitly by the model, which consists of the prior (a priori preference for solutions) and the likelihood (data generating process). Approximating the posterior can introduce additional bias that is not reflected in the model specification. We fix the likelihood and only allow the prior to change. The main idea of this work is to treat the approximation as an exact posterior to the same likelihood and reverse-engineer the corresponding implicitly used prior:



This implicit prior describes the inductive bias of the approximation in terms of an a priori preference for solutions. Figure 3.1 shows an example of inference based on posterior approximations that are biased towards solutions of small norm. This corresponds to effectively using a different prior with more mass on solutions of small norm than the explicitly chosen prior.

Let $(f(\cdot|\theta))_\theta$ be the likelihood and $(q(\cdot|y))_y$ the approximations to the posteriors $(p(\cdot|y))_y$. It is reasonable to define the implicit prior to the approximations by fixing an observation y and simply reverting Bayes' theorem[2] $\pi_y(\theta) \propto_\theta q(\theta|y)/f(y|\theta)$. Unfortunately, π_y generally depends on the observation y. This means that the approximations to different observations can correspond to different implicit priors, in which case no single distribution $\tilde{\pi}$ satisfies $q(\theta|y) \propto_\theta \tilde{\pi}(\theta)f(y|\theta)$. We only have the following weaker interpretation:

> Inference based on the approximate posteriors $(q(\cdot|y))_y$ is exact Bayesian inference with the same likelihood $(f(\cdot|\theta))_\theta$, but the prior is chosen from the family $(\pi_y)_y$ depending on the observation y.

[2]Note that π_y can be improper, that is, not integrable.

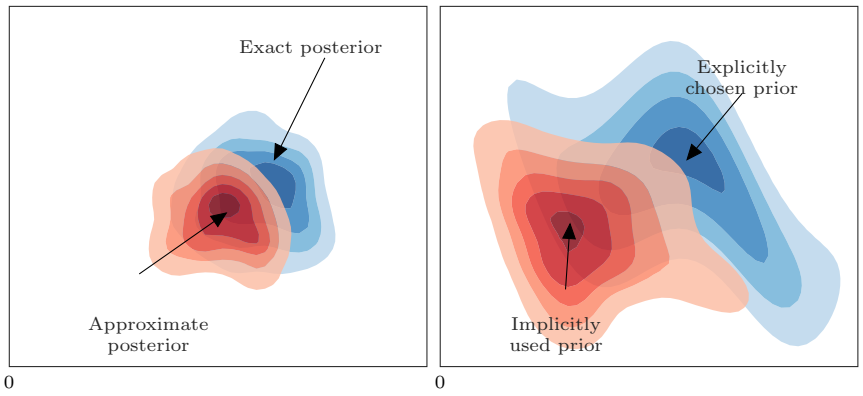


Figure 3.1: **Left:** a posterior approximation is biased towards solutions of small norm. **Right:** the approximation corresponds to the exact posterior under another implicitly defined prior, which is itself biased towards solutions of small norm.

Of course, the prior should not depend on the observation if we want to interpret it as the a priori preference for solutions. To understand the inductive bias of the approximations, we need an observation-independent distribution to compromise between this family of priors. We look at this problem through the lens of incompatible conditional distributions (Arnold and Press, 1989). This yields a natural solution based on pseudo-Gibbs sampling, which we call the *Gibbs prior*. An introduction to incompatible conditionals and pseudo-Gibbs sampling is given in Section 3.1.1.

Observation-(in)dependent diagnostics A diagnostic can either treat an approximation under a *fixed observation* $q(\cdot|y)$ or assess the average behavior of the approximation method *across observations* $(q(\cdot|y))_y$. These different tasks can show opposing behavior because an approximation can be good on specific instances but bad in general, or vice versa. Diagnosing a single approximation helps to understand and improve the inference under the fixed observation, but does not inform about how the approximation method performs in other cases. In our setting, this task is performed by the distributions π_y. However, we are interested in the systematic bias of the whole approximation method, which is why we search for an observation-independent compromise between the π_y. This kind of diagnostic does not guarantee the same behavior on any fixed observation, but helps to understand the method itself.

Contributions

- We investigate the novel approach of diagnosing approximate Bayesian inference methods in terms of their inductive bias. We show that this requires a compromise and reframe it as a problem of incompatible conditional distributions.

- We propose the Gibbs prior as a natural solution to the above problem (Section 3.2) and as a diagnostic tool. It is based on pseudo-Gibbs sampling, which is widely applicable and easy to implement.

- We demonstrate how the Gibbs prior can be used to discover the inductive bias of approximate Bayesian inference methods in a Gaussian toy example (Section 3.3) and two intractable Bayesian models (Section 3.7).

3.1 Related work

We divide the literature for diagnostics into two broad categories, depending on how they assess an approximation mismatch. Diagnostics in the first category compute a divergence between (quantities related to) the posterior and its approximation. Gorham and Mackey (2015, 2017) compute Stein discrepancies between the posterior and its approximation. Cusumano-Towner and Mansinghka (2017) compute the symmetric KL divergence between the approximation and another baseline approximation. Domke (2021) computes the symmetric KL divergence between the true joint distribution $p(y)p(\theta|y)$ and its approximation $p(y)q(\theta|y)$. Huggins et al. (2020) use the Wasserstein distance to bound the error of posterior point estimates. Diagnostics in the second category consider derived quantities that are known exactly under the true posterior and test whether they deviate under the approximations. Xing et al. (2020) compare a distortion map for posterior cumulative distribution functions to the identity. Yu et al. (2021) compare average posterior means and covariances to prior means and covariances. Cook et al. (2006) initiate another line of work based on the distribution of posterior quantiles, which is tested for uniformity; a corrected implemenation is presented by Talts et al. (2018). Yao et al. (2018) relax the uniformity test of Cook et al. (2006) and only test for symmetry. They also present another diagnostic based on Pareto-smoothed importance sampling. Prangle et al. (2014) test for uniformity of p-values related to the coverage property; this method is extended by Rodrigues et al. (2018). Our diagnostic also falls into this category where the Gibbs prior is compared to the original prior. The above diagnostics can also be divided by whether they analyze approximation methods for fixed or general observations. Our goal of diagnosing average approximation behavior is shared by Cook et al. (2006); Domke (2021); Talts et al. (2018); Yao et al. (2018); Yu et al. (2021).

Our diagnostic is based on sampling alternatingly from likelihood and approximation. The same technique was originally used by Geweke (2004) under the name *successive-conditional simulator* with the same goal of diagnosing approximations. Although both diagnostics are based on the same technique, they apply it differently: Geweke (2004) uses the simulator without reference to compatibility for generating tuples $(\tilde{\theta}_i, \tilde{y}_i)_i$, which are tested against samples from the Bayesian model $(\theta_i, y_i)_i$ to assess whether the approximations are exact; we focus on the marginal values $(\tilde{\theta}_i)_i$ that describe the implicitly used prior to assess the inductive bias. Our diagnostic is also similar in spirit

to Joshi and Ruggeri (2020) who link distortions in the likelihood to distortions in the prior.

3.1.1 Incompatible conditionals and pseudo-Gibbs sampling

When treating approximations as exact posteriors we inevitably face the problem of compatibility, which we shortly introduce in this paragraph. A bivariate model can be specified explicitly through its joint distribution $p(\theta, y)$, for example as in Bayesian models with a marginal $p(\theta)$ (prior) and a conditional distribution $p(y|\theta)$ (likelihood). Alternatively, the model can be specified implicitly through its conditional distributions $p(\theta|y)$ and $p(y|\theta)$. Joint modeling simplifies theoretical analysis because closed-form expressions are available, whereas conditional modeling is less accessible, but more flexible and interpretable. However, an arbitrary pair of conditional distributions can be *incompatible*, meaning that there exists no joint distribution which produces these conditionals, and if it exists it does not have to be unique (Arnold and Press, 1989). Arnold et al. (2001) argues that "in general, reasonable-seeming conditional models will not be compatible with any single joint distribution". For example, consider the following Bayesian model with real-valued latent variables θ and observations y: for an improper prior $\pi(\theta) = 1$ and a Gaussian likelihood $f(y|\theta) = \mathcal{N}(y|\theta, 1)$ Bayes' theorem yields the posterior $p(\theta|y) = \mathcal{N}(\theta|y, 1)$. Even though both conditional distributions—the likelihood and the posterior—are proper distributions, there exists no proper joint distribution because the corresponding marginal π is improper. Hence, the conditionals f and p are incompatible. But incompatibility is no all-or-nothing property: even if there exists no joint distribution, one might still look for the joint distribution that is "most" compatible with the given conditionals, which leads to notions such as near-compatibility and ε-compatibility (Arnold et al., 2002). There exist algorithms for assessing the compatibility of conditional distributions (Kuo and Wang, 2011; Kuo et al., 2017) based on fractions of conditional densities, but most of this theory is restricted to discrete settings.

Gibbs sampling is one of the most natural ways of accessing the joint distribution of a conditionally specified model. It works by using the conditional distributions to define a time-reversible Markov chain whose stationary distribution is the joint distribution (Geman and Geman, 1984; Hastings, 1970). Gibbs sampling is well-understood and theoretically sound if the conditionals are compatible, but what happens if they are incompatible? Despite the fact that no joint distribution exists, the Markov chain defined by the conditionals can still converge to a unique stationary distribution, which represents a compromise between the incompatible conditionals (Muré, 2019). In this case, Gibbs sampling is called pseudo-Gibbs sampling. Pseudo-Gibbs samplers are widely used, for example in dependency networks (Heckerman et al., 2001) and missing data imputation (Hughes et al., 2014; Van Buuren et al., 2006). Characterizing the stationary distribution of a pseudo-Gibbs sampler is ongoing research (Chen and Ip, 2014; Kuo and Wang, 2019; Muré, 2019).

3.2 Method

3.2.1 Preliminaries

Let $\pi(\theta)$ be a proper prior distribution on a space of latent variables $\theta \in \Theta$ and $f(y|\theta)$ a positive likelihood on a space of observations $y \in \mathcal{Y}$. The corresponding posterior distribution is denoted by $p(\theta|y)$. For every fixed y let $q(\theta|y)$ denote the approximation to the posterior given by the approximate method in question. For sampling-based methods this distribution cannot be evaluated because it is specified only implicitly through samples, which suffices for our diagnostic. We denote the families of distributions as $F \coloneqq (f(\cdot|\theta))_{\theta\in\Theta}$, $P \coloneqq (p(\cdot|y))_{y\in\mathcal{Y}}$, and $Q \coloneqq (q(\cdot|y))_{y\in\mathcal{Y}}$. The families F and Q are called *compatible* if there exists a joint distribution on $\Theta \times \mathcal{Y}$ which has F and Q as conditionals. They are called *incompatible* if they are not compatible (Arnold and Press, 1989).

Our goal is to understand the inductive bias of inference based on the approximations $q(\theta|y)$ in terms of an a priori preference for solutions. The bias is fully encoded in the original prior $\pi(\theta)$ if the approximation is perfect. However, a mismatch $q(\theta|y) \neq p(\theta|y)$ can introduce additional bias, which is not captured by the original prior. The main idea of this work is to treat the approximation as an exact posterior and look for the corresponding prior distribution $\tilde{\pi}(\theta)$. This new prior describes the combination of explicitly encoded bias $\pi(\theta)$ and implicitly incurred bias because of approximation mismatch. We can then compare those priors to gain insights into how the approximation changes the inductive bias.

3.2.2 Assessing the inductive bias of posterior approximations with Gibbs priors

This section describes the problem of finding a prior to the approximations from the perspective of incompatible conditionals. We first motivate the problem by considering fixed observations and then propose a solution based on pseudo-Gibbs sampling.

For a fixed observation $y \in \mathcal{Y}$, the implicit *pointwise prior* π_y corresponding to $q(\cdot|y)$ is defined via

$$\pi_y(\theta) \propto_\theta \frac{q(\theta|y)}{f(y|\theta)}\,. \tag{3.1}$$

This describes the inductive bias of the approximation $q(\cdot|y)$ for a fixed observation, but it is not necessarily the same across different observations. The pointwise prior π_y will depend on y if and only if the conditional families F and Q are incompatible, which is a simple consequence of the definition. Informally, the scatter of the family $(\pi_y)_{y\in\mathcal{Y}}$ is an indicator for the degree of compatibility: in the compatible case, all π_y are concentrated at some distribution $\pi_y \equiv \tilde{\pi}$, which is the implicit prior to the approximations. As the

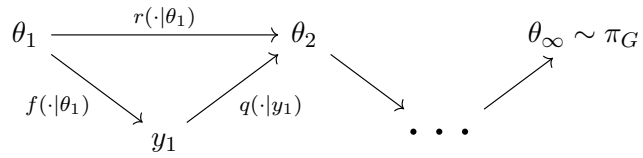


Figure 3.2: Schematic diagram of samples from the Gibbs chain (Definition 2) with auxiliary variables y_t. The distribution of θ_t converges to the Gibbs prior π_G.

compatibility decreases, $(\pi_y)_{y\in\mathcal{Y}}$ gets more scattered (see Figure 3.3a). One possible measure of incompatibility is discussed in Section 3.6. As a sanity check, observe that a perfect approximation $Q = P$ recovers the original prior $\pi = \pi_y$ for every y.

Ideally, the inductive bias of approximate inference could be explained by a single prior independent from the observation, like a prior in fully Bayesian inference. But as the above considerations show, this is not possible if the family $(\pi_y)_{y\in\mathcal{Y}}$ contains different members who offer conflicting explanations. Therefore, we search for a compromise that reasonably represents the different π_y. We do so by looking at the situation from the perspective of conditional distributions: a joint distribution on $\Theta \times \mathcal{Y}$ (Bayesian model) is specified indirectly through the conditionals F (likelihood) and Q (posterior approximations). We want to obtain the corresponding Θ-marginal (prior). A standard way to access the joint distribution via its conditionals is Gibbs sampling (Casella and George, 1992; Geman and Geman, 1984). Gibbs sampling starts with any initial point (θ_0, y_0) in the joint space and alternatingly updates θ given y and then y given θ. Under some assumptions, this vector converges to a sample from the joint distribution. Although Gibbs sampling assumes that the involved conditionals are compatible, it can be used the same way if they are incompatible. In this case it is referred to as *pseudo-Gibbs sampling*, a term coined by Heckerman et al. (2001). Pseudo-Gibbs sampling leads us to the following candidate prior:

Definition 2 (Gibbs prior). For two families of distributions $(f(\cdot|\theta))_{\theta\in\Theta}$ on $\mathcal{Y}$ and $(q(\cdot|y))_{y\in\mathcal{Y}}$ on Θ consider the discrete-time Markov chain on Θ whose transition function is given by

$$r(\theta'|\theta) = \mathbb{E}_{Y\sim f(\cdot|\theta)}\left[q(\theta'|Y)\right] . \tag{3.2}$$

This chain is called the *Gibbs chain*. Any stationary distribution of this Markov chain is called a *Gibbs prior* and denoted by π_G.

The Gibbs chain is illustrated in Figure 3.2. A single step of the chain according to Eq. (3.2) can be simulated with an auxiliary variable y: first sample from the likelihood $y \sim f(\cdot|\theta)$ and then from the approximation $\theta' \sim q(\cdot|y)$. Under the caveat of incompatibility, we have the following intuition for the Gibbs prior:

The Gibbs prior describes the a priori preference for solutions of the approximate inference method.

A simple reformulation of the stationarity condition for π_G offers two alternative representations

$$\pi_G(\theta) = \int_{\mathcal{Y}} g(y) q(\theta|y) \, \mathrm{d}y \tag{3.3}$$

$$= \int_{\mathcal{Y}} \tilde{g}(y) f(y|\theta) \pi_y(\theta) \, \mathrm{d}y \,, \tag{3.4}$$

where $g(y) = \int_\Theta \pi_G(\tilde{\theta}) f(y|\tilde{\theta}) \, \mathrm{d}\tilde{\theta}$ and $\tilde{g}(y) = g(y) / \int_\Theta \pi_y(\tilde{\theta}) f(y|\tilde{\theta}) \, \mathrm{d}\tilde{\theta}$ are weighting functions and Eq. (3.4) requires all π_y to be proper. Eq. (3.3) shows that the Gibbs prior is a mixture of the pointwise approximations. This suggests that consistent trends between approximations and posteriors are reflected in the Gibbs prior, for example underestimation of the norm as in Figure 3.1. Eq. (3.4) relates back to our original motivation of a compromise between $(\pi_y)_{y \in \mathcal{Y}}$ and shows that the Gibbs prior is a mixture of these distributions, reweighted by the likelihood.

Proposition 3 (Existence and uniqueness of Gibbs priors). *Consider two families of distributions $F = (f(\cdot|\theta))_{\theta \in \Theta}$ on $\mathcal{Y}$ and $Q = (q(\cdot|y))_{y \in \mathcal{Y}}$ on Θ. Let M be the corresponding Gibbs chain from Definition 2.*

(i) If F and Q are compatible with joint distribution $p(\theta, y)$, then the marginal $p(\theta)$ is a Gibbs prior. If M is additionally irreducible, then it is the only Gibbs prior.

(ii) If Θ and $\mathcal{Y}$ are finite, then there exists a Gibbs prior. If additionally F or Q are positive, then the Gibbs prior is unique.

Proof (sketch). The first statement of part (i) is a standard Gibbs sampling result; it can be proven by verifying the detailed balance equation for $p(\theta)$, which implies that M is a reversible Markov chain and $p(\theta)$ a stationary distribution. The statement about uniqueness is trivial, because Gibbs priors are defined as stationary distributions of M. A list of sufficient criteria in different settings is given in Arnold and Press (1989). Part (ii) concerns the existence of a (unique) stationary distribution. This condition is a standard result for finite Markov chains, for more general cases see Norris and Norris (1998). □

Proposition 3 admits additional interpretations in our Bayesian setting, where F is the likelihood and Q some approximation to the posterior. Part (i) states that if Q is the exact posterior under some other prior $\tilde{\pi}$, then this prior is recovered by the Gibbs prior $\pi_G = \tilde{\pi}$. Part (ii) shows that Gibbs priors exist under much weaker assumptions than compatibility of F and Q. There are only few other results about the Gibbs chain and its Gibbs priors in the general incompatible case. Muré (2019) shows that Gibbs priors are an optimal compromise between incompatible conditionals among a restricted set of distributions. For discrete distributions, Kuo and Wang (2019) show that the transitions of the Gibbs chain can be interpreted as iterative projections with respect to the KL divergence.

3.2.3 Sampling from the Gibbs prior

Algorithm 2 Simulating the Gibbs chain[3]

input Likelihood f, approximate inference method q, number of steps T
output Correlated samples $(\theta_1, \ldots, \theta_T)$ from π_G

$\theta_0 \leftarrow$ Arbitrary initialization, e. g. sample from $\pi(\cdot)$
for $t \leftarrow 0$ **to** $T - 1$ **do**
  $y_t \leftarrow$ Randomly sample from $f(\cdot|\theta_t)$
  $q(\cdot|y_t) \leftarrow$ Approximation to $p(\cdot|y_t)$
  $\theta_{t+1} \leftarrow$ Randomly sample from $q(\cdot|y_t)$
end for

Algorithm 2 describes how to obtain a sequence of correlated samples from the Gibbs prior. Since it is defined as the stationary distribution of the Gibbs chain, this is achieved by simply simulating the chain as in Figure 3.2. This approach is very generally applicable because it only requires sampling from the approximate posteriors, but not evaluating their density. The complexity depends largely on the complexity of computing the approximations to the posterior, which has to be redone every step for a different observation. The number of steps needed to assure convergence depends on the mixing speed of the Markov chain. Under the exact posterior, the Gibbs chain mixes fast if there are few observations. Informally, the posterior $p(\theta|y) \propto \pi(\theta)f(y|\theta)$ relies heavily on the the prior π (the stationary distribution) which ensures that the chain converges to its stationary distribution quickly. When there are many observations, the posterior concentrates and the high correlation between parameters and observations leads to slow mixing. In that sense, Algorithm 2 is more practical under few observations; this case is arguably more interesting because posterior inference gets easier as the number of observations increases. To ensure that the resulting samples actually correspond to the Gibbs prior, we recommend to monitor convergence of the Gibbs chain (Roy, 2020).

3.2.4 How to use the Gibbs prior

There are two principled ways of using the Gibbs prior to diagnose an approximate inference method. The first way is to assess the quality of the approximation by quantifying the distance to the original prior π with some divergence measure $D(\pi_G, \pi)$, or testing the hypothesis $H_0 : \pi_G = \pi$. A large discrepancy between π_G and π indicates a bad approximation, because a perfect approximation would yield $\pi_G = \pi$. The second way is to understand the inductive bias that the approximation imposes by examining the shift in mass from π to π_G. A direct comparison might not be enlightening if the latent space Θ is large; instead, one could visualize their differences (Lloyd and Ghahramani, 2015) or compare the distribution of summary statistics $g\colon \Theta \to \mathbb{R}$.

[3]Code available at https://github.com/tml-tuebingen/gibbs-prior-diagnostic

Note that there are caveats to this interpretation of the Gibbs prior due to incompatibility of likelihood and approximations. Thinking of the Gibbs prior as the effectively used prior for approximate inference becomes less valid for stronger incompatibility, because the family of pointwise priors $(\pi_y)_{y\in\mathcal{Y}}$ requires a stronger compromise. This is also demonstrated in the next section.

Summary We conclude this section by summarizing the three broad cases that can occur when comparing the Gibbs prior π_G with the original prior π:

1. $\pi_G \approx \pi$: the Gibbs prior is close to the original prior, which suggests that the approximations do not introduce additional bias. In particular, this is the case when the approximations are close to the true posterior. The reverse implication is not necessarily true (Section 3.5.1).
2. $\pi_G \neq \pi$: the Gibbs prior differs from the original prior, which implies that the approximations differ from the true posterior. This means that the approximations introduce additional bias, which can be assessed by interpreting the Gibbs prior as the effectively used prior. The validity of this interpretation depends on the compatibility between likelihood and approximations.
3. The Gibbs chain in Algorithm 2 does not converge. This can have multiple reasons: the approximations are good but the prior π is improper, the approximations are bad, or the chain was not run long enough. We recommend to use the diagnostic conservatively and dismiss it in these cases to avoid falsely rejecting a good approximation. To exclude the last case of running the Gibbs chain not long enough, the convergence of the chain should be monitored.

3.3 Illustrative toy example

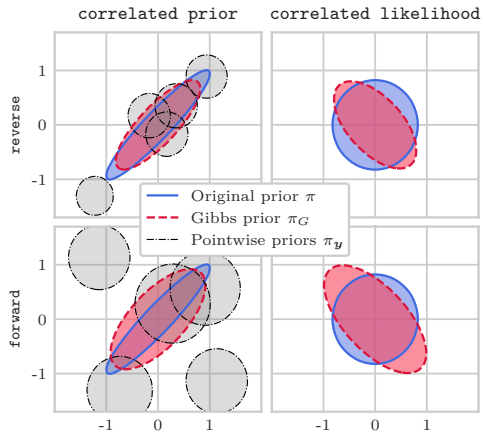
We now give a simple example to demonstrate the concepts from the previous section.

3.3.1 Gaussian toy model

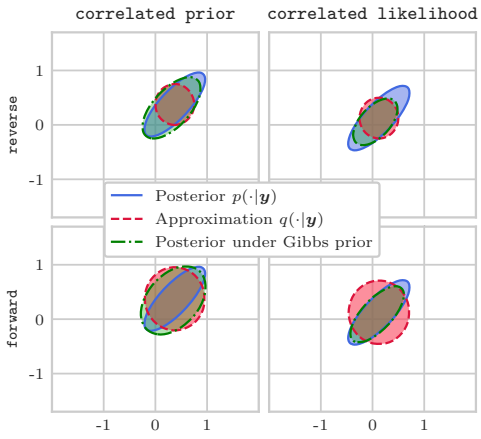
Consider the problem of estimating the mean $\theta \in \mathbb{R}^d$ of a d-dimensional Gaussian distribution with known covariance matrix based on n independent samples $y_1, \dots, y_n \in \mathbb{R}^d$. Placing a Gaussian prior on θ yields the Bayesian model

$$\begin{aligned} \theta &\sim \mathcal{N}(\mu_0, \Sigma_0)\,, \\ y_i|\theta &\overset{\text{indep.}}{\sim} \mathcal{N}(\theta, \Sigma)\,, \quad i = 1, \dots, n\,, \end{aligned} \tag{3.5}$$

where $\mu_0 \in \mathbb{R}^d$ and $\Sigma_0, \Sigma \in \mathbb{R}^{d\times d}$ are positive definite. The observations are collected in a matrix $\boldsymbol{y} = (y_1, \dots, y_n)^\top \in \mathbb{R}^{n\times d}$. We consider four different settings for variational inference in this model, which are determined by the following two choices:



(a) **Prior distributions.** Original prior, Gibbs prior, and pointwise priors for different $\boldsymbol{y}$ (same in both plots).



(b) **Posterior distributions.** Posterior, its approximation, and posterior under the Gibbs prior at fixed $\boldsymbol{y}$.

Figure 3.3: Distributions of interest for the variational inference settings described in Section 3.3.1 with $d = 2$ and $n = 1$. The setting `correlated prior` uses $\Sigma_0 = I$ and a Σ which is strongly correlated along $(1 \quad 1)^\top$. For `correlated likelihood` Σ_0 and Σ are interchanged. Colored areas show superlevel density sets with mass 0.3.

Correlated posterior We choose the prior and likelihood covariance matrices such that the posterior distribution has correlated components. This can be achieved by either a correlated prior and isotropic likelihood (referred to as `correlated prior`) or an isotropic prior and a correlated likelihood (referred to as `correlated likelihood`).

Variational approximation We consider the mean field variational approximation (Bishop, 2006). This method approximates the posterior with the variational family $\mathcal{Q}_{\text{MF}}$, which consists of all distributions on $\mathbb{R}^d$ with independent components. For the objective we consider the commonly used reverse KL divergence

$$q(\cdot|\boldsymbol{y}) \coloneqq \operatorname*{arg\,min}_{q \in \mathcal{Q}_{\text{MF}}} \text{KL}\left(q \;||\; p(\cdot|\boldsymbol{y})\right) \tag{3.6}$$

(referred to as `reverse`) or the forward KL divergence

$$q(\cdot|\boldsymbol{y}) \coloneqq \operatorname*{arg\,min}_{q \in \mathcal{Q}_{\text{MF}}} \text{KL}\left(p(\cdot|\boldsymbol{y}) \;||\; q\right) \tag{3.7}$$

(referred to as `forward`).

These settings are simple enough so that all distributions of interest are Gaussians and can be computed in closed form. This includes the posteriors $p(\cdot|\boldsymbol{y})$, the approximations $q(\cdot|\boldsymbol{y})$, the pointwise priors $\pi_{\boldsymbol{y}}$, and the Gibbs prior π_G. For details see Section 3.4, which also provides numerical justifications for the following arguments about biases.

3.3.2 Bias discovery using the Gibbs prior

Both approximations `reverse` and `forward` have two known biases, compactness and loss of correlation (Turner and Sahani, 2011). These biases can now also be discovered with the Gibbs prior. Figure 3.3a shows the priors and Gibbs priors and Figure 3.3b shows the corresponding posteriors and approximations.

Bias: compactness One known bias of mean field variational inference is the compactness of the approximations as measured by the entropy (Turner and Sahani, 2011): comparing the approximations to the true posterior in Figure 3.3b shows that they are too compact for `reverse` and not compact enough for `forward`. The same behavior can be observed on the prior level: the Gibbs prior is more compact than the prior for `reverse` and less compact for `forward`.

Bias: loss of correlation The variational approximations cannot capture any correlation between the coordinates by definition of the variational family $\mathcal{Q}_{\mathrm{MF}}$. This bias is easily understood on the posterior level, but it is less obvious what this means in terms of an a priori preference for solutions. In fact, this corresponding preference depends on the source of the posterior correlation and cannot be explained by the posterior alone. For `correlated prior`, the posterior correlation is caused by the prior correlation. Uncorrelated approximations therefore correspond to an uncorrelated prior. The Gibbs priors confirm this intuition by being less correlated than the prior. For `correlated likelihood`, the posterior correlation is caused by the likelihood correlation. Here, the Gibbs priors show that the approximations correspond to a prior whose correlation is orthogonal to the likelihood correlation. Intuitively, the orthogonal correlations of prior and likelihood "cancel out" to produce uncorrelated posteriors.

3.3.3 Is the Gibbs prior a prior?

The approximations are exact posteriors under the Gibbs prior if and only if the approximations are compatible to the likelihood. Equivalently, this is the case when the family of pointwise priors $(\pi_{\boldsymbol{y}})_{\boldsymbol{y}\in\mathcal{Y}}$ concentrates at a single distribution. Figure 3.3a shows $\pi_{\boldsymbol{y}}$ for various $\boldsymbol{y}$. For `correlated prior` they differ strongly and for `correlated likelihood` they are improper and therefore not shown. In both settings, this implies that the conditionals are incompatible as is typically the case. This is confirmed by Figure 3.3b, which shows that the posteriors under the Gibbs prior do not exactly coincide with the approximations. Despite these incompatibilities, this example shows that the Gibbs prior can discover inductive biases of the approximate methods. The Gibbs prior should therefore be thought of as a summary statistic for the inductive bias (see Section 3.5 for more details).

3.4 Results and proofs for the Gaussian toy example

3.4.1 Distributions in the Gaussian toy example

This section computes the distributions of interest for the Bayesian model defined in Eq. (3.5) from Section 3.3. This includes the posterior, the approximations, and the pointwise prior in Proposition 4, as well as the Gibbs prior in Theorem 5.

Proposition 4 (Posterior, approximations, and pointwise priors). *Consider the Bayesian model defined in Eq. (3.5) and let $\theta \in \Theta$ and $\boldsymbol{y} = (y_1, \ldots, y_n) \in \mathcal{Y}$.*

(i) The posterior distribution is given by

$$p(\theta|\boldsymbol{y}) = \mathcal{N}(\theta|\mu_n, \Sigma_n) \,, \tag{3.8}$$

where $\Sigma_n = \left(\Sigma_0^{-1} + n\Sigma^{-1}\right)^{-1}$, $\mu_n = \Sigma_n \left(\Sigma_0^{-1}\mu_0 + n\Sigma^{-1}\overline{\boldsymbol{y}}\right)$, and $\overline{\boldsymbol{y}} = 1/n \sum_{j=1}^n y_j$.

(ii) The mean field variational approximation is given by

$$q(\theta|\boldsymbol{y}) = \mathcal{N}(\theta|\mu_n, \Lambda_n) \,, \tag{3.9}$$

where

$$\Lambda_n = \begin{cases} \operatorname{diag}\left(\Sigma_n^{-1}\right)^{-1}, & \text{for } q \text{ defined via Eq. (3.6) (}\texttt{reverse}\text{)} \\ \operatorname{diag}\left(\Sigma_n\right), & \text{for } q \text{ defined via Eq. (3.7) (}\texttt{forward}\text{)} \end{cases} . \tag{3.10}$$

Hereby, the diag *operator keeps the diagonal entries of a matrix and sets all off-diagonal entries to 0.*

(iii) Whether the pointwise prior $\pi_{\boldsymbol{y}}$ is a proper distribution depends on the matrix $\Lambda_n^{-1} - n\Sigma^{-1}$. If it is positive definite, then

$$\pi_{\boldsymbol{y}}(\theta) \propto \mathcal{N}\left(\theta \middle| \mu_{\boldsymbol{y}}, \tilde{\Sigma}\right) \,,$$

where $\tilde{\Sigma} = \left(\Lambda_n^{-1} - n\Sigma^{-1}\right)^{-1}$ and $\mu_{\boldsymbol{y}} = \tilde{\Sigma}\left(\Lambda_n^{-1}\mu_n - n\Sigma^{-1}\overline{\boldsymbol{y}}\right)$. Otherwise, $\pi_{\boldsymbol{y}}$ is improper. In particular, $\pi_{\boldsymbol{y}}$ is always proper in the setting `correlated prior`*.*

Theorem 5 (Gibbs prior). *The Gibbs marginal π_G to the Bayesian model defined in Eq. (3.5) is given by*

$$\pi_G(\theta) = \mathcal{N}(\theta|\mu_0, \Sigma_G) \,. \tag{3.11}$$

Hereby, μ_0 is the mean of the prior distribution π and Σ_G satisfies the Lyapunov equation

$$A\Sigma_G A^\top - \Sigma_G + B = 0 \tag{3.12}$$

where $A = n\Sigma_n\Sigma^{-1}$ and $B = \Lambda_n + n\Sigma_n\Sigma^{-1}\Sigma_n$ with Σ_n and Λ_n defined as in Proposition 4.

Proof of Proposition 4. Recall the density function of a multivariate normal distribution $\mathcal{N}(\theta|\mu,\Sigma)$, which is given by

$$\begin{aligned}\mathcal{N}(\theta|\mu,\Sigma) &= (2\pi)^{-\frac{d}{2}}\det(\Sigma)^{-\frac{1}{2}}\exp\left(-\frac{1}{2}(\theta-\mu)^\top\Sigma^{-1}(\theta-\mu)\right)\\ &\propto_\theta \exp\left(-\frac{1}{2}\left[\theta^\top\Sigma^{-1}\theta - 2\theta^\top\Sigma^{-1}\mu\right]\right).\end{aligned}$$

Proof of (i)**.** First observe that up to proportionality the likelihood $f(\boldsymbol{y}|\theta)$ as a function of θ depends only on the average observation $\overline{\boldsymbol{y}} = 1/n\sum_{j=1}^n y_j$

$$\begin{aligned}f(\boldsymbol{y}|\theta) = \prod_{j=1}^n \mathcal{N}(y_j|\theta,\Sigma) &\propto_\theta \prod_{j=1}^n \exp\left(-\frac{1}{2}\left[\theta^\top\Sigma^{-1}\theta - 2\theta^\top\Sigma^{-1}y_j\right]\right)\\ &= \exp\left(-\frac{1}{2}\left[\theta^\top n\Sigma^{-1}\theta - 2\theta^\top n\Sigma^{-1}\overline{\boldsymbol{y}}\right]\right)\\ &\propto_\theta \mathcal{N}(\overline{\boldsymbol{y}}|\theta, 1/n\Sigma)\,. \end{aligned} \tag{3.13}$$

Together with $\pi(\theta) = \mathcal{N}(\theta|\mu_0,\Sigma_0)$, Bayes' theorem yields

$$\begin{aligned}p(\theta|\boldsymbol{y}) &\propto_\theta \pi(\theta)f(\boldsymbol{y}|\theta)\\ &\propto_\theta \exp\left(-\frac{1}{2}\left[\theta^\top\Sigma_0^{-1}\theta - 2\theta^\top\Sigma_0^{-1}\mu_0\right]\right)\exp\left(-\frac{1}{2}\left[\theta^\top n\Sigma^{-1}\theta - 2\theta^\top n\Sigma^{-1}\overline{\boldsymbol{y}}\right]\right)\\ &= \exp\left(-\frac{1}{2}\left[\theta^\top\left(\Sigma_0^{-1}+n\Sigma^{-1}\right)\theta - 2\theta^\top\left(\Sigma_0^{-1}\mu_0 + n\Sigma^{-1}\overline{\boldsymbol{y}}\right)\right]\right)\\ &= \exp\left(-\frac{1}{2}\left[\theta^\top\Sigma_n^{-1}\theta - 2\theta^\top\Sigma_n^{-1}\mu_n\right]\right)\\ &\propto \mathcal{N}(\theta|\mu_n,\Sigma_n)\,.\end{aligned}$$

Note that $\Sigma_0^{-1} + n\Sigma^{-1}$ is positive definite as the sum of two positive definite matrices, and therefore $\Sigma_n = \left(\Sigma_0^{-1} + n\Sigma^{-1}\right)^{-1}$ is positive definite as well.

Proof of (ii)**.** By definition of the mean-field variational family, every variational density factorizes as $q(\theta|\boldsymbol{y}) = \prod_{j=1}^m q_j(\theta_j)$. For the setting `forward` we refer to (Bishop, 2006, Section 10.1.2), where it is shown that the optimal q_j simply coincide with the marginal densities of the posterior $q_j(\theta_j) = p(\theta_j|\boldsymbol{y}) = \mathcal{N}\left(\theta_j\middle|(\mu_n)_j,(\Sigma_n)_{j,j}\right)$.

For the other setting `reverse` let $\mathbb{E}_{-j}$ denote the expectation over all latent variables θ_i except θ_j with respect to the factorized distribution $\prod_{i\neq j} q_i(\theta_i)$. To simplify the following computation, we abbreviate $\Sigma_n^{-1} =: \Gamma$ and $\mu_n =: \mu$ (now μ_k refers to the k-th

component of μ_n). We use that the optimal solution satisfies the recursive update rule

$$
\begin{aligned}
q_j(\theta_j) &\propto_{\theta_j} \exp\left(\mathbb{E}_{-j} \log p(\theta, \boldsymbol{y})\right) && \text{(Bishop (2006))} \\
&\propto_{\theta_j} \exp\left(\mathbb{E}_{-j} \log p(\theta|\boldsymbol{y})\right) \\
&\propto_{\theta_j} \exp\left(-\frac{1}{2}\mathbb{E}_{-j}\left[(\theta-\mu)^\top \Gamma\,(\theta-\mu)\right]\right) && \left(p(\theta|\boldsymbol{y}) = \mathcal{N}\left(\theta\middle|\mu, \Gamma^{-1}\right) \text{ by Eq. (3.8)}\right) \\
&\propto_{\theta_j} \exp\left(-\frac{1}{2}\left(\Gamma_{j,j}\,(\theta_j-\mu_j)^2 + 2\,(\theta_j-\mu_j)\sum_{k\neq j}\Gamma_{j,k}\,(m_k-\mu_k)\right)\right) \\
& && (m_k \coloneqq \mathbb{E}_{q_k}\theta_k) \\
&\propto_{\theta_j} \exp\left(-\frac{1}{2\Gamma_{j,j}^{-1}}\left(\theta_j - \mu_j + \frac{1}{\Gamma_{j,j}}\sum_{k\neq j}\Gamma_{j,k}(m_k-\mu_k)\right)^2\right) \\
&\propto_{\theta_j} \mathcal{N}\left(\theta_j\middle|\mu_j - \frac{1}{\Gamma_{j,j}}\sum_{k\neq j}\Gamma_{j,k}(m_k-\mu_k), \Gamma_{j,j}^{-1}\right) \\
&= \mathcal{N}\left(\theta_j\middle|m_j, \Gamma_{j,j}^{-1}\right). && \text{(Definition of } m_j\text{)}
\end{aligned}
$$

This shows that the solutions q_j are normally distributed and have the claimed variance $\Gamma_{j,j}^{-1} = \left(\Sigma_n^{-1}\right)_{j,j}^{-1}$. However, their means m_j are only recursively determined and to conclude the proof, we need to show that $m_j = \mu_j$. The last equation in the previous computation gives the recursive relation

$$
\begin{aligned}
& m_j = \mu_j - \frac{1}{\Gamma_{j,j}}\sum_{k\neq j}\Gamma_{j,k}(m_k-\mu_k) && \forall j = 1,\dots,m \\
\Leftrightarrow\quad & \frac{1}{\Gamma_{j,j}}\sum_{k=1}^{m}\Gamma_{j,k}m_k = \frac{1}{\Gamma_{j,j}}\sum_{k=1}^{m}\Gamma_{j,k}\mu_k && \forall j = 1,\dots,m \\
\Leftrightarrow\quad & \langle\Gamma_j, \boldsymbol{m}\rangle = \langle\Gamma_j, \mu\rangle && \forall j = 1,\dots,m \\
\Leftrightarrow\quad & \Gamma\boldsymbol{m} = \Gamma\mu\,,
\end{aligned}
$$

where Γ_j denotes the j-th row of Γ and $\boldsymbol{m}$ is the vector containing all m_k. Since Γ is positive definite, the last equality implies $\boldsymbol{m} = \mu$ and concludes the proof.

Proof of *(iii)***.** We can compute the pointwise prior $\pi_{\boldsymbol{y}}(\theta)$ with its definition in Eq. (3.1) with Eq. (3.9) for $q(\theta|\boldsymbol{y})$ and Eq. (3.13) for $f(\boldsymbol{y}|\theta)$ as

$$
\begin{aligned}
\pi_{\boldsymbol{y}}(\theta) \propto_\theta \frac{q(\theta|\boldsymbol{y})}{f(\boldsymbol{y}|\theta)} &\propto_\theta \frac{\mathcal{N}\left(\theta|\mu_n, \Lambda_n\right)}{\mathcal{N}\left(\overline{\boldsymbol{y}}|\theta, 1/n\Sigma\right)} && (3.14) \\
&\propto_\theta \frac{\exp\left(-\frac{1}{2}\left[\theta^\top\Lambda_n^{-1}\theta - 2\theta^\top\Lambda_n^{-1}\mu_n\right]\right)}{\exp\left(-\frac{1}{2}\left[\theta^\top n\Sigma^{-1}\theta - 2\theta^\top n\Sigma^{-1}\overline{\boldsymbol{y}}\right]\right)} \\
&= \exp\left(-\frac{1}{2}\left[\theta^\top\left(\Lambda_n^{-1} - n\Sigma^{-1}\right)\theta - 2\theta^\top\left(\Lambda_n^{-1}\mu_n - n\Sigma^{-1}\overline{\boldsymbol{y}}\right)\right]\right). && (3.15)
\end{aligned}
$$

If $\Lambda_n^{-1} - n\Sigma^{-1}$ is positive definite, we can continue the computation

$$\begin{aligned}\pi_{\boldsymbol{y}}(\theta) &\propto_\theta \exp\left(-\frac{1}{2}\left[\theta^\top\left(\Lambda_n^{-1} - n\Sigma^{-1}\right)\theta - 2\theta^\top\left(\Lambda_n^{-1}\mu_n - n\Sigma^{-1}\overline{\boldsymbol{y}}\right)\right]\right)\\ &\propto_\theta \exp\left(-\frac{1}{2}\left[\theta^\top\tilde{\Sigma}^{-1}\theta - 2\theta^\top\tilde{\Sigma}^{-1}\mu_{\boldsymbol{y}}\right]\right)\\ &\propto_\theta \mathcal{N}\left(\theta\middle|\mu_{\boldsymbol{y}}, \tilde{\Sigma}\right).\end{aligned}$$

If $\Lambda_n^{-1} - n\Sigma^{-1} =: S$ is not positive definite, then we can show that $\pi_{\boldsymbol{y}}$ is improper. In this case, S has an eigenvalue $\lambda \leq 0$ and corresponding eigenvector $v \in \mathbb{R}^d$ with $\|v\| = 1$. Consider the hypercylinder A of points whose distance to the axis $\mathbb{R}v$ is at most 1, formally defined as

$$A \coloneqq \{\theta \in \mathbb{R}^d \mid \theta = tv + w, \text{ where } t \in \mathbb{R}, w \in v^\perp, \|w\| = 1\},$$

where $v^\perp = \{w \in \mathbb{R}^d \ : \ \langle v, w\rangle = 0\}$. Abbreviate $\gamma \coloneqq \Lambda_n^{-1}\mu_n - n\Sigma^{-1}\overline{\boldsymbol{y}}$ and collect all constants in $C > 0$ (can change at different steps). Then we can lower bound the right hand side of Eq. (3.14) on A via

$$\begin{aligned}\frac{q(\theta|\boldsymbol{y})}{f(\boldsymbol{y}|\theta)} &= C\exp\left(-\frac{1}{2}\theta^\top S\theta + \langle\theta, \gamma\rangle\right)\\ &= C\exp\left(-\frac{1}{2}(tv+w)^\top S(tv+w) + \langle tv+w, \gamma\rangle\right) && (\theta = tv + w \in A)\\ &= C\exp\left(\underbrace{-\frac{1}{2}\lambda t^2}_{\geq 0} + \langle v, \gamma\rangle t + \langle w, \gamma\rangle - \frac{1}{2}w^\top Sw\right)\\ & && (Sv = \lambda v,\ \|v\| = 1,\ \langle v, w\rangle = 0)\\ &\geq C\exp\left(\langle v, \gamma\rangle t + \langle w, \gamma\rangle - \frac{1}{2}w^\top Sw\right)\\ &\geq C\exp\left(\langle v, \gamma\rangle t\right). && (\langle w, \gamma\rangle - \tfrac{1}{2}w^\top Sw \text{ is bounded for } \|w\| \leq 1)\end{aligned}$$

Using this lower bound, we can lower bound the integral over A through

$$\int_A \frac{q(\theta|\boldsymbol{y})}{f(\boldsymbol{y}|\theta)}\,\mathrm{d}\theta \geq C\int_{\mathbb{R}} \exp\left(\langle v, \gamma\rangle t\right)\mathrm{d}t = \infty.$$

Hence $\pi_{\boldsymbol{y}}$ is improper.

The last statement is that $\pi_{\boldsymbol{y}}$ is always proper in the setting `correlated prior` ($\Sigma = I$), which means we need to show that S has strictly positive eigenvalues.

We treat the cases `reverse` and `forward` separately. For `reverse`, it is

$$S = \Lambda_n^{-1} - n\Sigma^{-1} = \operatorname{diag}\left(\Sigma_0^{-1} + nI\right) - nI = \operatorname{diag}\left(\Sigma_0^{-1}\right). \quad \text{(diag is a linear operator)}$$

The diagonal entries of the symmetric positive definite matrix Σ_0^{-1} are lower bounded by its smallest eigenvalue $\lambda_{\min}\left(\Sigma_0^{-1}\right) > 0$, which follows from the Courant–Fischer–Weyl min-max principle. Since these diagonal entries are the eigenvalues of S, this implies that S is positive definite. For the other case `forward`, we have

$$S = \Lambda_n^{-1} - n\Sigma^{-1} = \operatorname{diag}\left(\left(\Sigma_0^{-1} + n\right)^{-1}\right)^{-1} - n\,.$$

We again need to bound the diagonal elements of $\left(\Sigma_0^{-1} + n\right)^{-1}$ with its eigenvalues. A similar argument as above yields

$$\lambda_{\max}\left(\operatorname{diag}\left(\left(\Sigma_0^{-1} + n\right)^{-1}\right)\right) \leq \lambda_{\max}\left(\left(\Sigma_0^{-1} + n\right)^{-1}\right) = \frac{1}{\lambda_{\min}\left(\Sigma_0^{-1}\right) + n}\,. \tag{3.16}$$

With this, the eigenvalues of S are bouded by

$$\begin{aligned}
\lambda_{\min}(S) = \lambda_{\min}\left(\operatorname{diag}\left(\left(\Sigma_0^{-1} + n\right)^{-1}\right)^{-1} - n\right) &= \frac{1}{\lambda_{\max}\left(\operatorname{diag}\left(\left(\Sigma_0^{-1} + n\right)^{-1}\right)\right)} - n \\
&\geq \frac{1}{\frac{1}{\lambda_{\min}\left(\Sigma_0^{-1}\right)+n}} - n \qquad \text{(Eq. (3.16))} \\
&= \lambda_{\min}\left(\Sigma_0^{-1}\right) > 0\,.
\end{aligned}$$

□

To prove Theorem 5 we require some general properties of Gaussian densities in Lemma 6 and Lemma 7 because the proofs consist mainly of rearranging Gaussian densities. Next we compute the transition function of the Markov chain from Definition 2 in Proposition 8. We then prove Theorem 5 by guessing that the stationary distribution is Gaussian and verifying the stationary equation.

Lemma 6 (Some properties of Gaussians). *Let $\mathcal{N}(x|\mu,\Sigma)$ denote the density of a Gaussian distribution $\mathcal{N}(\mu,\Sigma)$ on $\mathbb{R}^d$ at $x \in \mathbb{R}^d$ with mean $\mu \in \mathbb{R}^d$ and positive definite covariance $\Sigma \in \mathbb{R}^{d\times d}$. Then the following equalities hold*

(i) $\mathcal{N}(x+y|\mu,\Sigma) = \mathcal{N}(x|\mu-y,\Sigma)$ *and* $\mathcal{N}(x|\mu,\Sigma) = \mathcal{N}(\mu|x,\Sigma)$ *for* $x, y \in \mathbb{R}^d$.

(ii) Let $A \in \mathbb{R}^{d\times d}$ be non-singular. Then $\mathcal{N}(Ax|\mu,\Sigma) = C_{A,\Sigma}\mathcal{N}\left(x\middle|A^{-1}\mu, A^{-1}\Sigma A^{-T}\right)$, where $C_{A,\Sigma} \in \mathbb{R}$ is a constant that depends only on A and Σ.

(iii) Let $\mu_1, \mu_2 \in \mathbb{R}^d$ and $\Sigma_1, \Sigma_2 \in \mathbb{R}^{d\times d}$ positive definite. Then the convolution of two Gaussian densities corresponds to the sum of two independent Gaussians, i.e., for $z \in \mathbb{R}^d$ it holds

$$\begin{aligned}
\int_{\mathbb{R}^d} \mathcal{N}(z-x|\mu_1,\Sigma_1)\,\mathcal{N}(x|\mu_2,\Sigma_2)\,\mathrm{d}x &= \left[\mathcal{N}(\cdot|\mu_1,\Sigma_1) * \mathcal{N}(\cdot|\mu_2,\Sigma_2)\right](z) \\
&= \mathcal{N}(z|\mu_1+\mu_2,\Sigma_1+\Sigma_2)\,.
\end{aligned}$$

Proof. Points (i) and (iii) are trivial. For (ii), we compute

$$\begin{aligned}\mathcal{N}(Ax|\mu,\Sigma) &= C_\Sigma \exp\left(-\frac{1}{2}(Ax-\mu)^\top\Sigma^{-1}(Ax-\mu)\right)\\ &= C_\Sigma \exp\left(-\frac{1}{2}(x-A^{-1}\mu)^\top A^\top\Sigma^{-1}A(x-A^{-1}\mu)\right)\\ &= C_{A,\Sigma}\mathcal{N}\left(x\middle|A^{-1}\mu, A^{-1}\Sigma A^{-T}\right).\end{aligned}$$

Note that $A^{-1}\Sigma A^{-T}$ is positive definite as well: symmetry is obvious and it holds

$$x^\top A^{-1}\Sigma A^{-T}x = (A^{-T}x)^\top\Sigma(A^{-T}x) > 0$$

for $x \neq 0$ because A is non-singular and Σ is positive definite. □

The next lemma computes an integral that appears both in computing the transition function of the Gibbs chain and in computing its stationary distribution.

Lemma 7. *Let $\theta', a, \mu_2 \in \mathbb{R}^d$, $A \in \mathbb{R}^{d\times d}$ non-singular, and $\Sigma_1, \Sigma_2 \in \mathbb{R}^{d\times d}$ positive definite. Then it is*

$$\int_{\mathbb{R}^d}\mathcal{N}\left(\theta'\middle|a+Ax,\Sigma_1\right)\mathcal{N}(x|\mu_2,\Sigma_2)\,\mathrm{d}x = \mathcal{N}\left(\theta'\middle|a+A\mu_2,\Sigma_1+A\Sigma_2A^\top\right).$$

Proof. We start by rephrasing the first density

$$\begin{aligned}\mathcal{N}\left(\theta'\middle|a+Ax,\Sigma_1\right) &= \mathcal{N}\left(Ax\middle|\theta'-a,\Sigma_1\right) && \text{(Lemma 6, } (i)\text{)}\\ &= C_{A,\Sigma_1}\mathcal{N}\left(x\middle|A^{-1}(\theta'-a),A^{-1}\Sigma_1A^{-T}\right) && \text{(Lemma 6, } (ii)\text{)}\\ &= C_{A,\Sigma_1}\mathcal{N}\left(A^{-1}\theta'-x\middle|A^{-1}a,A^{-1}\Sigma_1A^{-T}\right). && \text{(Lemma 6, } (i)\text{)}\end{aligned}$$

This yields

$$\begin{aligned}&\int_{\mathbb{R}^d}\mathcal{N}\left(\theta'\middle|a+Ax,\Sigma_1\right)\mathcal{N}(x|\mu_2,\Sigma_2)\,\mathrm{d}x\\ &\propto'_\theta \int_{\mathbb{R}^d}\mathcal{N}\left(A^{-1}\theta'-x\middle|A^{-1}a,A^{-1}\Sigma_1A^{-T}\right)\mathcal{N}(x|\mu_2,\Sigma_2)\,\mathrm{d}x\\ &= \mathcal{N}\left(A^{-1}\theta'\middle|A^{-1}a+\mu_2,A^{-1}\Sigma_1A^{-T}+\Sigma_2\right) && \text{(Lemma 6, } (iii)\text{)}\\ &\propto'_\theta \mathcal{N}\left(\theta'\middle|a+A\mu_2,\Sigma_1+A\Sigma_2A^\top\right). && \text{(Lemma 6, } (ii)\text{)}\end{aligned}$$

□

We can now compute the transition function of the Gibbs chain from Definition 2.

Proposition 8. *The transition function of the Gibbs chain is given by Gaussian distributions*

$$r(\theta'|\theta) = \mathcal{N}\left(\theta'\middle|a+A\theta,B\right), \tag{3.17}$$

where $\theta, \theta' \in \mathbb{R}^d$, $a = \Sigma_n\Sigma_0^{-1}\mu_0$, $A = n\Sigma_n\Sigma^{-1}$, and $B = \Lambda_n + n\Sigma_n\Sigma^{-1}\Sigma_n$.

Proof. Let $\theta, \theta' \in \mathbb{R}^d$. By definition, the transition function of the Gibbs chain is given by

$$r(\theta'|\theta) = \int_{\mathbb{R}^{n\times d}} q(\theta'|\boldsymbol{y}) f(\boldsymbol{y}|\theta)\,\mathrm{d}\boldsymbol{y} = \int_{\mathbb{R}^d} q(\theta'|\overline{\boldsymbol{y}})\overline{f}(\overline{\boldsymbol{y}}|\theta)\,\mathrm{d}\overline{\boldsymbol{y}}\,,$$

where we have transformed the integral to the mean $\overline{\boldsymbol{y}}$, because q only depends on $\boldsymbol{y}$ through $\overline{\boldsymbol{y}}$. The corresponding push-forward measure is given by $\overline{f}(\overline{\boldsymbol{y}}|\theta) = \mathcal{N}(\overline{\boldsymbol{y}}|\theta, 1/n\Sigma)$. Using Proposition 4 and expressing μ_n with a and A, we have that $q(\theta'|\overline{\boldsymbol{y}}) = \mathcal{N}(\theta'|\mu_n, \Lambda_n) = \mathcal{N}(\theta'|a + A\overline{\boldsymbol{y}}, \Lambda_n)$. Putting everything together, we get

$$\begin{aligned} r(\theta'|\theta) &= \int_{\mathbb{R}^d} \mathcal{N}\left(\theta'\middle|a + A\overline{\boldsymbol{y}}, \Lambda_n\right) \mathcal{N}\left(\overline{\boldsymbol{y}}\middle|\theta, \frac{1}{n}\Sigma\right) \mathrm{d}\overline{\boldsymbol{y}} \\ &= \mathcal{N}\left(\theta'\middle|a + A\theta, \Lambda_n + A\frac{1}{n}\Sigma A^\top\right)\,. \qquad \text{(Lemma 7)} \end{aligned}$$

The equality $A\frac{1}{n}\Sigma A^\top = n\Sigma_n\Sigma^{-1}\Sigma_n$ concludes the proof. □

We are now ready to prove Theorem 5.

Proof of Theorem 5. The proof is based on guessing that the stationary distribution is Gaussian. We first show that Gaussian distributions are closed under taking a step with the transition function and then derive the parameters of the stationary distribution based on the stationary equation.

Let $p(\theta) = \mathcal{N}(\theta|m, M)$ with $m \in \mathbb{R}^d$ and $M \in \mathbb{R}^{d\times d}$ positive definite. Using Lemma 7 and Proposition 8, the distribution after one step Rp is given by

$$\begin{aligned} Rp(\theta') = \int_{\mathbb{R}^d} r(\theta'|\theta)p(\theta)\,\mathrm{d}\theta &= \int_{\mathbb{R}^d} \mathcal{N}\left(\theta'\middle|a + A\theta, B\right) \mathcal{N}(\theta|m, M)\,\mathrm{d}\theta \\ &= \mathcal{N}\left(\theta'\middle|a + Am, B + AMA^\top\right)\,. \end{aligned} \tag{3.18}$$

If a p satisfies the stationary equation $p = Rp$, then it is the stationary distribution $p = \pi_G$. Using that Rp is again Gaussian, Eq. (3.18) shows that this is satisfied if and only if

$$m = a + Am \quad \text{and} \quad M = B + AMA^\top\,.$$

The solution for the mean equation is obtained by rearranging and plugging in the definitions of a, A and Σ_n:

$$\begin{aligned} m = (I - A)^{-1} a = \left(I - n\Sigma_n\Sigma^{-1}\right)^{-1} \Sigma_n\Sigma_0^{-1}\mu_0 &= \left(\Sigma_n^{-1} - n\Sigma^{-1}\right)\Sigma_0^{-1}\mu_0 \\ &= \left(\Sigma_0^{-1} + n\Sigma^{-1} - n\Sigma^{-1}\right)^{-1}\Sigma_0^{-1}\mu_0 \\ &= \mu_0\,. \end{aligned}$$

The stationary equation for the covariance matrix M is equivalent to the Lyapunov equation

$$AMA^\top - M + B = 0\,,$$

which has a unique solution. This concludes the proof. □

3.4.2 Numerical evaluation of the biases in the Gaussian toy example

This section presents numerical values that back up the statements about the biases from Section 3.3.2.

The first bias is the compactness of the mean-field approximations. Table 3.1 shows the compactness of all relevant distributions as measured by the entropy, which is given by $d/2(1+\ln(2\pi)) + 1/2\ln(\det\Sigma)$ for a Gaussian distribution $\mathcal{N}(\mu,\Sigma)$. Under the setting `forward` approximations q are less compact than the exact posterior p. This is reflected by the priors as the Gibbs prior π_G is less compact than the exact prior π. Under the setting `reverse`, this trend is reversed: approximations are more compact than the exact posterior, and the Gibbs prior is more compact than the exact prior.

The second bias is the loss of correlation under the mean-field approximations. Table 3.2 shows the correlation between different components θ_1 and θ_2 of the 2-dimensional latent variable $\theta = (\theta_1, \theta_2)$ only for the prior distributions, because the components of the approximations are by definition uncorrelated. Recall that the exact posterior distribution was the same in both settings, but under `correlated prior` the posterior correlation was due to prior correlation, whereas under `correlated likelihood` it was due to likelihood correlation. In the setting `correlated prior`, the Gibbs prior is less correlated than the prior. In the setting `correlated likelihood`, the prior is uncorrelated, but the Gibbs prior is negatively correlated to "cancel out" the positive correlation of the likelihood covariance.

Table 3.1: Compactness of various distributions across settings as measured by the entropy. First value is under the setting `correlated prior` and second value is under the setting `correlated likelihood`. Note that the covariance of exact and approximate posterior does not depend on the observation.

Entropy	`forward`	`reverse`
Prior π	2.24 / 2.84	2.24 / 2.84
Gibbs prior π_G	2.82 / 3.15	2.21 / 2.52
Exact posterior p	1.50 / 1.50	1.50 / 1.50
Approximate posterior q	1.97 / 1.97	1.02 / 1.02

Table 3.2: Correlation of prior distributions across settings as measured by the covariance $\mathrm{Cov}(\theta_1, \theta_2)$ between the components of $\theta = (\theta_1, \theta_2) \in \mathbb{R}^2$. First value is under the prior distribution π and second value is under the corresponding Gibbs prior π_G.

Covariance $\mathrm{Cov}(\theta_1, \theta_2)$	`forward`	`reverse`
`correlated prior`	1.45 / 0.91	1.45 / 0.74
`correlated likelihood`	0 / -1.13	0 / -0.52

3.5 The Gibbs prior is a summary statistic

3.5.1 Different approximations can have the same Gibbs prior

This section shows that there are fewer Gibbs priors than conditional distributions because different conditionals can define the same Gibbs chain. For the sake of simplicity, we consider the finite setting with latent space $\Theta = [n] = \{1, \dots, n\}$ and observation space $\mathcal{Y} = [m]$ with $n, m \in \mathbb{N}$. Here the likelihood is given by the stochastic matrix $F \in \mathbb{R}^{n \times m}$ and the approximation by another stochastic matrix $Q \in \mathbb{R}^{m \times n}$, that is, F and Q have non-negative entries and their rows sum to 1. The Gibbs chain from Definition 2 is defined via the transition matrix $P = FQ \in \mathbb{R}^{n \times n}$, which is again a stochastic matrix, and the Gibbs prior is a probability vector $\pi_G \in \mathbb{R}^n$.

The next proposition shows that different approximations $Q \neq \tilde{Q}$ can define the same Gibbs chain. In particular, they define the same Gibbs prior.

Proposition 9. *For $n \geq 2$ let $F \in \mathbb{R}^{n \times m}, Q \in \mathbb{R}^{m \times n}$ be stochastic matrices with entries in $(0, 1)$ and $\ker F \neq \{0\}$. Then there exists a stochastic matrix $\tilde{Q} \in \mathbb{R}^{m \times n}$ with $Q \neq \tilde{Q}$ that satisfies*

$$FQ = F\tilde{Q}\,. \tag{3.19}$$

In particular, both Markov chains have the same stationary distribution.

Proof. The main idea is to define a suitable perturbation W such that $\tilde{Q} = Q + W$ is a stochastic matrix that satisfies Eq. (3.19).

Let $0 \neq x_0 \in \ker F$ and $0 \neq w \in \mathbf{1}^\perp = \{x \in \mathbb{R}^n \mid x^\top \mathbf{1} = 0\}$, where $\mathbf{1} \in \mathbb{R}^n$ denotes the vector whose entries are all 1. The vectors x_0 and w can be chosen non-zero by the assumptions $\ker F \neq \{0\}$ and $n \geq 2$. With these vectors, we define the perturbation matrix $W \coloneqq x_0 w^\top \in \mathbb{R}^{m \times n}$ and $\tilde{Q} \coloneqq Q + W \in \mathbb{R}^{m \times n}$. First observe that $Q \neq \tilde{Q}$, because $x_0, w \neq 0$ implies $W \neq 0$. Using $x_0 \in \ker F$, we verify Eq. (3.19) by computing

$$F\tilde{Q} = F(Q + W) = FQ + \underbrace{Fx_0}_{=0} w^\top = FQ\,.$$

It remains to show that $\tilde{Q}$ is a stochastic matrix. We may assume that w was chosen such that the first condition $0 \leq \tilde{Q} = Q + W = Q + x_0 w^\top$ holds; otherwise, w can be scaled by an arbitrarily small constant such that this inequality is satisfied, which is always possible because $Q > 0$ by assumption. The other condition is that the rows of $\tilde{Q}$ sum to 1, which we verify with $Q\mathbf{1} = \mathbf{1}$ (because Q is a stochastic matrix) and $w \in \mathbf{1}^\perp$ by computing

$$\tilde{Q}\mathbf{1} = \underbrace{Q\mathbf{1}}_{=\mathbf{1}} + x_0 \underbrace{w^\top \mathbf{1}}_{=0} = \mathbf{1}\,.$$

For the second statement we only need to verify that the stationary distribution of the Markov chain defined with the transition matrix $P = FQ$ indeed uniquely exists. This is the case because the assumptions $F, Q > 0$ imply $P > 0$, hence the corresponding Markov chain is positive recurrent with finite state space. This implies the existence of a unique stationary distribution. □

Example 10. An example for Proposition 9 with $n = 2$ and $m = 3$ is given by the matrices

$$F = \begin{pmatrix} .1 & .4 & .5 \\ .3 & .2 & .5 \end{pmatrix}, \quad Q = \begin{pmatrix} .2 & .8 \\ .4 & .6 \\ .5 & .5 \end{pmatrix}, \quad \tilde{Q} = \begin{pmatrix} .1 & .9 \\ .3 & .7 \\ .6 & .4 \end{pmatrix},$$

which satisfy $Q \neq \tilde{Q}$ and

$$FQ = \begin{pmatrix} .43 & .57 \\ .39 & .61 \end{pmatrix} = F\tilde{Q}\,.$$

3.5.2 A weaker notion of compatibility between conditional distributions is sufficient

In this section, we argue that the notion of compatibility between conditional distributions is actually stricter than necessary for assessing whether the Gibbs prior provides a useful explanation. We do so by introducing a weaker notion of compatibility under which the Gibbs prior retains a strong interpretation. First, we recap the setting as presented in Section 3.2. For a Bayesian model with likelihood F we are given approximations Q to the true posterior, and our goal is to assess their inductive bias in terms of an a priori preference for solutions. We propose to consider another fully Bayesian model $\mathcal{M}_G$, specified with the same likelihood F and the Gibbs prior π_G. The Gibbs prior π_G can then be used to reason about the inductive bias of Q if the conditional distributions F and Q are compatible, because then the posteriors under $\mathcal{M}_G$ coincide with Q.

However, even when they are different, the Bayesian model $\mathcal{M}_G$ can accurately describe inference based on Q. This is achieved by considering the whole pipeline of inference instead of inference based on a fixed observation: starting with an unknown true latent parameter θ, we observe some data through the likelihood $y \sim f(\cdot|\theta)$, based on which we use the approximations to estimate the latent parameter $\theta' \sim q(\cdot|y)$. This process is summarized in the probabilities of estimating θ' if the true parameter is θ, which are precisely the transition probabilities of the Gibbs chain. Defining the same Gibbs chain as the approximations is therefore sufficient for the Bayesian model $\mathcal{M}_G$ to qualify as an interpretable reformulation. This leads to the following weaker notion of compatibility between conditional distributions:

Definition 11 (Weak compatibility of conditional distributions). For two families of conditional distributions $F = (f(\cdot|\theta))_{\theta\in\Theta}$ on $\mathcal{Y}$ and $Q = (q(\cdot|y))_{y\in\mathcal{Y}}$ on Θ, let π_G denote the corresponding Gibbs prior from Definition 2. Let $P_G = (p_G(\cdot|y))_{y\in\mathcal{Y}}$ denote the posteriors under to the Bayesian model specified by π_G and F. Then F and Q are called *weakly compatible*, if the Gibbs chain of F and Q coincides with the Gibbs chain of F and P_G, that is,

$$\mathbb{E}_{Y\sim f(\cdot|\theta)}\left[q(\theta'|Y)\right] = \mathbb{E}_{Y\sim f(\cdot|\theta)}\left[p_G(\theta'|Y)\right] \quad \forall\theta \in \Theta\,.$$

As the naming suggests, weak compatibility of two conditional distributions is strictly weaker than compatibility: compatibility trivially implies weak compatibility, whereas Proposition 9 shows that the converse is not true. Since the Gibbs prior can be used to reason about conditionals that are only weakly compatible, this means that it is useful in more situations than what compatibility suggests. In particular, there exist different conditional distributions Q which justifiably get assigned the same Gibbs prior, because they yield the same Gibbs chain.

3.6 Measuring the degree of compatibility

Most existing literature focuses on the question whether families of conditional distributions are exactly compatible (Kuo and Wang, 2011; Kuo et al., 2017). However, in the context of this work, the more relevant question is how incompatible they are, requiring a practical way to measure the degree of compatibility. This leads to notions such as near-compatibility and ε-compatibility (Arnold et al., 2002) and is based on computing some distance between joint distributions, which involve the conditional distributions (Ghosh and Balakrishnan, 2015).

In this section, we present a practical way of measuring the degree of compatibility between likelihood and approximations. Recall that the Gibbs chain from Definition 2 is based on alternate sampling from likelihood and approximation as depicted in Figure 3.2. A sequence of this chain has the form $(\theta_1, y_1, \theta_2, y_1, \dots)$ with $\theta_t \in \Theta$ and $y_t \in \mathcal{Y}$. We then defined the Gibbs prior π_G as the limiting distribution of the θ_t, but

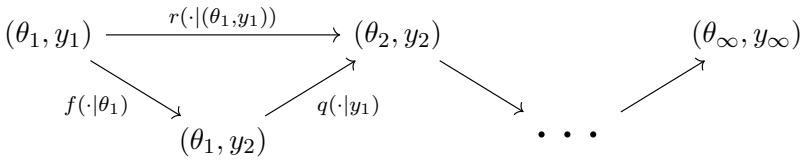


Figure 3.4: Schematic diagram of samples from the Gibbs chain from Definition 12, where one step first updates y with f and then θ with q. The distribution of θ_t converges to the Gibbs prior π_G and the distribution of y_t converges to p_G.

we can analogously consider the limiting distribution of the $\mathcal{Y}$-components y_t. To this end, we generalize Definition 2 by looking viewing the Gibbs chain as a Markov chain on $\Theta \times \mathcal{Y}$:

Definition 12 (Gibbs chain (extension to Definition 2)). For two families of distributions $(f(\cdot|\theta))_{\theta\in\Theta}$ on $\mathcal{Y}$ and $(q(\cdot|y))_{y\in\mathcal{Y}}$ on Θ consider the discrete-time Markov chain on $\Theta \times \mathcal{Y}$ whose transition function is given by

$$r((\theta', y')|(\theta, y)) = f(y'|\theta)q(\theta'|y')\,.$$

This chain is called the *Gibbs chain*. The projection onto the Θ-components is a Markov chain on Θ, any stationary distribution of which is called a *Gibbs prior* and denoted by π_G. The projection onto the $\mathcal{Y}$-components is a Markov chain on $\mathcal{Y}$, any stationary distribution of which is and denoted by p_G.

Kuo and Wang (2019) also studied this Gibbs chain for discrete distributions. We specified a joint distribution on $\Theta\times\mathcal{Y}$ with the Gibbs prior π_G as the Θ-marginal and the likelihood F as the corresponding conditional. The main observation for measuring the degree of compatibility between F and Q is that we can also specify a joint distribution from the other direction, that is, with p_G as the $\mathcal{Y}$-marginal and Q as the corresponding conditional. We abbreviate those two joint distributions with $\pi_G F$ and $p_G Q$. They coincide if and only if F and Q are compatible. It is therefore natural to measure the degree of compatibility via some divergence between them.

A practical algorithm We can obtain (correlated) samples from the two joint distributions $\pi_G F$ and $p_G Q$ with the same Gibbs chain used for obtaining samples from the Gibbs prior π_G. Simulating the Gibbs chain is described by Figure 3.4. It yields a sequence $(\theta_1, y_1, \theta_2, y_1, \dots)$, where the marginal distributions of θ_t and y_t converge to π_G and p_G, respectively. Since the components are updated alternatingly with the conditional distributions, we can pair the entries to obtain samples from the joint distributions. However, the order of the pairing is important:

- $(\theta_t, y_t) \sim \pi_G F$, because $\theta_t \sim \pi_G$ (for large t) and $y_t \sim f(\cdot|\theta_t)$
- $(\theta_{t+1}, y_t) \sim p_G Q$, because $y_t \sim p_G$ (for large t) and $\theta_{t+1} \sim q(\cdot|y_t)$

This leads to basically the same algorithm as Algorithm 2, except that the auxiliary variables y_t are stored as well and paired accordingly with no computational overhead:

Algorithm 3 Simulating the Gibbs chain (samples from the joint distributions $\pi_G F$ and $p_G Q$)

input Likelihood f, approximate inference method q, number of steps T
output Correlated samples $(\theta_t, y_t)_{t=0}^{T-1}$ from π_G and $(\theta_{t+1}, y_t)_{t=0}^{T-1}$ from p_G

$\theta_0 \leftarrow$ Arbitrary initialization, e. g. sample from $\pi(\cdot)$
for $t \leftarrow 0$ **to** $T-1$ **do**
  $y_t \leftarrow$ Randomly sample from $f(\cdot|\theta_t)$
  $q(\cdot|y_t) \leftarrow$ Approximation to $p(\cdot|y_t)$
  $\theta_{t+1} \leftarrow$ Randomly sample from $q(\cdot|y_t)$
end for

Example 13 (Gaussian conditional distributions). We demonstrate Algorithm 3 for two pairs of Gaussian conditional distributions with $\Theta = \mathcal{Y} = \mathbb{R}$, one compatible and one incompatible. The first example is taken from Arnold et al. (2001) and given by the conditional distributions

$$f(y|\theta) = \mathcal{N}\left(y \middle| \frac{4}{1+\theta^2}, \frac{1}{1+\theta^2}\right) \quad \text{and} \quad q(\theta|y) = \mathcal{N}\left(\theta \middle| \frac{4}{1+y^2}, \frac{1}{1+y^2}\right).$$

These conditionals are compatible with the bivariate joint density

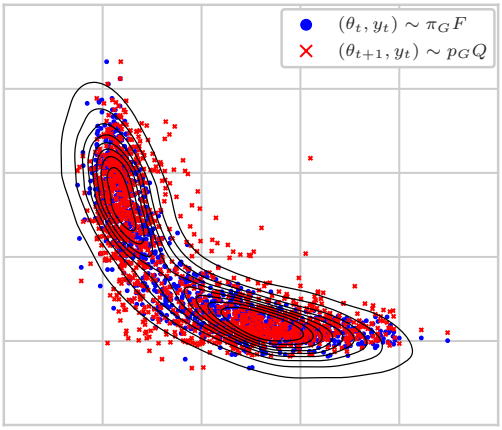
$$p(\theta, y) = \exp\left((1,\ \theta,\ \theta^2)\begin{pmatrix} c & 4 & -1/2 \\ 4 & 0 & 0 \\ -1/2 & 0 & -1/2 \end{pmatrix}\begin{pmatrix} 1 \\ y \\ y^2 \end{pmatrix}\right),$$

where $c \in \mathbb{R}$ plays the role of the normalizing constant. The corresponding samples from Algorithm 3 are shown in Figure 3.5a together with a contour plot of the true joint density $p(\theta, y)$. All three joint densities $\pi_G F$, $p_G Q$, and p overlap, confirming the compatibility of these conditional distributions.

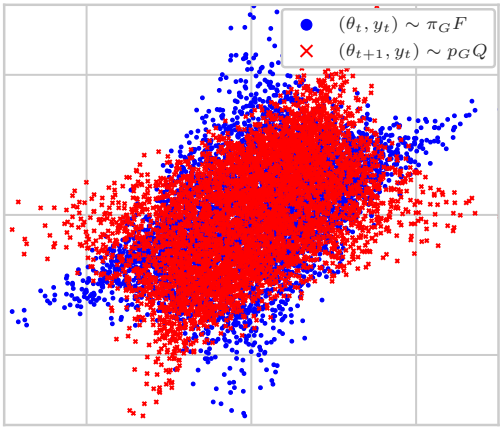
The other example is given by the conditional densities

$$f(y|\theta) = \mathcal{N}\left(y \middle| \frac{\theta}{2}, \frac{1}{1+\theta^2}\right) \quad \text{and} \quad q(\theta|y) = \mathcal{N}\left(\theta \middle| \frac{y}{2}, \frac{1}{1+y^2}\right).$$

These conditional distributions are incompatible (Arnold et al. (2001) gives a full characterization of compatible Gaussian conditional distributions). Therefore the joint distributions $\pi_G F$ and $p_G Q$ cannot coincide exactly. This is confirmed by Figure 3.5b, which shows samples from the two joint distributions. Based on these samples we could now measure some kind of divergence between the two distributions to assess the degree of compatibility.



(a) Correlated samples from $\pi_G F$ and $p_G Q$ for two compatible conditionals with the underlying joint distribution as contours. The conditionals being compatible is equivalent to all three joint distributions coinciding.



(b) Correlated samples from $\pi_G F$ and $p_G Q$ for two incompatible conditionals. The conditionals being incompatible is equivalent to those two joint distributions being different.

Figure 3.5: Samples from Algorithm 3 for the Gaussian conditional distributions of Example 13.

3.7 Experiments

We experiment with the Gibbs prior as a diagnostic tool for various approximations in two Bayesian models. For more details and convergence monitoring of the Gibbs chains see Section 3.7.3.

Baseline We compare our findings to the diagnostic Talts et al. (2018). This diagnostic is based on the stationarity equation of the prior π under the Gibbs chain, but only considers 1-step transitions with some test statistics $f\colon \Theta \to \mathbb{R}$. Under random samples $\tilde{\theta} \sim \pi$, $\tilde{y} \sim f(\cdot|\tilde{\theta})$, and $\theta_1, \dots \theta_L \sim q(\cdot|\tilde{y})$, the rank of $f(\tilde{\theta})$ in $\{f(\theta_1), \dots, f(\theta_L)\}$ is computed. This is repeated over multiple draws of $(\tilde{\theta}, \tilde{y})$, which gives a histogram of the ranks. Since the histogram is uniform under the exact posterior, any deviations from uniformity indicate an approximation mismatch. We allocate this method the same computational resources in terms of posterior draws as our Gibbs chain.

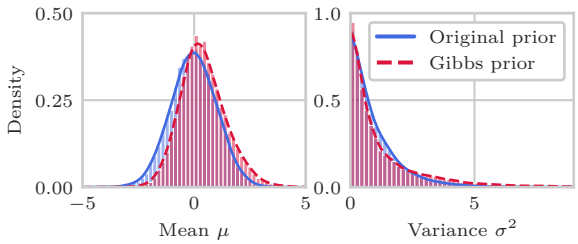


Figure 3.6: Marginal distributions of prior and Gibbs prior for the sum of log-normals model. A comparison shows that the approximation overestimates μ and puts more mass on extreme values for σ^2.

3.7.1 Sum of log-normals

Setup Our first model describes the sum of $L = 10$ independent samples from a log-normal distribution and is given by

$$\mu \sim \mathcal{N}(0, 1), \quad \sigma^2 \sim \text{Gamma}(1, 1)\,,$$
$$x_l|\theta = (\mu, \sigma^2) \overset{\text{indep.}}{\sim} \text{LogNormal}(\mu, \sigma^2), \quad y = \sum_{l=1}^{L} x_l\,.$$

Since the corresponding likelihood is infeasible we approximate the posterior in a two-step procedure: first, we replace the likelihood by its Fenton-Wilkinson approximation (Fenton, 1960), which is another log-normal distribution with matching first two moments, and then we use a Laplace approximation to the posterior of this new model.

Bias discovery To discover the bias of this approximation we simulate the Gibbs prior based on 10,000 iterations of Algorithm 2 and show it alongside the original prior in Figure 3.6. The first observation is that the Gibbs prior does not coincide with the original prior, which implies that the approximation is not exact. Furthermore, the deviation between the two distributions is systematic. For the mean μ, the Gibbs prior has a similar shape as the original prior, but is shifted to the right. This implies that the approximations systematically overestimate μ. For the variance σ^2, the Gibbs prior puts more mass on extreme values, which means that there is no systematic under- or overestimation. Compare these findings to Rodrigues et al. (2018) who consider a fixed approximation to an observation y drawn from $\theta = (0, 1)$. They confirm that μ is overestimated, but also find that σ^2 is underestimated. This does not contradict our findings, because they analyze the approximation to a *fixed* observation, while we analyze the approximations *across* observations. The other baseline Talts et al. (2018) is shown in the first two histograms of Figure 3.8 for the coordinates of $\theta = (\mu, \sigma^2)$ as summary statistics, that is, $f_i(\theta) = \theta_i$. The histogram for μ exceeds the confidence region at the smallest rank, which also suggests overestimation. For σ^2, the deviation from uniformity is not strong enough to deduce a systematic approximation mismatch.

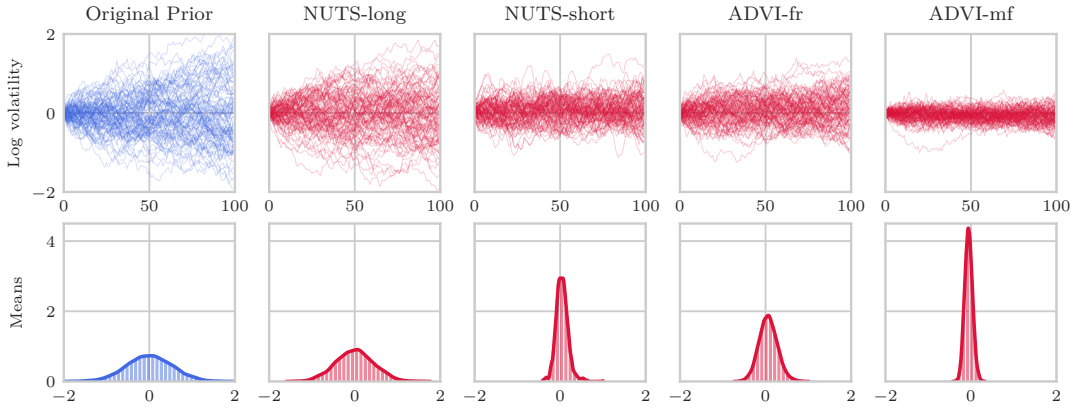


Figure 3.7: **Top row:** Samples of $\theta \in \mathbb{R}^{100}$ from original prior (blue) and Gibbs priors (red) under various approximations. **Bottom row:** Histograms of the summary statistic $\theta \mapsto 1/100 \sum_{i=1}^{100} \theta_i$, which is the mean value of a time series. Methods that are closer to the prior introduce less bias.

3.7.2 Stochastic volatility

Setup Stochastic volatility models are used in mathematical finance for time series to describe the latent variation of trading price (called the returns). We consider a model similar to Hoffman and Gelman (2014):

$$\theta_i|\theta_{i-1} \sim \mathcal{N}(\theta_i, \sigma^2), \quad i = 1, \ldots, T\,,$$
$$y_i \overset{\text{indep.}}{\sim} \text{StudentT}(\nu, 0, \exp\theta_i), \quad i = 1, \ldots, T\,,$$

where $\theta_0 = 0, \sigma = .09, \nu = 12$, and $T = 100$. The latent parameters $\theta = (\theta_1, \ldots, \theta_T)$ follow a Gaussian random walk and describe the log volatility of the returns $y = (y_1, \ldots, y_T)$, which are independent given θ. As posterior inference methods, we investigate the Hamiltonian Monte Carlo method NUTS (Hoffman and Gelman, 2014) with different number of steps (10 for NUTS-short and 40 for NUTS-long) and the variational inference method ADVI (Kucukelbir et al., 2017), which comes in a less powerful mean-field (ADVI-mf) and more powerful full-rank (ADVI-fr) variant.

Bias discovery For each approximation method, we can again use the corresponding Gibbs prior in two ways: we test *whether* it deviates from the original prior to assess exactness of the approximation, and if it does, we inspect *how* it deviates to assess the systematic bias. Figure 3.7 shows samples from original prior and Gibbs priors under the approximations alongside the distribution of means for each time series as a summary statistic. Each Gibbs chain was simulated for 10,000 steps, which took 13 hours for ADVI-fr and roughly 5 hours for the other methods on a GPU. We observe that the Gibbs prior for the long MCMC chain is almost identical to the prior, which

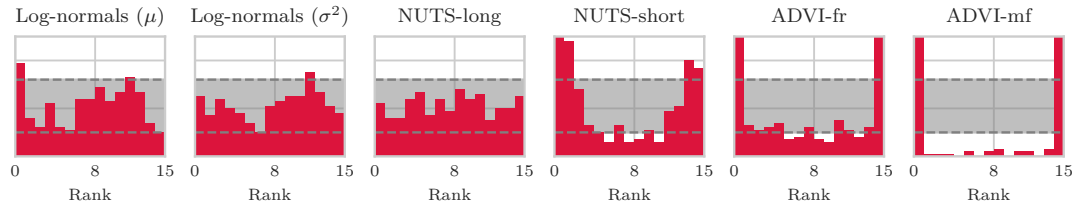


Figure 3.8: Histograms of rank statistics for the baseline Talts et al. (2018). First two histograms are for Section 3.7.1 with coordinates as summary statistics, other histograms are for Section 3.7.2 with the mean. Gray band shows a 99% confidence interval under the exact posterior. Deviations from uniformity indicate approximation mismatch.

confirms that this method is accurate; the Gibbs prior for the corresponding short chain is further away from the prior and closer to the initialization of the chain because it has not fully converged. The method ADVI-mf shows a strong deviation from the prior by concentrating on less extreme values of the latent variables. This indicates that the approximation is overly compact compared to the true posterior. The same phenomenon was already observed for mean field variational inference in Section 3.3. It can also be observed for ADVI-fr, but is less pronounced because the method is strictly more powerful. The baseline Talts et al. (2018) is shown in the last four histograms of Figure 3.8 for the same summary statistic as in Figure 3.7, the mean value of θ. For NUTS-long, the histogram stays within the confidence region, which confirms that this method is accurate. The other three methods show a $\cup$-shape, which is most pronounced for ADVI-mf. This indicates that the methods are overly compact and is in line with our findings. While this baseline can in principle also discover systematic approximation mismatches in terms of over-/underestimation and compactness, the Gibbs prior provides a more complete and nuanced picture.

3.7.3 Experimental details

In this section, we give more details on the Bayesian models and approximations to their posteriors which are considered in Section 3.7. We used the python library numpyro (Phan et al., 2019) for the posterior approximation methods Laplace, NUTS, and ADVI.

Baseline Talts et al. (2018) We allocate this baseline the same resources as the corresponding Gibbs chain in terms of draws from the posterior. That is, if our Gibbs chain runs for M steps, the baseline repeats N draws $(\tilde{\theta}, \tilde{y})$ with $\theta_1, \ldots, \theta_L \sim q(\cdot|\tilde{y})$ such that $N \cdot L \approx M$. Specifically, we choose $N = 323$ and $L = 31$. For the histograms in Figure 3.8, we re-binned once to reduce noise.

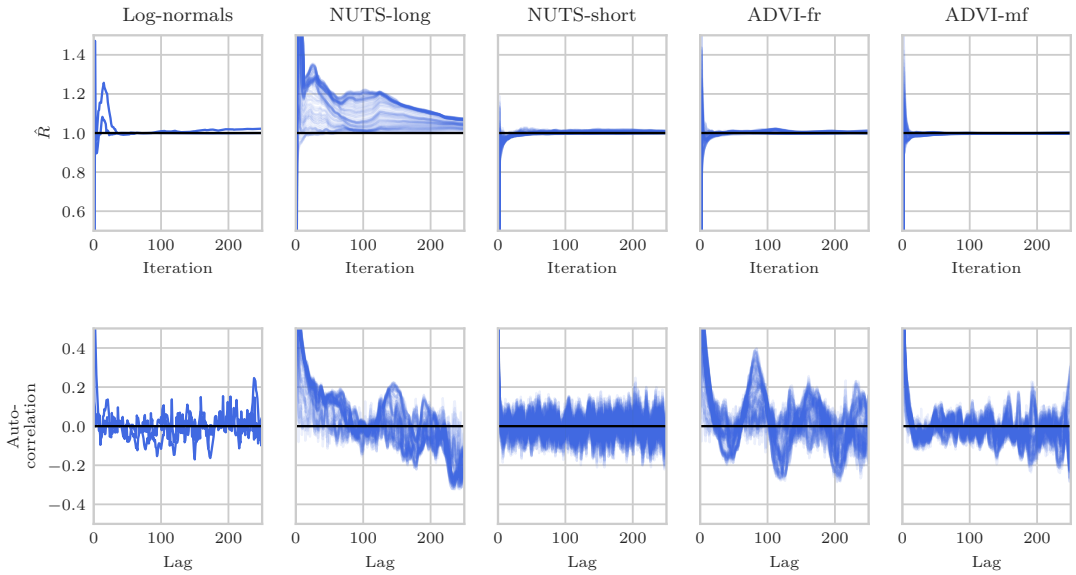


Figure 3.9: Gelman-Rubin diagnostic $\hat{R}$ (**top row**) and lag-k autocorrelation (**bottom row**) for the Gibbs chains from Section 3.7 with one curve per dimension. Section 3.7.1 (**first column**) has $d = 2$ dimensions and Section 3.7.2 (**other columns**) have $d = 100$ dimensions. Values $\hat{R} \approx 1$ or lag-k autocorrelation ≈ 0 indicate convergence of the Gibbs chain.

Convergence monitoring We monitor the convergence of our Gibbs chains with two standard measures, the Gelman-Rubin diagnostic $\hat{R}$ (Gelman and Rubin, 1992) and the lag-k autocorrelation. Both are shown in Figure 3.9 for all experiments from Section 3.7. The Gelman-Rubin diagnostic $\hat{R}$ uses multiple chains to compute the ratio of between-chain variance to within-chain variance. A ratio $\hat{R} \approx 1$ indicates convergence. The top row of Figure 3.9 shows that this value is reached quickly in all cases except for NUTS-long, which takes longer to converge. Potential explanations are that convergence is generally slower in the high-dimensional setting ($d = 100$ for NUTS-long compared to $d = 2$ for Log-normals) and that the other less accurate methods introduce additional bias that promotes faster convergence. The lag-k autocorrelation is defined as the correlation of a sequence with its shifted version by k steps. A high autocorrelation of a Markov chain indicates slow mixing and thus slower convergence. The bottom row of Figure 3.9 shows the autocorrelation for the Gibbs chains, which eventually oscillate around 0 due to finite sample noise. The autocorrelation gets close to 0 quickly for the low-dimensional setting log-normals and for the less accurate methods NUTS-short and ADVI-mf in the high-dimensional setting. Only the more accurate methods in the high-dimensional setting NUTS-long and ADVI-fr take longer to reach 0. This hints towards slower convergence and is in line with Talts et al. (2018), who predict slow convergence of the Gibbs chain when the parameters are strongly correlated to

the observations. In particular, we expect this to be the case for a large number of observations.

3.7.4 Sum of log-normals

The Bayesian model has latent parameters $\theta = (\mu, \sigma^2) \in \mathbb{R} \times \mathbb{R}_{>0}$, on which we place a prior $\pi(\theta)$ with independent marginal distributions $\mu \sim \mathcal{N}(0, 1)$ and $\sigma^2 \sim \text{Gamma}(1, 1)$. The likelihood $f(y|\theta)$ for an observation $y > 0$ is given by an L-fold convolution of a log-normal distribution, that is, $y|\theta \sim \text{LogNormal}^{*L}(\mu, \sigma^2)$.

To obtain the approximation $q(\theta|y)$ to the true posterior $p(\theta|y)$ of this model, we employ the following two-step procedure:

1. Define an approximate likelihood $\tilde{f}(y|\theta)$ as the Fenton-Wilkinson approximation to the true likelihood f, which is another log-normal distribution with matching first two moments. Specifically, $\tilde{f}(\cdot|\theta)$ describes the distribution $\text{LogNormal}(\alpha, \beta^2)$, where

$$\begin{aligned} \alpha &= \mu + \log L + 0.5\left(\sigma^2 - \beta^2\right) , \\ \beta^2 &= \log\left[\frac{\exp\sigma^2 - 1}{L} + 1\right] . \end{aligned}$$

2. Define $q(\theta|y)$ as the Laplace-approximation to the posterior of this new model $\tilde{p}(\theta|y) \propto_\theta \pi(\theta)\tilde{f}(y|\theta)$. This means that $q(\cdot|y)$ describes a bivariate normal distribution $\mathcal{N}(\theta_y^*, \Sigma_y)$ with

$$\begin{aligned} \theta_y^* &= \arg\max_\theta \pi(\theta)\tilde{f}(y|\theta) \\ \Sigma_y &= -H_{\log\tilde{p}}^{-1} , \end{aligned}$$

   where $H_{\log\tilde{p}}$ describes the Hessian matrix of $\theta \mapsto \log\Big(\pi(\theta)\tilde{f}(y|\theta)\Big)$.

3.7.5 Stochastic volatility

This model is a simplified model of the one described in Hoffman and Gelman (2014), who place additional prior distributions on the parameters σ, ν, and θ_0. We made the simplifying choice $\theta_0 = 0$. The other hyperparameters $\sigma = .09$ and $\nu = 12$ were chosen by taking the posterior means under S&P500 dataset. The posterior was approximated with NUTS where the priors were $\sigma \sim \text{Exp}(50)$ and $\nu \sim \text{Exp}(0.1)$, following Hoffman and Gelman (2014).

Measuring compactness and divergence Table 3.3 supplements our statements about the bias of the approximation methods in Section 3.7.2. Regarding compactness, we can confirm that the methods NUTS-short, ADVI-fr, and ADVI-mf are overly compact compared to the original prior. Regarding divergence, we see that the method NUTS-long is closest to the original prior and the restrictive method ADVI-mf is farthest. The more powerful versions yield Gibbs priors that are closer to the original prior, that is, NUTS-long is closer than NUTS-short and ADVI-fr is closer than ADVI-mf.

Table 3.3: Compactness and distance to original prior for the approximation methods of Section 3.7.2. Compactness is measured by the Frobenius norm of the empirical covariance matrix and distance to the original prior is measured by the maximum mean discrepancy under the Gaussian kernel $k(x, y) = \exp(-\|x - y\|^2/2)$.

	Original prior	NUTS-long	NUTS-short	ADVI-fr	ADVI-mf
Compactness	34.18	23.81	3.37	6.09	1.21
Distance to original prior	0	0.014	0.035	0.021	0.179

3.8 Conclusion and future work

Conclusion We describe a novel diagnostic approach for assessing the inductive bias of approximate Bayesian inference methods. A reformulation of this problem leads to a natural solution, which we call the Gibbs prior. We demonstrate how it can be used to discover the inductive bias in various examples.

Future work The Gibbs prior compromises between many pointwise priors. The precise nature of this compromise is intricate, offering several avenues for future analysis. While we introduced the Gibbs prior in the context of approximate Bayesian methods, it can be defined for any generative method returning a distribution over latent variables given an observation. Another direction is using the pointwise priors as observation-dependent diagnostics. They do not suffer from incompatibility, but can be more challenging to sample from if the approximation density is unknown.

Broader impact Recently, there has been a surge of interest in interpretable and explainable machine learning algorithms. One principled way of explaining an algorithm is to inspect its inductive bias, which describes the preferred solutions independent of the data. While the inductive bias is specified only implicitly for most algorithms, it is made explicit in Bayesian inference through prior and likelihood. Unfortunately, this transparency is concealed for approximate Bayesian inference, because approximations

introduce additional hidden bias. We present a method to uncover this inductive bias again, which opens up a new paradigm for the practical evaluation of approximate inference.

Chapter 4

A consistent estimator for confounding strength

A common machine learning task is to learn the influence of features x on a target variable y from a set of observations $\{(x_i, y_i)\}_{i=1}^n$. In many applications, we are not only interested in the statistical problem of predicting y after *observing* x; instead, we ask the causal question of how y changes after *intervening* on x. Unfortunately, the causal dependence structure between x and y is in general not identifiable from their statistical dependencies (Pearl, 2009b). Simply regressing y on x attributes all dependencies to direct causal influence and is therefore only appropriate when x causes y without hidden confounders. However, this solution can be grossly misleading in the other possible cases where y causes x or both are caused by a common confounder (Reichenbach, 1956).

For example, assume we want to predict how increasing a person's education x affects their income y. It could be that a higher education is a requirement for well-paying jobs (education causes income), in which case increasing the education directly increases the income. However, even if we rule out the possibility that income causes education, education and income could both be affected by some hidden confounders such as the socioeconomic status of the parents. A priori, it is unclear to what extent the observed statistical dependence between x and y is due to direct causal influence or due to such confounding factors.

This fundamental non-identifiability issue of causal from observational structure can be addressed in different ways. One way is access to additional data such as data from different environments (Heinze-Deml et al., 2018; Peters et al., 2016) or instrumental variables (Bowden and Turkington, 1990; Imbens and Angrist, 1994), which reduces the causal learning problem to a statistical one. Alternatively, one can assume that the underlying causal model follows a certain data-generating process such as additive noise models (Hoyer et al., 2008a; Kano et al., 2003; Zhang and Hyvärinen, 2009). This reduces the number of causal models which can explain a given observational structure

and therefore mitigates the non-identifiability. A more abstract approach to choose a causal model among those compatible with an observational structure is to postulate certain information-theoretic properties of the causal model. For example, the causal directions are those that maximize conditional entropies or the causal factorization of the joint distribution is the one with minimal Kolmogorov complexity (Bloebaum et al., 2018; Janzing and Schölkopf, 2010; Marx and Vreeken, 2019; Sun et al., 2006).

In this work, we theoretically analyze the confounding strength estimator by Janzing and Schölkopf (2018). This estimator assumes that x causes y and aims to estimate the strength of unobserved confounding based on observational data $\{(x_i, y_i)\}_{i=1}^n$. Here, the confounding strength is defined as the discrepancy between the causal effect of x on y and the statistical regression vector. To mitigate the non-identifiability, the estimator considers a linear Gaussian causal model under the assumption of independent causal mechanisms, a common assumption in causal learning (Janzing and Schölkopf, 2010; Lemeire and Janzing, 2013; Peters et al., 2017). Abstractly, this principle states that the different causal mechanisms share no information. While the task of confounding strength estimation remains ill-posed in finite dimensions, it becomes solvable in the high-dimensional limit due to concentration of measure phenomena. Crucially, this approach therefore requires large dimension d to reduce the non-identifiability error, but at the same time requires an even larger number of samples $n \gg d$ to reduce the finite-sample error. This is because it uses the empirical covariance matrix and regression vector in an intermediate step to estimate the corresponding population quantities, which is only consistent for $n \gg d$. It is therefore not guaranteed that this estimator is consistent in the high-dimensional regime. We address this issue by analyzing this estimator, from here on referred to as the plug-in estimator, in the proportional asymptotic regime $n, d \to \infty$ with $d/n \to \gamma \in [0, 1)$ and make the following contributions:

- We derive the asymptotic behavior of the plug-in estimator for confounding strength from Janzing and Schölkopf (2018) in the proportional asymptotic regime and show that it is not generally consistent. We also show that the approach based on population instead of finite-sample quantities is consistent.
- We derive a consistent estimator for confounding strength by correcting the above estimator with tools from random matrix theory.
- We demonstrate the improvement experimentally on finite-dimensional data from our causal model.

This work is structured as follows. Section 4.1 gives an overview of related work on causal inference under unobserved confounding. Section 4.2 introduces the confounded causal model, the measure of confounding strength, and basic notions from random matrix theory which are needed for the analysis. Section 4.3 describes the general approach of Janzing and Schölkopf (2018) and shows that it is consistent based on population quantities in Section 4.3.1, but generally biased based on plug-in quantities

in Section 4.3.2. A corrected, consistent estimator for confounding strength is then derived in Section 4.4. Section 4.8 concludes with a discussion.

4.1 Related work

Learning causal relationships under the presence of unobserved confounding has been investigated by multiple works. Hoyer et al. (2008b) detect the causal direction in linear non-Gaussian models based on the structure of the mixing matrix and Janzing et al. (2009) do so for non-linear additive noise models. Janzing et al. (2011) detect low-complexity confounding based on a purity criterion for conditional distributions. Kaltenpoth and Vreeken (2019) decide whether a causal model is confounded based on the algorithmic Markov condition. Chen et al. (2022) consider the stability of the regression vectors under different environments as an indication for causal influence.

Our work falls into another line of work that detects confounding based on the assumption of independent causal mechanisms. This assumption induces certain non-generic alignments between the coefficients of the observational distribution, which can be used to identify confounding. Bellot and van der Schaar (2021) use this assumption to learn a sparse causal DAG under dense confounding. Janzing and Schölkopf (2017) introduce the notion of confounding strength and estimate it under scalar confounding. Their method is based on the observation that a weighted spectral measure of the covariance matrix concentrates in high dimensions. Liu and Chan (2018) build on this idea by moving from the spectral measure to its first moment. Janzing and Schölkopf (2018) extend this setting to multivariate confounding, which is the setting of our work. Janzing (2019) considers a subsequent task of learning a causal model with ridge regression. It uses an estimate of confounding strength to choose an appropriate regularization parameter, which is motivated by an analogy between finite sample error and confounding. Vankadara et al. (2022) generalize the notion of confounding strength beyond independent causal mechanisms and characterize the relationship between confounding strength and the causal risk of ridge regression in the high-dimensional limit.

Another related field is sensitivity analysis for treatment-effect studies based on observational data. Sensitivity analysis aims to quantify how sensitive causal conclusions are to potential unobserved confounding (Cornfield et al., 2009). Since this task suffers from the same non-identifiability issue as described above, early work relies on assumptions about the unobserved confounder (Flanders and Khoury, 1990; VanderWeele and Arah, 2011). A more recent, popular approach without assumptions gives bounds based on two (unknown) sensitivity parameters for how strong confounding would need to be in order to explain away any observed statistical associations between treatment and effect (Ding and VanderWeele, 2016; Peña, 2022; Sjölander, 2020). The region of sensitivity parameters that explain away associations can be condensed into a single E-value, which acts as a measure of confounding strength and can be computed from observational data (VanderWeele and Ding, 2017; VanderWeele et al., 2019).

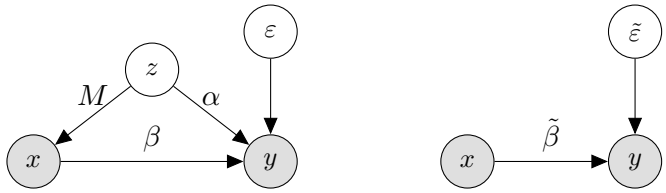


Figure 4.1: **Left:** DAG corresponding to the causal model (4.1). **Right:** corresponding observational model as in Proposition 14 with $\tilde{\varepsilon} \sim \mathcal{N}(0, \tilde{\sigma}^2)$. Unobserved variables are dashed.

4.2 Preliminaries

This preliminary section introduces our confounded causal and a notion of confounding strength in Section 4.2.1, as well as basic tools from random matrix theory needed for analysis in Section 4.2.2.

4.2.1 The confounded causal model

We first describe the problem setup and introduce basic quantities. We consider a confounded causal model with linear conditionals and Gaussian distributions. Specifically, we define the causal model in terms of its structural equations

$$\begin{aligned} z &\sim \mathcal{N}(0, I_l)\,, \\ \varepsilon &\sim \mathcal{N}(0, \sigma^2)\,, \\ x &= Mz\,, \\ y &= x^T\beta + z^T\alpha + \varepsilon\,. \end{aligned} \tag{4.1}$$

Figure 4.1 shows the corresponding directed acyclic graph (DAG). The model depends on a set of hyperparameters $\alpha \in \mathbb{R}^l, \beta \in \mathbb{R}^d, M \in \mathbb{R}^{d\times l}$ with $l \geq d$ and the noise $\sigma^2 \geq 0$. All variables x, y, z have mean 0 and the covariance of the features is given by $\Sigma \coloneqq \mathrm{Cov}(x) = MM^T \in \mathbb{R}^{d\times d}$. We additionally assume that M has full rank d such that Σ is invertible. We use the notation $\|x\|_\Sigma^2 \coloneqq x^T\Sigma x$ for the generalized norm, M^+ for the pseudo-inverse of M, and $M^{+T} \coloneqq (M^+)^T$ as shorthand.

By construction, β describes the causal influence of x on y. This is formally captured by the interventional distribution of the *do*-calculus (Pearl, 2009a) under which $y = x_0^T\beta + z^T\alpha + \varepsilon$ is only a random variable in z, ε and therefore $\mathbb{E}_{y|do(x=x_0)}y = x_0^T\beta$. However, we do not assume access to interventional data; instead, we only observe values values (x, y). The corresponding statistical dependencies between x and y are captured by the usual conditional distribution:

Lemma 14 (Observational distribution). *For the causal model (4.1), the observational distribution of y given x is $y|x \sim \mathcal{N}(x^T\tilde{\beta}, \tilde{\sigma}^2)$, where $\tilde{\beta} = \beta + M^{+T}\alpha$ and $\tilde{\sigma}^2 = \sigma^2 + \|\alpha\|_{I_l - M^+M}^2$.*

Proof. Since $z \sim \mathcal{N}(0, I_l)$ is Gaussian and $x = Mz$ is a linear map, it is a standard result that $z^T|x$ is Gaussian again with parameters $z^T|x \sim \mathcal{N}(x^T M^{+T}, I - M^+ M)$. Subsequently, we have $z^T\alpha|x \sim \mathcal{N}(x^T M^{+T}\alpha, \|\alpha\|^2_{I-M^+M})$. With $y = x^T\beta + z^T\alpha + \varepsilon$, we arrive at

$$y|x \sim \mathcal{N}(x^T(\beta + M^{+T}\alpha), \sigma^2 + \|\alpha\|^2_{I-M^+M}) = \mathcal{N}(x^T\tilde{\beta}, \tilde{\sigma}^2)\,.$$

□

The statistical parameter $\tilde{\beta}$ can also be viewed as the result of regressing y on x on the population level, that is, $\tilde{\beta} = \mathrm{Cov}(x)^+ \mathrm{Cov}(x, y)$. Notice that $\tilde{\beta}$ is equal to the causal parameter β up to an error term $M^{+T}\alpha$, which results from the influence of the confounder z on y. This error term cannot be identified even if we have access to the full joint distribution $\mathbb{P}_{(x,y)}$, which demonstrates the fundamental non-identifiability issue of causal learning. To quantify the error of incorrectly treating $\tilde{\beta}$ as the causal parameter, Janzing and Schölkopf (2017) propose the following measure of confounding strength:

Definition 15 (Measure of confounding strength, (Janzing and Schölkopf, 2017)). The *confounding strength* ζ for the causal model (4.1) is defined as the relative error between statistical parameter $\tilde{\beta}$ and causal parameter β via

$$\zeta := \frac{\|\tilde{\beta} - \beta\|^2}{\|\beta\|^2 + \|\tilde{\beta} - \beta\|^2}\,. \tag{4.2}$$

The confounding strength ζ takes values in $[0, 1]$, where $\zeta = 0$ describes the unconfounded case $\alpha = 0$ for which $\tilde{\beta} = \beta$ and $\zeta = 1$ describes the purely confounded case $\beta = 0$. A larger confounding strength implies that the statistical parameter is further away from the causal parameter.

The goal of this work is to estimate the confounding strength based on finite samples $\{(x_i, y_i)\}_{i=1}^n \subset \mathbb{R}^d \times \mathbb{R}$ from the observational distribution $\mathbb{P}_{(x,y)}$, which we compactly write as $X \in \mathbb{R}^{d\times n}$ and $Y \in \mathbb{R}^n$. We define two quantities which are central to the following estimators, namely the sample covariance matrix $\hat{\Sigma} := \frac{1}{n}XX^T$ and the result of regressing Y on X, $\hat{\beta} := (\frac{1}{n}XX^T)^+\frac{1}{n}XY$.

4.2.2 Basic tools from random matrix theory

We briefly recap some standard tools and results from random matrix theory to analyze the following estimators for confounding strength in the high-dimensional regime. The analysis is based on the following two objects, which capture the spectrum of a matrix:

Definition 16 (Empirical spectral distribution and Stieltjes transform). Let $\Sigma \in \mathbb{R}^{d\times d}$ be a symmetric matrix with eigenvalues $\lambda_1, \ldots, \lambda_d$. The *empirical spectral*

distribution of Σ is defined as the normalized counting measure of its eigenvalues $\mu_\Sigma := \frac{1}{d}\sum_{i=1}^{d}\delta_{\lambda_i}$. The corresponding *Stieltjes transform* of this measure is defined as the function $m_\Sigma(z) := \sum_{i=1}^{d}\frac{1}{\lambda_i - z}$ for $z \in \mathbb{C} \setminus \{\lambda_1, \ldots, \lambda_d\}$.

We need to characterize the spectral behavior of the empirical covariance matrix $\hat{\Sigma} = \frac{1}{n}XX^T \in \mathbb{R}^{d\times d}$ and the closely related empirical kernel matrix $\hat{K} = \frac{1}{n}X^TX \in \mathbb{R}^{n\times n}$. The following standard result relates their limiting spectra to the spectrum of the population covariance in terms of Stieltjes transforms:

Theorem 17 (Asymptotics of the sample covariance matrix, (Silverstein and Bai, 1995)). *Let $n, d \to \infty$ such that $d/n \to \gamma \in (0, \infty)$ and assume that the empirical spectral distribution of the covariance Σ converges, that is, $\mu_\Sigma \xrightarrow{a.s.} \nu$ with corresponding Stieltjes transform m_ν. Then it holds that $\mu_{\hat{\Sigma}} \xrightarrow{a.s.} \mu$ and $\mu_{\hat{K}} \xrightarrow{a.s.} \tilde{\mu}$ as $d \to \infty$, where $\mu, \tilde{\mu}$ are the unique measures having Stieltjes transforms $m(z)$ and $\tilde{m}(z)$, respectively. For $z \in \mathbb{C} \setminus \mathbb{R}_+$, they satisfy*

$$m(z) = \frac{1}{\gamma}\tilde{m}(z) + \frac{1-\gamma}{\gamma z}\,, \tag{4.3}$$

$$m_\nu\left(-\frac{1}{\tilde{m}(z)}\right) = -zm(z)\tilde{m}(z)\,. \tag{4.4}$$

A corresponding version of Eq. (4.3) holds in finite dimensions and simply reflects the fact that $\hat{\Sigma}$ and $\hat{K}$ share the same eigenvalues up to the eigenvalue 0 with multiplicity $|n - d|$. Eq. (4.4) is the main result that connects the limiting Stieltjes transforms of the empirical matrices $\hat{\Sigma}$ and $\hat{K}$ to the limiting Stieltjes transform of the population covariance Σ. The solution m to this equation remains implicitly defined in all but the simplest case $\Sigma = I_d$, where m is the Stieltjes transform of a Marčenko-Pastur distribution.

4.3 Asymptotic behavior of the population and plug-in estimators for confounding strength

In this section, we describe the general approach for estimating confounding strength based on the assumption of independent causal mechanisms (Janzing and Schölkopf, 2018). We show that the estimator is consistent based on population quantities in Section 4.3.1, but is generally biased for $n \not\gg d$ based on sample (plug-in) quantities in Section 4.3.2.

The main ingredient to tackle the non-identifiability of the causal model is the assumption of independent causal mechanisms, a common assumption in causal learning (Janzing and Schölkopf, 2010). This abstract principle states that the physical mechanisms of a causal model that transfers causes to effect share no information. A possible translation for the causal model (4.1) is the assumption that the mechanisms α and

β are drawn from independent rotationally invariant distributions. Specifically, we assume that α and β are independent with $\alpha \sim \mathcal{N}(0, \sigma_\alpha^2 I_l)$ and $\beta \sim \mathcal{N}(0, \sigma_\beta^2 I_d)$ for unknown hyperparameters $\sigma_\alpha^2, \sigma_\beta^2 \geq 0$. Intuitively, this assumption facilitates estimation because it implies a certain alignment between the covariance matrix $\Sigma = MM^T$ and the regression vector $\tilde{\beta} = \beta + M^{+T}\alpha$: for large confounding α, the error term $M^{+T}\alpha$ is aligned with small singular value directions of M. Correspondingly, $\tilde{\beta}$ is aligned with small eigendirections of Σ.

Assumption 18. We make the following assumptions about the (sequence) of causal models:

(A1) The parameters α, β of model (4.1) are independently sampled with $\alpha \sim \mathcal{N}(0, \sigma_\alpha^2 I_l)$ and $\beta \sim \mathcal{N}(0, \sigma_\beta^2 I_d)$ for hyperparameters $\sigma_\alpha^2, \sigma_\beta^2 \geq 0$.

(A2) The number of samples n, data dimension d, and latent confounder dimension l are in the proportional asymptotic regime, that is, $n, d, l \to \infty$ such that $d/n \to \gamma \in (0, 1)$ and $l/d \to \tilde{\gamma} \geq 1$.

(A3) The empirical spectral distribution μ_Σ of the population covariance Σ converges almost surely as $d \to \infty$ to a distribution ν with bounded support, that is, $\operatorname{supp}(\nu) \subseteq [h_1, h_2]$ with $0 < h_1 \leq h_2 < \infty$.

Assumption (A1) is the assumption of independent causal mechanisms. Assumption (A2) captures that this approach to confounding strength estimation requires high dimensions so that concentration effects can mitigate the non-identifiability issue. We exclude the case $\gamma \geq 1$, because there estimation of the term $\frac{1}{d}\operatorname{Tr}(\Sigma^{-1})$ (which later turns out to be relevant) is hard, see Couillet and Liao (2022, Remark 2.11) for a discussion. The restriction on the latent dimensions $\tilde{\gamma} \geq 1$ ensures that $l \geq d$ so that the population covariance $\Sigma = MM^T$ with $M \in \mathbb{R}^{d \times l}$ can be full rank. This is necessary for Assumption (A3) because we require the limiting support to be bounded away from zero.

Remark 19. The assumption of independent causal mechanisms alone does not resolve the non-identifiability issue and it also does not enable estimation of the multivariate vectors α or β. However, *scalar* functions of these parameters can concentrate in high dimensions. In particular, this happens for confounding strength.

The following key lemma states that random quadratic forms can concentrate around their trace.

Lemma 20 (Quadratic-form-close-to-the-trace, (Bai and Silverstein, 2010, Lemma B.26)). *Let $x = (x^1, \ldots, x^d) \in \mathbb{R}^d$ have independent entries x^i of zero mean, unit variance and $\mathbb{E}[|x^i|^K] \leq \nu_K$ for some $K \geq 1$. Then for $A \in \mathbb{R}^{d \times d}$ and $k \geq 1$,*

$$\mathbb{E}\left[|x^T A x - \operatorname{Tr} A|^k\right] \leq C_k \left[\left(\nu_4 \operatorname{Tr}\left(AA^T\right)\right)^{k/2} + \nu_{2k} \operatorname{Tr}\left(AA^T\right)^{k/2}\right],$$

for some constant $C_k > 0$ independent of d. In particular, if the operator norm of A satisfies $\|A\| \leq 1$ and the entries of x have bounded eigth-order moment,

$$\mathbb{E}\left[\left(x^T Ax - \operatorname{Tr} A\right)^4\right] \leq Cd^2\,,$$

for some $C > 0$ independent of d, and consequently

$$\frac{1}{d}x^T Ax - \frac{1}{d}\operatorname{Tr} A \xrightarrow[d\to\infty]{a.s.} 0\,.$$

Using this lemma, we directly obtain concentration of the confounding strength.

Corollary 21 (Confounding strength concentrates). *Under Assumption 18,*

$$\zeta - \frac{\tau^{pop}\cdot\theta^*}{1+\tau^{pop}\cdot\theta^*} \xrightarrow{a.s.} 0\,, \tag{4.5}$$

where $\tau^{pop} \coloneqq \frac{1}{d}\operatorname{Tr}(\Sigma^{-1})$ and $\theta^* \coloneqq \sigma_\alpha^2/\sigma_\beta^2$.

Proof. By rewriting the confounding strength from Eq. (4.2) in terms of the hyperparameters α, β, M, we see that it consists only of quadratic forms. These can be controlled by Lemma 20, which yields

$$\zeta = \frac{\frac{1}{d}\alpha^T M^+ M^{+T}\alpha}{\frac{1}{d}\beta^T I_d \beta + \frac{1}{d}\alpha^T M^+ M^{+T}\alpha} \overset{a.s.}{\approx} \frac{\frac{1}{d}\operatorname{Tr}(M^+M^{+T})\sigma_\alpha^2}{\frac{1}{d}\operatorname{Tr}(I_d)\sigma_\beta^2 + \frac{1}{d}\operatorname{Tr}(M^+M^{+T})\sigma_\alpha^2} = \frac{\tau^{\text{pop}}\cdot\theta^*}{1+\tau^{\text{pop}}\cdot\theta^*}\,.$$

□

It only remains to estimate the trace term τ^{pop} and the ratio θ^*. In the following, we distinguish between three different kinds of estimators for various quantities: estimators based on the population quantities $\Sigma, \tilde{\beta}$, based on the plug-in quantities $\hat{\Sigma}, \hat{\beta}$, and consistent estimators derived by random matrix theory. For example, we write τ^{pop}, τ^{plg}, or τ^{RMT}.

4.3.1 The population estimator for confounding strength is consistent

First, we consider estimation based on the population quantities Σ and $\tilde{\beta}$, which basically assumes that there are no finite-sample issues. In this case, $\tau^{\text{pop}} = \frac{1}{d}\operatorname{Tr}(\Sigma^{-1})$ is known and does not need to be estimated. To estimate $\theta^* = \sigma_\alpha^2/\sigma_\beta^2$ observe that Assumption 18(A1) on α and β implies $\tilde{\beta} = \beta + M^{+T}\alpha \sim \mathcal{N}(0, \sigma_\beta^2 + \sigma_\alpha^2\Sigma^{-1})$. With respect to the uniform distribution on the sphere S^{d-1}, the distribution of the normalized vector $\tilde{\beta}/\|\tilde{\beta}\|$ has the log density $\log p_{\theta^*}(v) = -.5(\log\det(\Sigma+\theta^*) + d\log\langle v, \Sigma(\Sigma+$

$\theta^*)^{-1}v\rangle - \log\det\Sigma)$, where $v \in S^{d-1}$. Correspondingly, θ^* can then be estimated via maximum likelihood estimation as[1]

$$\theta^{\text{pop}} = \arg\min_{\theta\geq 0} f^{\text{pop}}(\theta), \quad \text{where} \quad f^{\text{pop}}(\theta) = \frac{1}{d}\log\det(\Sigma+\theta) + \log\langle\frac{\tilde{\beta}}{\|\tilde{\beta}\|}, \Sigma(\Sigma+\theta)^{-1}\frac{\tilde{\beta}}{\|\tilde{\beta}\|}\rangle\,. \tag{4.6}$$

In summary, we consider the following population estimator for confounding strength.

Definition 22 (Population estimator for confounding strength). Given Σ and $\tilde{\beta}$, the *population estimator* for confounding strength ζ^{pop} is defined as

$$\zeta^{\text{pop}} = \frac{\tau^{\text{pop}}\cdot\theta^{\text{pop}}}{1+\tau^{\text{pop}}\cdot\theta^{\text{pop}}}\,, \tag{4.7}$$

where $\tau^{\text{pop}} = \frac{1}{d}\operatorname{Tr}(\Sigma^{-1})$ and θ^{pop} is given by Eq. (4.6).

We now analyze this estimator by analyzing the asymptotic behavior of θ^{pop} from Eq. (4.6). Since θ^{pop} is implicitly defined as the minimizer of the function f^{pop}, we first derive its asymptotic behavior as an intermediate step. Specifically, we consider its derivative, which is given by

$$\partial_\theta f^{\text{pop}}(\theta) = m_\Sigma(-\theta) - \frac{\langle\frac{\tilde{\beta}}{\|\tilde{\beta}\|}, \Sigma(\Sigma+\theta)^{-2}\frac{\tilde{\beta}}{\|\tilde{\beta}\|}\rangle}{\langle\frac{\tilde{\beta}}{\|\tilde{\beta}\|}, \Sigma(\Sigma+\theta)^{-1}\frac{\tilde{\beta}}{\|\tilde{\beta}\|}\rangle}\,. \tag{4.8}$$

This idea is realized in the next theorem, which shows that the confounding strength estimator based on population quantities is consistent as $n, d \to \infty, d/n \to \gamma \in (0,1)$.

Theorem 23 (Population estimator is consistent). *Under Assumption 18 with* $\theta^* > 0$,

1. *For every* $\theta \geq 0$, *the derivative of the function from Eqs. (4.6) satisfies*

$$\partial_\theta f_d^{pop}(\theta) \xrightarrow{a.s.} (\theta-\theta^*)\operatorname{Var}_{\lambda\sim\nu}\left[\frac{1}{\lambda+\theta}\right]\mathbb{E}_{\lambda\sim\nu}\left[\frac{\lambda+\theta^*}{\lambda+\theta}\right]^{-1}, \tag{4.9}$$

2. *For some* $C > \theta^*$ *and every* $d \in \mathbb{N}$, *let* θ_d^{pop} *be a root of* $\partial_\theta f_d^{pop}$ *in* $[0, C]$ *if it exists or* 0 *otherwise. Additionally, assume that* ν *is not degenerate. Then the sequence* $\{\theta_d^{pop}\}$ *converges to* θ^* *almost surely.*

[1] Maximum likelihood estimation on the density of $\tilde{\beta}$ directly leads to the same optimality condition for θ^{pop}.

Proof. We just present a proof sketch here, the full proof is deferred to Section 4.5. For the first statement about the population function $\partial_\theta f_d^{\text{pop}}$ we treat the three terms in Eq. (4.8) separately. The first term $m_\Sigma(-\theta)$ converges to $m_\nu(-\theta)$ by Assumption 18(A3). The two quadratic forms are handled by Lemma 20 after rewriting $\tilde{\beta} = \beta + M^{+T}\alpha = \begin{pmatrix}\sigma_\alpha M^{+T} & \sigma_\beta I_d\end{pmatrix} u$ for some $u \sim \mathcal{N}(0, I_{l+d})$. Plugging everything together and simplifying yields the result.

We prove the second statement by first upgrading the convergence of Eq. (4.9) to uniform convergence on $[0, C]$ using Vitali's convergence theorem (Titchmarsh et al., 1939), and then conclude that the roots converge to the unique root θ^* of the limiting function using Hurwitz's theorem (Titchmarsh et al., 1939). □

This theorem shows that the approach of minimizing the log probability based on population quantities in Eq. (4.6) correctly estimates θ^* in the limit. Therefore, Eq. (4.7) leads to a consistent estimator for confounding strength. For the second statement, it is necessary to assume that the limiting spectral distribution ν of Σ is not degenerate, because otherwise $\text{Var}_{\lambda\sim\nu}[1/(\lambda+\theta)] = 0$. In this case, Eq. (4.9) states that the derivative $\partial_\theta f^{\text{pop}}$ converges to the constant 0 function, which contains no information about θ^*. This is perfectly in line with the intuition presented for this approach: estimation of confounding strength is made possible by an alignment of $\tilde{\beta}$ with small eigendirections of Σ, but if Σ is a multiple of the identity (or, equivalently, the distribution of eigenvalues ν is degenerate), there is no particular small eigendirection.

4.3.2 The plug-in estimator for confounding strength is generally biased

The population estimator considered above crucially relies on the population quantities Σ and $\tilde{\beta}$, which are not directly available. In practice, we only have access to the corresponding empirical quantities $\hat{\Sigma}$ and $\hat{\beta}$ based on samples X, Y. This section considers the resulting plug-in estimator for confounding strength as introduced by Janzing and Schölkopf (2018) and shows in a similar asymptotic analysis that this estimator is generally biased. Formally, the plug-in estimator follows the same structure as Definition 22, but replaces the population quantities $\Sigma, \tilde{\beta}$ with the empirical quantities $\hat{\Sigma}, \hat{\beta}$.

Definition 24 (Plug-in estimator for confounding strength, (Janzing and Schölkopf, 2018)). The *plug-in estimator* for confounding strength ζ^{plg} is defined as

$$\zeta^{\text{plg}} = \frac{\tau^{\text{plg}} \cdot \theta^{\text{plg}}}{1 + \tau^{\text{plg}} \cdot \theta^{\text{plg}}}, \tag{4.10}$$

where $\tau^{\text{plg}} = \frac{1}{d}\operatorname{Tr}(\hat{\Sigma}^{-1})$ and θ^{plg} is given by

$$\theta^{\text{plg}} = \arg\min_{\theta \geq 0} f^{\text{plg}}(\theta), \quad \text{where} \quad f^{\text{plg}}(\theta) = \frac{1}{d}\log\det(\hat{\Sigma}+\theta) + \log\langle \frac{\hat{\beta}}{\|\hat{\beta}\|}, (\hat{\Sigma}(\hat{\Sigma}+\theta)^{-1})\frac{\hat{\beta}}{\|\hat{\beta}\|}\rangle\,. \tag{4.11}$$

The main issue with the plug-in estimator in the proportional asymptotic regime is that $\hat{\Sigma}$ and $\hat{\beta}$ are not consistent estimators for Σ and $\tilde{\beta}$. Any subsequent estimators are therefore also not guaranteed to be consistent. The first example of such behavior is given by the plug-in estimator $\tau^{\text{plg}} = \frac{1}{d}\operatorname{Tr}(\hat{\Sigma}^{-1})$ for $\tau^{\text{pop}} = \frac{1}{d}\operatorname{Tr}(\Sigma^{-1})$, one of the two quantities which need to be estimated in Eq. (4.5).

Proposition 25 (Asymptotic trace of inverse covariance). *Under Assumption 18, it holds*

$$\tau^{plg} - (1-\gamma)^{-1}\tau^{pop} \xrightarrow[d\to\infty]{a.s.} 0\,.$$

Proof. In terms of Stieltjes transforms, the statement reads $(1-\gamma)m_{\hat{\Sigma}}(0) - m_{\Sigma}(0) \xrightarrow[d\to\infty]{a.s.} 0$. The limiting empirical and population Stieltjes transforms are given by $m_{\hat{\Sigma}}(z) \xrightarrow{a.s.} m(z)$ and $m_{\Sigma}(z) \xrightarrow{a.s.} m_{\nu}(z)$ as $d \to \infty$, so it remains to relate $m(0)$ to $m_{\nu}(0)$. By combining equations (4.3) and (4.4) from Theorem 17, we get

$$m_{\nu}\left(-\frac{1}{\tilde{m}(z)}\right) = (1-\gamma-zm(z))\,m(z)\,.$$

Taking $z \to 0$, it is $1/\tilde{m}(z) \to 0$ and therefore we get by continuity that $m_{\nu}(0) = (1-\gamma)m(0)$. □

This result shows that the plug-in estimator for the trace of the inverse covariance matrix is off by a factor of $(1-\gamma)$. This factor is negligible in the case $n \gg d$ where $\gamma = d/n \approx 0$, but becomes increasingly relevant as γ grows.

Next, we treat the plug-in estimator θ^{plg} similarly as θ^{pop} in Theorem 23 and show that it is generally biased. Here, $\partial_\theta f^{\text{plg}}$ is given analogously to Eq. (4.8).

Theorem 26 (Plug-in estimator is generally biased). *Under Assumption 18 with* $\theta^* > 0$,

1. *For all* $\theta \geq 0$, *the derivative of the function from Eq. (4.11) satisfies*

$$\partial_\theta f_d^{plg}(\theta) \xrightarrow{a.s.} \left[\theta - (1+\gamma\tilde{\gamma})\theta^* + \gamma\theta^*(1-\theta m(-\theta))\left(1+\frac{M(-\theta)}{M(-\theta)-m(-\theta)^2}\right)\right]h(\theta)\,, \tag{4.12}$$

with $h(\theta) = (M(-\theta) - m(-\theta)^2)(1-\theta m(-\theta) + (1-2\gamma+\gamma\tilde{\gamma})\theta^* m(-\theta) + \gamma\theta\theta^* m(-\theta)^2)^{-1}$, *where* $m(-\theta) = \mathbb{E}_{\lambda\sim\mu}[1/(\lambda+\theta)]$, *and* $M(-\theta) = \mathbb{E}_{\lambda\sim\mu}[1/(\lambda+\theta)^2]$.

2. *For every $d \in \mathbb{N}$, let θ_d^{plg} be a root of $\partial_\theta f_d^{plg}$ if it exists or 0 otherwise. Additionally, assume that $\tilde{\gamma}$ does not satisfy*

$$\tilde{\gamma} = (1 - \theta^* m(-\theta^*)) \left(1 + \frac{M(-\theta^*)}{M(-\theta^*) - m(-\theta^*)^2}\right). \tag{4.13}$$

Then the sequence $\{\theta_d^{plg}\}$ almost surely does not converge to θ^*.

Proof. We again only sketch the proof here, the full proof is deferred to Section 4.6. The proof for the first statement follows the same strategy as in Theorem 23, but now deals with the sample quantities $\hat{\Sigma}, \hat{\beta}$ in place of the population quantities $\Sigma, \tilde{\beta}$. Similarly as for $\tilde{\beta}$, we treat $\hat{\beta}$ by combining the equations $\hat{\beta} = (XX^T)^+ XY$, $Y = X^T\tilde{\beta} + E$ for $E \sim \mathcal{N}(0, \tilde{\sigma}^2 I_n)$, and $\tilde{\beta} = \beta + M^{+T}\alpha$ to obtain $\hat{\beta} = \begin{pmatrix}\sigma_\alpha M^{+T} & \sigma_\beta I_d & \tilde{\sigma}(XX^T)^+ X\end{pmatrix} v$ for some $v \sim \mathcal{N}(0, I_{l+d+n})$. Additional complications arise because $\hat{\beta}$ depends on both the population term M and the empirical quantities. This produces mixed terms $\mathrm{Tr}[(\hat{\Sigma} + \theta)^{-1}\hat{\Sigma}\Sigma^+]$ for $k \in \{1, 2\}$, which need to be treated with a separate result by Ledoit and Péché (2011) in Lemma 34.

For the second statement, we use similar arguments as in the proof of Theorem 23 to show that the convergence $\theta_d^{\mathrm{plg}} \to \theta^*$ implies that θ^* is a root of the right hand side in Eq. (4.12). This is equivalent to Eq. (4.13), which does not hold by assumption. □

The limiting derivative for the plug-in estimator in Eq. (4.12) is phrased in terms of the limiting sample distribution μ instead of the limiting population distribution ν. The main structural difference to Eq. (4.9) is the existence of an additional term $\gamma\theta^*(1 - \theta m(-\theta))(1 + M(-\theta)/(M(-\theta) - m(-\theta)^2))$, which prevents a closed-form expression for the corresponding roots θ^{plg} of this function. We therefore cannot directly exclude the possibility that θ^* is a root, in which case the plug-in estimator would be consistent. However, by simply plugging in θ^* in the limiting derivative, we see that θ^* being a root is equivalent to the condition in Eq. (4.13). This condition generally does not hold, because the limiting ratio of dimensions $\tilde{\gamma} = \lim_{d,l\to\infty} l/d$ on the left hand side stands in no special relationship to the terms on the right hand side. Therefore, the plug-in estimator θ^{plg} is generally a biased estimator for θ^*. This means that the resulting plug-in estimator for confounding strength ζ^{plg} is generally a biased estimator for the true confounding strength ζ.

4.4 A consistent estimator for confounding strength

In this section, we derive a novel estimator for confounding strength using tools from random matrix theory. We show that this estimator consistently recovers the true confounding strength in the high-dimensional asymptotic limit $(n, d \to \infty, d/n \to \gamma \in (0, 1))$. To this end, we can derive a consistent estimator of θ^{RMT} by first consistently estimating $f^{\mathrm{pop}}(\theta)$ and then finding the minimizer of this function. While this

procedure indeed yields a consistent estimator, it is stochastic, which can adversely affect the optimization algorithm at finite d. Therefore, we also provide a consistent estimator based on finding the zeros of $\partial_\theta f^{\text{pop}}(\theta)$ which is deterministic given a fixed sample. Coupled with the consistent estimator for τ^{pop} in Proposition 25, we arrive at a consistent estimator for confounding strength.

4.4.1 A consistent estimator for $f^{\text{pop}}(\theta)$.

Recall from Eq. (4.6) that maximum likelihood estimation of θ^* is equivalent to the optimization problem

$$\theta^{\text{pop}} := \arg\min_{\theta \geq 0} f^{\text{pop}}(\theta), \quad \text{where} \quad f^{\text{pop}}(\theta) = \frac{1}{d}\log\det(\Sigma+\theta) + \log\left\langle \frac{\tilde{\beta}}{\|\tilde{\beta}\|}, \Sigma(\Sigma+\theta)^{-1}\frac{\tilde{\beta}}{\|\tilde{\beta}\|}\right\rangle .$$

To consistently estimate $f^{\text{pop}}(\theta)$, it suffices to consistently estimate the two quantities $\frac{1}{d}\log\det(\Sigma+\theta)$ and $\log\langle\frac{\tilde{\beta}}{\|\tilde{\beta}\|}, \Sigma(\Sigma+\theta)^{-1}\frac{\tilde{\beta}}{\|\tilde{\beta}\|}\rangle$. We derive such estimators in Theorems 27 and 28 using tools from random matrix theory. The main results are included here and we defer the proofs to Section 4.7.

Theorem 27 (A consistent estimator for log determinant, (Kammoun et al., 2011)). *For any $\theta \in \mathbb{R}^+$, let $W = X + \sqrt{\theta}E$, where $E \in \mathbb{R}^{d\times n}$ is a random matrix with standard normal entries. Then, as $d, n \to \infty$ such that $d/n \to \gamma \in (0,1)$,*

$$\log\theta + \frac{1}{d}\log\det\frac{1}{n\theta}WW^T + (1-\gamma)\log\frac{\gamma-1}{\gamma} + 1 - \frac{1}{d}\log\det(\Sigma+\theta) \xrightarrow{a.s.} 0\,.$$

In other words, the function $g_1(\theta) = \log\theta + \frac{1}{d}\log\det\frac{1}{n\theta}WW^T + (1-\gamma)\log((\gamma-1)/\gamma) + 1$ is a consistent estimator of $\log\det(\Sigma+\theta)$.

Proposition 28 (A consistent estimator for the quadform). *Under Assumption 18, for any $\theta \in \mathbb{R}^+$, let η be the unique solution in $\mathbb{R}^-$ satisfying $\tilde{m}(\eta) = 1/\theta$. Then, as $d, n \to \infty$ such that $d/n \to \gamma \in (0,1)$,*

$$\frac{\frac{1}{d}\langle\hat{\beta}, \hat{\Sigma}(\hat{\Sigma}-\eta)^{-1}\hat{\beta}\rangle - \frac{S}{\theta} - \frac{S(1-\gamma)}{\eta}}{\frac{1}{d}\|\hat{\beta}\|^2 - S\gamma m(0)} - \left\langle \frac{\tilde{\beta}}{\|\tilde{\beta}\|}, \Sigma(\Sigma+\theta)^{-1}\frac{\tilde{\beta}}{\|\tilde{\beta}\|}\right\rangle \xrightarrow{a.s.} 0,$$

where $S = (1-\gamma)^{-1}\|Y\|^2_{I-X^+X}/(nd)$.

In other words, the function $g_2(\theta) = \log\frac{\frac{1}{d}\langle\hat{\beta},\hat{\Sigma}(\hat{\Sigma}-\eta)^{-1}\hat{\beta}\rangle - \frac{S}{\theta} - \frac{S(1-\gamma)}{\eta}}{\frac{1}{d}\|\hat{\beta}\|^2 - S\gamma m(0)}$ is a consistent estimator of $\log\langle\frac{\tilde{\beta}}{\|\tilde{\beta}\|}, \Sigma(\Sigma+\theta)^{-1}\frac{\tilde{\beta}}{\|\tilde{\beta}\|}\rangle$. Thereby, for every $\theta \in \mathbb{R}^+$, as $n, d \to \infty$ as $d/n \to \gamma \in (0,1)$,

$$g_1(\theta) + g_2(\theta) - f^{\text{pop}}(\theta) \xrightarrow{a.s.} 0 \tag{4.14}$$

In other words, a consistent estimator of $f^{\text{pop}}(\theta)$ is be given by $f^{\text{RMT}}(\theta) \coloneqq g_1(\theta)+g_2(\theta)$.

Stochasticity of the estimation. Observe that the estimator for the log determinant given by $g_1(\theta)$ is not a deterministic function of a given sample X, Y since the matrix W is stochastic. Following arguments similar to the proof of Theorems 23 and 26[2], we can indeed obtain an asymptotically consistent estimator for confounding strength. However, at finite d our experiments suggest that the stochasticity can adversely affect the optimization step. Furthermore, the dependence of $g_1(\theta)$ on θ is highly non-linear. Iterative optimization procedures require multiple evaluations (and therefore estimation of) $g_1(\theta)$ which considerably increases the computation complexity. To overcome these limitations, we also provide a deterministic and consistent estimator of θ by first consistently estimating the function $\partial_\theta f^{\text{pop}}(\theta)$ for any $\theta \in \mathbb{R}^+$ and showing that the roots of the estimating function asymptotically converges to θ^*.

4.4.2 A consistent estiator for $\partial_\theta f^{\text{pop}}(\theta)$.

As derived in Eq. (4.8), the derivative of the lop probability function $f^{\text{pop}}(\theta)$ is given by

$$\partial_\theta f^{\text{pop}}(\theta) = \frac{\langle \frac{\tilde{\beta}}{\|\tilde{\beta}\|}, \Sigma(\Sigma+\theta)^{-1}\frac{\tilde{\beta}}{\|\tilde{\beta}\|}\rangle \cdot m_\Sigma(-\theta) - \langle \frac{\tilde{\beta}}{\|\tilde{\beta}\|}, \Sigma(\Sigma+\theta)^{-2}\frac{\tilde{\beta}}{\|\tilde{\beta}\|}\rangle}{\langle \frac{\tilde{\beta}}{\|\tilde{\beta}\|}, \Sigma(\Sigma+\theta)^{-1}\frac{\tilde{\beta}}{\|\tilde{\beta}\|}\rangle}\,.$$

In order to consistently estimate $\partial_\theta f^{\text{pop}}(\theta)$, it suffices to consistently estimate the three quantities $m_\Sigma(-\theta)$, $\langle \frac{\tilde{\beta}}{\|\tilde{\beta}\|}, \Sigma(\Sigma+\theta)^{-1}\frac{\tilde{\beta}}{\|\tilde{\beta}\|}\rangle$, and $\langle \frac{\tilde{\beta}}{\|\tilde{\beta}\|}, \Sigma(\Sigma+\theta)^{-2}\frac{\tilde{\beta}}{\|\tilde{\beta}\|}\rangle$. Proposition 28 provides us with a consistent estimator for the quantity $\langle \frac{\tilde{\beta}}{\|\tilde{\beta}\|}, \Sigma(\Sigma+\theta)^{-1}\frac{\tilde{\beta}}{\|\tilde{\beta}\|}\rangle$. In Propositions 29 and 30, we derive estimators for the remaining quantities.

Proposition 29 (Estimation of Stieltjes transform). *Under the assumptions of Theorem 17, for any $\theta \in \mathbb{R}^+$, let η be the unique solution in $\mathbb{R}^-$ satisfying $\tilde{m}(\eta) = 1/\theta$. Then, as $d, n \to \infty$ such that $d/n \to \gamma \in (0,1)$,*

$$-\frac{1}{\gamma\theta}\left(\frac{\eta}{\theta} - \gamma + 1\right) - m_\Sigma(-\theta) \xrightarrow{a.s.} 0\,.$$

Proof. From Theorem 17, we have that for any $z \in \mathbb{C}/\mathbb{R}^+$, $m_\nu(-\frac{1}{\tilde{m}(z)}) = (1-\gamma-zm(z))\,m(z)$. Letting $\eta \in \mathbb{R}^-$ such that $\tilde{m}(\eta) = 1/\theta$, we arrive at the estimator. □

[2] with an additional argument to deal with the stochasticity of the log det estimator.

Now, we present a consistent estimator of the quadratic form $\langle \frac{\tilde{\beta}}{\|\tilde{\beta}\|}, \Sigma(\Sigma+\theta)^{-2}\frac{\tilde{\beta}}{\|\tilde{\beta}\|}\rangle$. From Proposition 28, we know that for any $\theta \in \mathbb{R}^+$, $g_2(\theta)$ is a consistent estimator of $\langle \frac{\tilde{\beta}}{\|\tilde{\beta}\|}, \Sigma(\Sigma+\theta)^{-1}\frac{\tilde{\beta}}{\|\tilde{\beta}\|}\rangle$. To derive an estimator of the quadratic form $\langle \frac{\tilde{\beta}}{\|\tilde{\beta}\|}, \Sigma(\Sigma+\theta)^{-2}\frac{\tilde{\beta}}{\|\tilde{\beta}\|}\rangle$, we utilize the so-called derivative trick (Dobriban and Wager, 2018; Hastie et al., 2022). First observe that

$$\langle \tilde{\beta}, \Sigma(\Sigma+\theta)^{-2}\tilde{\beta}\rangle = -\partial_\theta \left(\langle \tilde{\beta}, \Sigma(\Sigma+\theta)^{-1}\tilde{\beta}\rangle \right).$$

Furthermore, for every fixed $\theta \in \mathbb{R}^+$, we know that as $n, d \to \infty$ and $d/n \to \gamma \in (0,1)$,

$$g_2(\theta) - \left\langle \frac{\tilde{\beta}}{\|\tilde{\beta}\|}, \Sigma(\Sigma+\theta)^{-1}\frac{\tilde{\beta}}{\|\tilde{\beta}\|} \right\rangle \xrightarrow{a.s.} 0$$

It is also easy to verify that $g_2(\theta) - \langle \frac{\tilde{\beta}}{\|\tilde{\beta}\|}, \Sigma(\Sigma+\theta)^{-1}\frac{\tilde{\beta}}{\|\tilde{\beta}\|}\rangle$ is analytic and uniformly bounded in θ in the domain $\mathbb{R}^+$. Therefore, we can apply Vitali's convergence theorem to show that the limit of the derivatives converges to the derivative of the limit. Therefore a consistent estimator for the quadratic form $\langle \frac{\tilde{\beta}}{\|\tilde{\beta}\|}, \Sigma(\Sigma+\theta)^{-2}\frac{\tilde{\beta}}{\|\tilde{\beta}\|}\rangle$ is given by $-\partial_\theta g_2(\theta)$ and is formally presented in Theorem 30.

Proposition 30 (Consistent estimator for quadratic form). *For any $\theta \in \mathbb{R}^+$, let η be the unique solution in $\mathbb{R}^-$ satisfying $\tilde{m}(\eta) = 1/\theta$ and let $\eta' = 1/(\theta^2\tilde{m}'(\eta))$. As $d, n \to \infty$ such that $d/n \to \gamma \in (0,1)$,*

$$\frac{\frac{\eta'}{d}\langle \hat{\beta}, \hat{\Sigma}(\hat{\Sigma}+\theta)^{-2}\hat{\beta}\rangle - \frac{S}{\theta^2} + \frac{S\eta'(1-\gamma)}{\eta^2}}{\frac{1}{d}\|\hat{\beta}\|^2 - S\gamma m(0)} - \left\langle \frac{\tilde{\beta}}{\|\tilde{\beta}\|}, \Sigma(\Sigma+\theta)^{-2}\frac{\tilde{\beta}}{\|\tilde{\beta}\|} \right\rangle \xrightarrow{a.s.} 0,$$

where $S = \frac{1}{(1-\gamma)nd}\|Y\|^2_{I-X^+X}/(nd)$.

From Propositions 28, 29, and 30, for any $\theta \in \mathbb{R}^+$, a consistent estimator of $\partial_\theta f^{\text{pop}}(\theta)$ is given by

$$h_{\text{RMT}}(\theta) := \frac{\frac{g_2(\theta)}{\gamma\theta}(\gamma - 1 - \frac{\eta}{\theta}) - \partial_\theta g_2(\theta)}{g_2(\theta)}.$$

The RMT estimator for confounding strength is then naturally defined via the roots of $h_{\text{RMT}}(\theta)$ and RMT-corrected estimate of τ^{pop} as is formally presented in Definition 31 which consistently estimates the the true confounding strength ζ.

Definition 31 (RMT estimator for confounding strength). The *RMT estimator* for confounding strength ζ^{RMT} can then be defined as

$$\zeta^{\text{RMT}} = \frac{\tau^{\text{RMT}} \cdot \theta^{\text{RMT}}}{1 + \tau^{\text{RMT}} \cdot \theta^{\text{RMT}}}, \tag{4.15}$$

where $\tau^{\text{RMT}} = (1-\gamma)\tau^{\text{plg}}$ and θ^{RMT} is a root of $h_{\text{RMT}}(\theta)$ if it exists and 0 otherwise.

Theorem 32 (RMT estimator is consistent). *Let θ_d^{RMT} be defined as a root of $h_{RMT}(\theta)$ in some $[0, C]$ for some $C < \infty$ if it exists or 0 otherwise. Additionally, assume that ν is not degenerate. Then, under Assumption 18 with $\theta^* > 0$, the sequence $\{\theta_d^{RMT}\}$ converges a.s to θ^*.*

4.5 Proof of Theorem 23

This section gives the full proof of Theorem 23 for the asymptotic behavior of the population estimator for confounding strength. We state the theorem here again for reference.

Theorem 23 (Population estimator is consistent). *Under Assumption 18 with $\theta^* > 0$,*

1. *For every $\theta \geq 0$, the derivative of the function from Eqs. (4.6) satisfies*

$$\partial_\theta f_d^{pop}(\theta) \xrightarrow{a.s.} (\theta - \theta^*) \operatorname{Var}_{\lambda\sim\nu}\left[\frac{1}{\lambda+\theta}\right] \mathbb{E}_{\lambda\sim\nu}\left[\frac{\lambda+\theta^*}{\lambda+\theta}\right]^{-1}, \tag{4.9}$$

2. *For some $C > \theta^*$ and every $d \in \mathbb{N}$, let θ_d^{pop} be a root of $\partial_\theta f_d^{pop}$ in $[0, C]$ if it exists or 0 otherwise. Additionally, assume that ν is not degenerate. Then the sequence $\{\theta_d^{pop}\}$ converges to θ^* almost surely.*

Proof. We first show Eq. (4.9). According to Eq. (4.8), the function is given by $\partial_\theta f^{\text{pop}}(\theta) = m_\Sigma(-\theta) - \frac{1}{d}\tilde{\beta}^T\Sigma(\Sigma+\theta)^{-2}\tilde{\beta} / \frac{1}{d}\tilde{\beta}^T\Sigma(\Sigma+\theta)^{-1}\tilde{\beta}$. The first term $m_\Sigma(-\theta)$ converges to $m_\nu(-\theta)$ by assumption. The two quadratic forms are handled by Lemma 20 after rewriting $\tilde{\beta} = \beta + M^{+T}\alpha = \begin{pmatrix}\sigma_\alpha M^{+T} & \sigma_\beta I_d\end{pmatrix} u$ for some $u \sim \mathcal{N}(0, I_{l+d})$, which is possible because by assumption $\alpha \sim \mathcal{N}(0, \sigma_\alpha^2 I_l)$ and $\beta \sim \mathcal{N}(0, \sigma_\beta^2 I_d)$ are independent.

$$
\begin{aligned}
\frac{1}{d}\tilde{\beta}^T\Sigma(\Sigma+\theta)^{-1}\tilde{\beta} &= \frac{1}{d}u^T\begin{pmatrix}\sigma_\alpha M^+\\ \sigma_\beta I_d\end{pmatrix}\Sigma(\Sigma+\theta)^{-1}\begin{pmatrix}\sigma_\alpha M^{+T} & \sigma_\beta I_d\end{pmatrix}u \\
&\overset{a.s.}{\approx} \frac{1}{d}\operatorname{Tr}\left[\begin{pmatrix}\sigma_\alpha M^+\\ \sigma_\beta I_d\end{pmatrix}\Sigma(\Sigma+\theta)^{-1}\begin{pmatrix}\sigma_\alpha M^{+T} & \sigma_\beta I_d\end{pmatrix}\right] && \text{(Lemma 20)} \\
&= \frac{1}{d}\operatorname{Tr}\left[\Sigma(\Sigma+\theta)^{-1}\begin{pmatrix}\sigma_\alpha M^{+T} & \sigma_\beta I_d\end{pmatrix}\begin{pmatrix}\sigma_\alpha M^+\\ \sigma_\beta I_d\end{pmatrix}\right] && \text{(Trace cyclic)} \\
&= \frac{1}{d}\operatorname{Tr}\left[\Sigma(\Sigma+\theta)^{-1}(\sigma_\alpha^2\Sigma^{-1}+\sigma_\beta^2 I_d)\right] && (\Sigma = MM^T) \\
&= \frac{\sigma_\beta^2}{d}\operatorname{Tr}\left[(\Sigma+\theta)^{-1}(\Sigma+\theta^*)\right] && (\theta^* = \sigma_\alpha^2/\sigma_\beta^2) \\
&\xrightarrow[d\to\infty]{a.s.} \sigma_\beta^2\mathbb{E}_{\lambda\sim\nu}\left[\frac{\lambda+\theta^*}{\lambda+\theta}\right]. && (\mu_\Sigma \to \nu)
\end{aligned}
$$

Similarly, we get $\frac{1}{d}\tilde{\beta}^T\Sigma(\Sigma+\theta)^{-2}\tilde{\beta} \xrightarrow[d\to\infty]{a.s.} \sigma_\beta^2\mathbb{E}_{\lambda\sim\nu}\left[\frac{\lambda+\theta^*}{(\lambda+\theta)^2}\right]$. Plugging everything together yields

$$
\begin{aligned}
\partial_\theta f^{\text{pop}}(\theta) &\xrightarrow[d\to\infty]{a.s.} m_\nu(-\theta) - \frac{\mathbb{E}_{\lambda\sim\nu}\left[\frac{\lambda+\theta^*}{(\lambda+\theta)^2}\right]}{\mathbb{E}_{\lambda\sim\nu}\left[\frac{\lambda+\theta^*}{\lambda+\theta}\right]} \\
&= \left(m_\nu(-\theta)\cdot\mathbb{E}_{\lambda\sim\nu}\left[\frac{\lambda+\theta^*}{\lambda+\theta}\right] - \mathbb{E}_{\lambda\sim\nu}\left[\frac{\lambda+\theta^*}{(\lambda+\theta)^2}\right]\right)\mathbb{E}_{\lambda\sim\nu}\left[\frac{\lambda+\theta^*}{\lambda+\theta}\right]^{-1}.
\end{aligned}
$$

Using $m_\nu(-\theta) = \mathbb{E}_{\lambda\sim\nu}\left[\frac{1}{\lambda+\theta}\right]$ and the identity $\frac{\lambda+\theta^*}{\lambda+\theta} = 1-(\theta-\theta^*)\frac{1}{1+\lambda}$, we can simplify the first factor

$$
\begin{aligned}
&m_\nu(-\theta)\cdot\mathbb{E}_{\lambda\sim\nu}\left[\frac{\lambda+\theta^*}{\lambda+\theta}\right] - \mathbb{E}_{\lambda\sim\nu}\left[\frac{\lambda+\theta^*}{(\lambda+\theta)^2}\right] \\
=&\mathbb{E}_{\lambda\sim\nu}\left[\frac{1}{\lambda+\theta}\right]\left(1-(\theta-\theta^*)\mathbb{E}_{\lambda\sim\nu}\left[\frac{1}{\lambda+\theta}\right]\right) - \mathbb{E}_{\lambda\sim\nu}\left[\frac{1}{\lambda+\theta}\right] + (\theta-\theta^*)\mathbb{E}_{\lambda\sim\nu}\left[\frac{1}{(\lambda+\theta)^2}\right] \\
=&(\theta-\theta^*)\left(\mathbb{E}_{\lambda\sim\nu}\left[\frac{1}{(\lambda+\theta)^2}\right] - \mathbb{E}_{\lambda\sim\nu}\left[\frac{1}{\lambda+\theta}\right]^2\right) \\
=&(\theta-\theta^*)\operatorname{Var}_{\lambda\sim\nu}\left[\frac{1}{\lambda+\theta}\right].
\end{aligned}
$$

This concludes the first part of the proof.

For the second statement, first observe that the almost sure convergence in Eq. (4.9) for each $\theta \geq 0$ implies that this convergence also holds almost surely on a countable set

such as $[0, C] \cap \mathbb{Q}$. Since each function $\partial_\theta f_d^{\text{pop}}$ is analytic and bounded on $[0, C]$, we can further upgrade Eq. (4.9) to almost surely uniform convergence on $[0, C]$ by Vitali's convergence theorem. Now let $(\theta_d^{\text{pop}})_{d\in\mathbb{N}}$ be a sequence of roots as described in the theorem and let $F^{pop}(\theta)$ denote the function on the right hand side of Eq. (4.9). First note that the functions $\partial_\theta f_d^{\text{pop}}$ eventually have a root θ_d^{pop} in $[0, C]$ with probability 1: since $\theta^* < C$, there exist θ_-, θ_+ with $0 < \theta_- < \theta^* < \theta_+ < C$ with $F^{pop}(\theta_-) < 0$ and $F^{pop}(\theta_+) > 0$. The convergence of the functions $\partial_\theta f_d^{\text{pop}}$ then implies that $\partial_\theta f_d^{\text{pop}}(\theta_-) < 0$ and $\partial_\theta f_d^{\text{pop}}(\theta_+) > 0$ eventually. Since $\partial_\theta f_d^{\text{pop}}$ is continuous, the intermediate value theorem then implies the existence of a root in $(\theta_-, \theta_+) \subset [0, C]$. The proof is concluded with Hurwitz's theorem, which states that the sequence of roots $(\theta_d^{\text{pop}})_{d\in\mathbb{N}}$ of analytic functions converges to the unique root θ^* of the limiting function. □

4.6 Proof of Theorem 26

For the proof of Theorem 26 about the asymptotic behavior of the plug-in estimator, we require additional technical statements. The first characterizes the asymptotic behavior of the statistical noise for our causal model.

Lemma 33 (Asymptotics of the statistical noise). *Under Assumption 18, the statistical noise $\tilde{\sigma}^2$ concentrates as*

$$\frac{\tilde{\sigma}^2}{d} - (\tilde{\gamma} - 1)\sigma_\alpha^2 \xrightarrow[d\to\infty]{a.s.} 0\,.$$

Proof. According to Proposition 14, the statistical noise is given by $\tilde{\sigma}^2 = \sigma^2 + \|\alpha\|^2_{I_l - M^+M}$. The term σ^2 is assumed to be constant, but the quadratic form $\|\alpha\|^2_{I_l - M^+M}$ grows with d and is controlled by Lemma 20 as

$$\begin{aligned}\frac{\tilde{\sigma}^2}{d} = \frac{\sigma^2}{d} + \frac{1}{d}\alpha^T(I_l - M^+M)\alpha \overset{a.s.}{\approx} \frac{\operatorname{Tr}(I_l - M^+M)}{d}\sigma_\alpha^2 &= \frac{(l - \operatorname{Tr}(MM^+))}{d}\sigma_\alpha^2 \\ &= \frac{l-d}{d}\sigma_\alpha^2 \\ &= (\tilde{\gamma} - 1)\sigma_\alpha^2\,.\end{aligned}$$

□

The second technical lemma covers the asymptotic behavior of traces that involve both the sample covariance matrix $\hat{\Sigma}$ and the population covariance matrix Σ:

Lemma 34 (Asymptotics of mixed terms). *Under Assumption 18, it holds for any $\theta \geq 0$ that*

$$\frac{1}{d}\operatorname{Tr}\left[\left(\hat{\Sigma} + \theta\right)^{-1}\hat{\Sigma}\Sigma^+\right] \xrightarrow[d\to\infty]{a.s.} \gamma\theta m(-\theta)^2 + (1-\gamma)m(-\theta)$$

and

$$\frac{1}{d}\operatorname{Tr}\left[\left(\hat{\Sigma}+\theta\right)^{-2}\hat{\Sigma}\Sigma^{+}\right]\xrightarrow[d\to\infty]{a.s.}-\gamma m(-\theta)^2+2\gamma\theta m(-\theta)M(-\theta)+(1-\gamma)M(-\theta)\,,$$

where $m(-\theta)=\mathbb{E}_{\lambda\sim\mu}\left[\frac{1}{\lambda+\theta}\right]$ *and* $M(-\theta)=\mathbb{E}_{\lambda\sim\mu}\left[\frac{1}{(\lambda+\theta)^2}\right]$.

Proof. The asymptotic behavior of these quadratic forms is not covered by Theorem 17, because the dependencies between $\hat{\Sigma}$ and Σ create complications. To treat these we require an additional result by Ledoit and Péché (2011) combined with Vitali's convergence theorem which, in our notation, states that

$$\frac{1}{d}\operatorname{Tr}\left((\hat{\Sigma}-z)^{-1}g(\Sigma)\right)\xrightarrow[d\to\infty]{a.s.}-\frac{1}{z}\mathbb{E}_{\lambda\sim\nu}\left[\frac{g(\lambda)}{\tilde{m}(z)\lambda+1}\right].$$

We first use this result to obtain the limit for $\frac{1}{d}\operatorname{Tr}\left((\hat{\Sigma}-z)^{-1}\Sigma^{+}\right)$ by considering $g(\lambda)=1/\lambda$ and the identity

$$-\frac{1}{z\lambda}\frac{1}{\tilde{m}(z)\lambda+1}=\frac{1}{z}\left(\frac{1}{\lambda-\left(-\frac{1}{\tilde{m}(z)}\right)}-\frac{1}{\lambda}\right),$$

which yields

$$\frac{1}{d}\operatorname{Tr}\left((\hat{\Sigma}-z)^{-1}\Sigma^{+}\right)\xrightarrow[d\to\infty]{a.s.}\mathbb{E}_{\lambda\sim\nu}\left[-\frac{1}{z\lambda}\frac{1}{\tilde{m}(z)\lambda+1}\right]=\frac{1}{z}m_\nu\left(-\frac{1}{\tilde{m}(z)}\right)-\frac{1}{z}m_\nu(0)\,,$$

where we recall that $m_\nu(z)=\mathbb{E}_{\lambda\sim\nu}[\frac{1}{\lambda-z}]$. To relate the population Stieltjes transform m_ν back to the sample Stieltjes transforms m and $\tilde{m}$, we can use the identities from Theorem 17 to obtain

$$\begin{aligned}\frac{1}{d}\operatorname{Tr}\left((\hat{\Sigma}-z)^{-1}\Sigma^{+}\right)&\xrightarrow[d\to\infty]{a.s.}-\gamma m(z)\tilde{m}(z)-\frac{1}{z}m_\nu(0) && \text{(Eq. (4.4))}\\ &=-\gamma m(z)^2+\frac{1-\gamma}{z}m(z)-\frac{1}{z}m_\nu(0)\,. && \text{(Eq. (4.3))}\end{aligned}$$

Evaluating the above expression at $z=-\theta$ then yields

$$\frac{1}{d}\operatorname{Tr}\left((\hat{\Sigma}+\theta)^{-1}\Sigma^{+}\right)\xrightarrow[d\to\infty]{a.s.}-\gamma m(-\theta)^2-\frac{1-\gamma}{\theta}m(-\theta)+\frac{1}{\theta}m_\nu(0)\,.$$

All that remains is to relate $(\hat{\Sigma}+\theta)^{-1}\Sigma^{+}$ to the terms we are interested in. Using the identity $(\hat{\Sigma}+\theta)^{-1}\hat{\Sigma}=I-\theta(\hat{\Sigma}+\theta)^{-1}$, we get the first statement of this lemma

$$\begin{aligned}\frac{1}{d}\operatorname{Tr}\left[\left(\hat{\Sigma}+\theta\right)^{-1}\hat{\Sigma}\Sigma^{+}\right]=&\frac{1}{d}\operatorname{Tr}\left[\Sigma^{+}\right]-\theta\frac{1}{d}\operatorname{Tr}\left[\left(\hat{\Sigma}+\theta\right)^{-1}\Sigma^{+}\right]\\ \xrightarrow[d\to\infty]{a.s.}&m_\nu(0)-\theta\left(-\gamma m(-\theta)^2-\frac{1-\gamma}{\theta}m(-\theta)+\frac{1}{\theta}m_\nu(0)\right)\\ =&\gamma\theta m(-\theta)^2+(1-\gamma)m(-\theta)\,.\end{aligned}$$

The second statement of this lemma also follows directly by taking the derivative, which can be exchanged with the limit $d \to \infty$ using similar arguments as after Proposition 29, to obtain

$$\begin{aligned}\frac{1}{d}\operatorname{Tr}\left[\left(\hat{\Sigma}+\theta\right)^{-2}\hat{\Sigma}\Sigma^{+}\right] &= -\partial_\theta \frac{1}{d}\operatorname{Tr}\left[\left(\hat{\Sigma}+\theta\right)^{-1}\hat{\Sigma}\Sigma^{+}\right] \\ &\xrightarrow[d\to\infty]{a.s.} -\partial_\theta\left(\gamma\theta m(-\theta)^2 + (1-\gamma)m(-\theta)\right) \\ &= -\gamma m(-\theta)^2 + 2\gamma\theta m(-\theta)M(-\theta) + (1-\gamma)M(-\theta)\,,\end{aligned}$$

where the last step used $\partial_\theta m(-\theta) = M(-\theta)$. □

We are now ready to give the full proof of Theorem 26.

Theorem 26 (Plug-in estimator is generally biased). *Under Assumption 18 with* $\theta^* > 0$,

1. *For all* $\theta \geq 0$, *the derivative of the function from Eq. (4.11) satisfies*

$$\partial_\theta f_d^{plg}(\theta) \xrightarrow{a.s.} \left[\theta - (1+\gamma\tilde{\gamma})\theta^* + \gamma\theta^*(1-\theta m(-\theta))\left(1+\frac{M(-\theta)}{M(-\theta)-m(-\theta)^2}\right)\right] h(\theta)\,, \tag{4.12}$$

   with $h(\theta) = (M(-\theta) - m(-\theta)^2)(1 - \theta m(-\theta) + (1 - 2\gamma + \gamma\tilde{\gamma})\theta^* m(-\theta) + \gamma\theta\theta^* m(-\theta)^2)^{-1}$, *where* $m(-\theta) = \mathbb{E}_{\lambda\sim\mu}[1/(\lambda+\theta)]$, *and* $M(-\theta) = \mathbb{E}_{\lambda\sim\mu}\left[1/(\lambda+\theta)^2\right]$.

2. *For every* $d \in \mathbb{N}$, *let* θ_d^{plg} *be a root of* $\partial_\theta f_d^{plg}$ *if it exists or* 0 *otherwise. Additionally, assume that* $\tilde{\gamma}$ *does not satisfy*

$$\tilde{\gamma} = (1 - \theta^* m(-\theta^*))\left(1 + \frac{M(-\theta^*)}{M(-\theta^*) - m(-\theta^*)^2}\right)\,. \tag{4.13}$$

   Then the sequence $\{\theta_d^{plg}\}$ *almost surely does not converge to* θ^*.

Proof. We first show Eq. (4.12). This proof for the plug-in quantities $\hat{\Sigma}, \hat{\beta}$ follows the same strategy as the proof of Theorem 23 for $\Sigma, \tilde{\beta}$, but additional complications arise because $\hat{\beta}$ asymptotically depends on both the population term M and the empirical quantities. Similarly as for $\tilde{\beta}$, we treat $\hat{\beta}$ by combining the equations $\hat{\beta} = (XX^T)^+XY$, $Y = X^T\tilde{\beta} + E$ for $E \sim \mathcal{N}(0, \tilde{\sigma}^2 I_n)$, and $\tilde{\beta} = \beta + M^{+T}\alpha$ to obtain

$$\hat{\beta} = \begin{pmatrix}\sigma_\alpha M^{+T} & \sigma_\beta I_d & \tilde{\sigma}(XX^T)^+X\end{pmatrix} v \quad \text{for some } v \sim \mathcal{N}(0, I_{l+d+n})\,.$$

As before, we get for $k \in \{1, 2\}$ that

$$
\begin{aligned}
&\frac{1}{d}\hat\beta^T\hat\Sigma(\hat\Sigma+\theta)^{-k}\hat\beta \\
&= \frac{1}{d}v^T\begin{pmatrix}\sigma_\alpha M^+\\ \sigma_\beta I_d\\ \tilde\sigma X^T(XX^T)^+\end{pmatrix}\hat\Sigma(\hat\Sigma+\theta)^{-k}\begin{pmatrix}\sigma_\alpha M^{+T} & \sigma_\beta I_d & \tilde\sigma(XX^T)^+X\end{pmatrix}v \\
&\overset{a.s.}{\approx} \frac{1}{d}\operatorname{Tr}\left[\begin{pmatrix}\sigma_\alpha M^+\\ \sigma_\beta I_d\\ \tilde\sigma X^T(XX^T)^+\end{pmatrix}\hat\Sigma(\hat\Sigma+\theta)^{-k}\begin{pmatrix}\sigma_\alpha M^{+T} & \sigma_\beta I_d & \tilde\sigma(XX^T)^+X\end{pmatrix}\right] && \text{(Lemma 20)} \\
&= \frac{1}{d}\operatorname{Tr}\left[\hat\Sigma(\hat\Sigma+\theta)^{-k}\begin{pmatrix}\sigma_\alpha M^{+T} & \sigma_\beta I_d & \tilde\sigma(XX^T)^+X\end{pmatrix}\begin{pmatrix}\sigma_\alpha M^+\\ \sigma_\beta I_d\\ \tilde\sigma X^T(XX^T)^+\end{pmatrix}\right] && \text{(Trace cyclic)} \\
&= \frac{1}{d}\operatorname{Tr}\left[\hat\Sigma(\hat\Sigma+\theta)^{-k}\left(\sigma_\alpha^2\Sigma^+ + \sigma_\beta^2 I_d + \frac{\tilde\sigma^2}{n}\hat\Sigma^{-1}\right)\right] \\
&= \frac{1}{d}\operatorname{Tr}\left[\hat\Sigma(\hat\Sigma+\theta)^{-k}\left(\sigma_\alpha^2\Sigma^+ + \sigma_\beta^2 I_d + \gamma(\tilde\gamma-1)\sigma_\alpha^2\hat\Sigma^{-1}\right)\right] && \text{(Lemma 33)} \\
&= \frac{\sigma_\beta^2}{d}\operatorname{Tr}\left[(\hat\Sigma+\theta)^{-k}(\hat\Sigma+\gamma(\tilde\gamma-1)\theta^*)\right] + \theta^*\frac{\sigma_\beta^2}{d}\operatorname{Tr}\left[(\hat\Sigma+\theta)^{-k}\hat\Sigma\Sigma^+\right].
\end{aligned}
$$

The second term contains both the population term Σ and the sample term $\hat\Sigma$, which is treated separately in Lemma 34. For readability, we use the shorthand notation $m = \mathbb{E}_{\lambda\sim\mu}\left[\frac{1}{\lambda+\theta}\right]$ and $M = \mathbb{E}_{\lambda\sim\mu}\left[\frac{1}{(\lambda+\theta)^2}\right]$, under which the limit for the first term is given by

$$
\frac{1}{d}\operatorname{Tr}\left[(\hat\Sigma+\theta)^{-k}(\hat\Sigma+\gamma(\tilde\gamma-1)\theta^*)\right] \xrightarrow[d\to\infty]{a.s.} \begin{cases}1-\theta m+\gamma(\tilde\gamma-1)\theta^* m, & \text{for } k=1\\ m-\theta M+\gamma(\tilde\gamma-1)\theta^* M, & \text{for } k=2\end{cases}.
$$

Combined with Lemma 34, this yields

$$
\frac{1}{d}\hat\beta^T\hat\Sigma(\hat\Sigma+\theta)^{-k}\hat\beta \xrightarrow[d\to\infty]{a.s.} \begin{cases}1-\theta m+\theta^*(\gamma\theta m^2+(1-2\gamma+\gamma\tilde\gamma)m), & \text{for } k=1\\ m-\theta M+\theta^*(-\gamma m^2+2\gamma\theta mM+(1-2\gamma+\gamma\tilde\gamma)M), & \text{for } k=2\end{cases}.
$$

Together with $m_{\hat\Sigma}(-\theta) \xrightarrow[d\to\infty]{a.s.} m$, this covers the individual components of $\partial_\theta f^{\mathrm{plg}}(\theta) = m_{\hat\Sigma}(-\theta) - \frac{1}{d}\hat\beta^T\hat\Sigma(\hat\Sigma+\theta)^{-2}\hat\beta/\frac{1}{d}\hat\beta^T\hat\Sigma(\hat\Sigma+\theta)^{-1}\hat\beta$. It remains to plug everything in, which

we do after factoring out the denominator $\frac{1}{d}\hat{\beta}^T\hat{\Sigma}(\hat{\Sigma}+\theta)^{-1}\hat{\beta}$ to obtain

$$
\begin{aligned}
& m_{\hat{\Sigma}}(-\theta)\cdot\frac{1}{d}\hat{\beta}^T\hat{\Sigma}(\hat{\Sigma}+\theta)^{-1}\hat{\beta}-\frac{1}{d}\hat{\beta}^T\hat{\Sigma}(\hat{\Sigma}+\theta)^{-2}\hat{\beta} \\
\xrightarrow[d\to\infty]{a.s.}\quad & m\cdot\left[1-\theta m+\theta^*(\gamma\theta m^2+(1-2\gamma+\gamma\tilde{\gamma})m)\right] \\
& -\left[m-\theta M+\theta^*(-\gamma m^2+2\gamma\theta mM+(1-2\gamma+\gamma\tilde{\gamma})M)\right] \\
=&(\theta-(1-2\gamma+\gamma\tilde{\gamma})\theta^*)\cdot(M-m^2)+\gamma\theta^*\left(\theta m^3+m^2-2\theta mM\right) \\
=&(\theta-(1-2\gamma+\gamma\tilde{\gamma})\theta^*)\cdot(M-m^2)+\gamma\theta^*(2m^2-2M-(1-\theta m)m^2+2(1-\theta m)M) \\
=&(\theta-(1+\gamma\tilde{\gamma})\theta^*)\cdot(M-m^2)+\gamma\theta^*(1-\theta m)(2M-m^2) \\
=&\left[\theta-(1+\gamma\tilde{\gamma})\theta^*+\gamma\theta^*(1-\theta m)(1+\frac{M}{M-m^2})\right]\cdot(M-m^2)\,,
\end{aligned}
$$

which concludes the first part of the proof.

For the second statement, observe that Eq. (4.13) is equivalent to $F^{plg}(\theta^*)=0$, where F^{plg} is the function on the right hand side of Eq. (4.12). The assumption in this theorem therefore states that $F^{plg}(\theta^*)\neq 0$. Let $(\theta_d^{\text{plg}})_{d\in\mathbb{N}}$ be the sequence described in the theorem. In the case where $\partial_\theta f_d^{\text{plg}}$ does not have a root infinitely often, we have $\theta_d^{\text{plg}}=0$ infinitely often and therefore $\theta_d^{\text{plg}}\not\to\theta^*$ as $d\to\infty$ since $\theta^*\neq 0$. Therefore, now assume that θ_d^{plg} is a root of $\partial_\theta f_d^{\text{plg}}$ eventually. Assume that the claim is false, that is, $\theta_d^{\text{plg}}\xrightarrow[d\to\infty]{}\theta^*$ with positive probability. Similarly to the proof of Theorem 23, we get that the convergence in Eq. (4.12) holds almost surely uniformly on $[0,C]$ for some $C>\theta^*$. The convergence $\theta_d^{\text{plg}}\to\theta^*$ also implies that $\theta_d^{\text{plg}}\in[0,C]$ eventually. Putting everything together, we get for sufficiently large d that

$$
\begin{aligned}
|F^{plg}(\theta^*)| &= |F^{plg}(\theta^*)-\partial_\theta f_d^{\text{plg}}(\theta_d^{\text{plg}})| && (\partial_\theta f_d^{\text{plg}}(\theta_d^{\text{plg}})=0)\\
&\leq |\partial_\theta f_d^{\text{plg}}(\theta_d^{\text{plg}})-F^{plg}(\theta_d^{\text{plg}})|+|F^{plg}(\theta_d^{\text{plg}})-F^{plg}(\theta^*)| \\
&\leq \sup_{\theta\in[0,C]}|\partial_\theta f_d^{\text{plg}}(\theta)-F^{plg}(\theta)|+|F^{plg}(\theta_d^{\text{plg}})-F^{plg}(\theta^*)| \\
&\xrightarrow[d\to\infty]{} 0\,,
\end{aligned}
$$

where the first summand goes to 0 by uniform convergence and the second summand goes to 0 by continuity of F^{plg} and $\theta_d^{\text{plg}}\to\theta^*$. This implies $F^{plg}(\theta^*)=0$, which is a contradiction. □

4.7 RMT consistent estimators for quantitites of interest

Theorem 35 (Consistent estimation of statistical noise). *Under the model in Eq. (4.1),*

$$\frac{1}{1-\gamma}\frac{\|Y\|^2_{I-X^+X}}{nd} - \frac{\tilde{\sigma}^2}{d} \xrightarrow{a.s} 0\,.$$

Proof.

$$\frac{1}{nd}\|Y\|^2 = \frac{1}{nd}\|X\tilde{\beta}+E\|^2 = \frac{1}{nd}\tilde{\beta}^T X^T X\tilde{\beta} + \frac{1}{nd}E^T E + \frac{2}{nd}\tilde{\beta}^T X^T E.$$

We know that the minimum l_2 norm estimator admits a following closed form solution given by $\hat{\beta} = (X^TX)^+X^TY = (X^TX)^+X^T(X\tilde{\beta}+E) \overset{w.h.p}{=} \tilde{\beta} + (X^TX)^+X^TE$, where we used the fact that $\text{rank}(X^TX) = d$ w.h.p to arrive at the last equality. Letting $\kappa = (X^TX)^+X^TE$, we have

$$\begin{aligned}\frac{1}{nd}\hat{\beta}^T X^T X\hat{\beta} &= \frac{1}{nd}(\tilde{\beta}+\kappa)^T X^T X(\tilde{\beta}+\kappa),\\ &= \frac{1}{nd}\tilde{\beta}^T X^T X\tilde{\beta} + \frac{1}{nd}\kappa^T X^T X\kappa + \frac{2}{nd}\tilde{\beta}^T X^T X\kappa.\end{aligned}$$

From the closed form expression for $\hat{\beta}$,

$$\begin{aligned}\frac{1}{nd}\hat{\beta}^T X^T X\hat{\beta} &= \frac{1}{nd}Y^T X(X^TX)^+X^TX(X^TX)^+X^TY,\\ &= \frac{1}{nd}Y^T X(X^TX)^+X^TY,\\ &= \frac{1}{nd}Y^T XX^+Y.\end{aligned}$$

Similarly substituting $\kappa = (X^TX)^+X^TE$, we have

$$\begin{aligned}\frac{1}{nd}\kappa^T X^T X\kappa &= \frac{1}{nd}E^T X(X^TX)^+X^TX(X^TX)^+X^TE,\\ &= \frac{1}{nd}E^T X(X^TX)^+X^TE,\\ &= \frac{1}{nd}E^T XX^+E,\\ &= \frac{\gamma\tilde{\sigma}^2}{d} + \mathcal{O}(1/\sqrt{d}).\end{aligned}$$

To derive the last equality, we first apply Lemma 20 to show that $\frac{1}{nd}E^TXX^+E = \frac{\tilde{\sigma}^2}{nd}\text{Tr}[XX^+] + \mathcal{O}(1/\sqrt{p})$. The equality follows using $\text{Tr}[AA^+] = rank(A)$ for any $A \in \mathbb{R}^{n\times d}$ and

$$\frac{1}{nd}E^TXX^+E = \frac{\gamma\tilde{\sigma}^2}{d} + \mathcal{O}(1/\sqrt{d}).$$

Now let us consider the term $\frac{2}{nd}\tilde{\beta}^TX^TX\kappa$.

$$\begin{aligned}\frac{2}{nd}\tilde{\beta}^TX^TX\kappa &= \frac{2}{nd}\tilde{\beta}^TX^TX(X^TX)^+X^TE,\\ &= \frac{2}{nd}\tilde{\beta}^TX^TE \to 0 \text{ as } d\to\infty \qquad \text{(Hoeffding's inequality)}\end{aligned}$$

Following similar arguments, we have

$$\frac{1}{nd}E^TE = \frac{\tilde{\sigma}^2}{d} + \mathcal{O}(\frac{1}{d\sqrt{n}})$$

Putting everything together, we have

$$\begin{aligned}\frac{1}{nd}\|Y\|^2 &= \frac{1}{nd}Y^TXX^+Y - \frac{\gamma\tilde{\sigma}^2}{d} + \frac{\tilde{\sigma}^2}{d} + \mathcal{O}(1/\sqrt{d})\\ \frac{\tilde{\sigma}^2}{d} &= \frac{1}{(1-\gamma)nd}\|Y\|^2_{I-XX^+} + \mathcal{O}(1/\sqrt{d}).\end{aligned}$$

□

Lemma 36 (Asymptotics of quadratic form with a deterministic sequence). *For any $\theta \in \mathbb{R}^+$, let η be the unique solution in $\mathbb{R}^-$ satisfying $\tilde{m}(\eta) = 1/\theta$. Then, for any deterministic sequence of vectors $\{v_d\}$ with uniformly bounded (Euclidean) norm, as $d, n \to \infty$ such that $d/n \to \gamma \in (0,1)$,*

$$\langle v_d, \hat{\Sigma}(\hat{\Sigma}-\eta)^{-1}v_d\rangle - \langle v_d, \Sigma(\Sigma+\theta)^{-1}v_d\rangle \longrightarrow 0.$$

Proof. Observe that for any $\eta < 0$,

$$\langle v_d, \hat{\Sigma}(\hat{\Sigma}-\eta)^{-1}v_d\rangle = \|v_d\|^2 - \langle v_d, (\hat{\Sigma}-\eta)^{-1}v_d\rangle.$$

The result follows from the Generalized Marchenko Pastur Theorem (Silverstein and Bai, 1995), which states that for any $\theta \in \mathbb{R}^+$,

$$\langle v_d, (\hat{\Sigma}-\eta)^{-1}v_d\rangle - \langle v_d, (\Sigma+\theta)^{-1}v_d\rangle \longrightarrow 0.$$

□

Proposition 28 (A consistent estimator for the quadform). *Under Assumption 18, for any $\theta \in \mathbb{R}^+$, let η be the unique solution in $\mathbb{R}^-$ satisfying $\tilde{m}(\eta) = 1/\theta$. Then, as $d, n \to \infty$ such that $d/n \to \gamma \in (0,1)$,*

$$\frac{\frac{1}{d}\langle\hat{\beta}, \hat{\Sigma}(\hat{\Sigma}-\eta)^{-1}\hat{\beta}\rangle - \frac{S}{\theta} - \frac{S(1-\gamma)}{\eta}}{\frac{1}{d}\|\hat{\beta}\|^2 - S\gamma m(0)} - \left\langle \frac{\tilde{\beta}}{\|\tilde{\beta}\|}, \Sigma(\Sigma+\theta)^{-1}\frac{\tilde{\beta}}{\|\tilde{\beta}\|} \right\rangle \xrightarrow{a.s.} 0,$$

where $S = (1-\gamma)^{-1}\|Y\|^2_{I-X^+X}/(nd)$.

Proof. Let η be the unique solution in $\mathbb{R}^-$ satisfying $\tilde{m}(\eta) = 1/\theta$. From Lemma 36, we have for any $\theta \in \mathbb{R}^+$, as $n, d \to \infty$ such that $d/n \to \gamma \in (0,1)$,

$$\langle \frac{\tilde{\beta}}{\|\tilde{\beta}\|}, \hat{\Sigma}(\hat{\Sigma}-\eta)^{-1}\frac{\tilde{\beta}}{\|\tilde{\beta}\|}\rangle - \langle \frac{\tilde{\beta}}{\|\tilde{\beta}\|}, \Sigma(\Sigma+\theta)^{-1}\frac{\tilde{\beta}}{\|\tilde{\beta}\|}\rangle \xrightarrow{a.s} 0 \tag{4.16}$$

Therefore, it suffices to consistently estimate $\langle \frac{\tilde{\beta}}{\|\tilde{\beta}\|}, \hat{\Sigma}(\hat{\Sigma}-\eta)^{-1}\frac{\tilde{\beta}}{\|\tilde{\beta}\|}\rangle$. First, we characterize the asymptotic behavior of $\frac{1}{d}\langle\hat{\beta}, \hat{\Sigma}(\hat{\Sigma}-\eta)^{-1}\hat{\beta}\rangle$, where $\hat{\beta} = \tilde{\beta} + \tilde{\sigma}^2(XX^T)^+XE$, where $E \sim \mathcal{N}0I_n$.

$$\frac{1}{d}\langle\hat{\beta}, \hat{\Sigma}(\hat{\Sigma}-\eta)^{-1}\hat{\beta}\rangle = \frac{1}{d}\langle\tilde{\beta}, \hat{\Sigma}(\hat{\Sigma}-\eta)^{-1}\tilde{\beta}\rangle + \frac{2\tilde{\sigma}^2}{d}\tilde{\beta}^T\hat{\Sigma}(\hat{\Sigma}-\eta)^{-1}(XX^T)^+XE + \frac{\tilde{\sigma}^2}{d}E^TX^T(XX^T)^+\hat{\Sigma}(\hat{\Sigma}-\eta)^{-1}(XX^T)^+XE\,.$$

The first term in the expansion resembles the quantity of interest.

For the second term, notice that, since $E \sim \mathcal{N}0I_n$,

$$\frac{2\tilde{\sigma}^2}{d}\tilde{\beta}^T\hat{\Sigma}(\hat{\Sigma}-\eta)^{-1}(XX^T)^+XE \sim \mathcal{N}(0, \|\frac{2\tilde{\sigma}^2}{d}X^T(XX^T)^+\hat{\Sigma}(\hat{\Sigma}-\eta)^{-1}\tilde{\beta}\|^2),$$

where

$$\begin{aligned}\left\|\frac{2\tilde{\sigma}^2}{d}X^T(XX^T)^+\hat{\Sigma}(\hat{\Sigma}-\eta)^{-1}\tilde{\beta}\right\|^2 &= \frac{4\tilde{\sigma}^2}{d^2}\tilde{\beta}^T\hat{\Sigma}(\hat{\Sigma}-\eta)^{-1}(XX^T)^+XX^T(XX^T)^+\hat{\Sigma}(\hat{\Sigma}-\eta)^{-1}\tilde{\beta} \\ &= \frac{4\tilde{\sigma}^2}{d^2n}\tilde{\beta}^T\hat{\Sigma}(\hat{\Sigma}-\eta)^{-1}\hat{\Sigma}^+\hat{\Sigma}(\hat{\Sigma}-\eta)^{-1}\tilde{\beta} \\ &\xrightarrow{a.s} 0\,.\end{aligned}$$

Therefore, the second term vanishes. For the last expression,

$$\begin{aligned}
&\frac{\tilde{\sigma}^2}{d} E^T X^T (XX^T)^+ \hat{\Sigma}(\hat{\Sigma} - \eta)^{-1} (XX^T)^+ XE \\
=&\frac{\tilde{\sigma}^2}{d} \frac{1}{n^2} E^T X^T \hat{\Sigma}^+ \hat{\Sigma}(\hat{\Sigma} - \eta)^{-1} \hat{\Sigma}^+ XE \\
\xrightarrow{a.s} &\frac{\tilde{\sigma}^2}{d} \frac{1}{n} \mathrm{tr}\left(\hat{\Sigma}^+ \hat{\Sigma}(\hat{\Sigma} - \eta)^{-1}\right) \quad \text{(Trace Lemma, conditioned on } X\text{)} \\
\xrightarrow{a.s} &\gamma \frac{\tilde{\sigma}^2}{d} m(\eta).
\end{aligned}$$

From Theorem 17, we know that

$$m(\eta) = \frac{1}{\gamma}\left(\tilde{m}(\eta) + \frac{1-\gamma}{\eta}\right) = \frac{1}{\gamma}\left(\frac{1}{\theta} + \frac{1-\gamma}{\eta}\right).$$

Therefore,

$$\frac{\tilde{\sigma}^2}{d} E^T X^T (XX^T)^+ \hat{\Sigma}(\hat{\Sigma} - \eta)^{-1} (XX^T)^+ XE - \frac{\tilde{\sigma}^2}{d}\left(\frac{1}{\theta} + \frac{1-\gamma}{\eta}\right) \xrightarrow{a.s} 0.$$

Following the same arguments, it is easy to verify that

$$\frac{1}{d}\|\hat{\beta}\|^2 - \frac{\tilde{\sigma}^2}{d}\gamma m(0) - \frac{1}{d}\|\tilde{\beta}\|^2 \xrightarrow{a.s} 0\,.$$

Combining the estimators with the result from Theorem 35, we have the desired result. □

Theorem 32 (RMT estimator is consistent). *Let θ_d^{RMT} be defined as a root of $h_{RMT}(\theta)$ in some $[0, C]$ for some $C < \infty$ if it exists or 0 otherwise. Additionally, assume that ν is not degenerate. Then, under Assumption 18 with $\theta^* > 0$, the sequence $\{\theta_d^{RMT}\}$ converges a.s to θ^*.*

Proof. The proof follows following the same arguments as in the proof of 23. □

4.8 Discussion

We analyze the asymptotic behavior of the confounding strength estimator by Janzing and Schölkopf (2018) in the high-dimensional proportional regime. While the approach is consistent under population quantities, the corresponding plug-in estimator is generally biased. We correct for this bias and present a consistent estimator using tools from random matrix theory. More generally, high dimensions can help to identify the causal model, but they also warrant adapted estimators if the number of samples does not grow even faster than the dimensions.

In this work, we focus on obtaining estimators that consistently estimate the true confounding strength in the proportional asymptotic regime. An important direction for future work is to obtain non-asymptotic guarantees of convergence of the RMT estimator ζ^{RMT}. Obtaining convergence rates would further enhance the applicability of the RMT estimator. We leave this for future work.

Faithful estimation of confounding strength can indeed facilitate causal learning from observational data, for instance, via regularization. This has been empirically demonstrated in Janzing (2019) and under the same model setting as ours, precisely characterized in Vankadara et al. (2022). However, it is important to practice caution in applying such techniques more generally since causal learning or even estimation of confounding strength is a very hard problem and does require strong assumptions.

Chapter 5

Discussion

In this thesis, I discussed the general concept of inductive bias for machine learning, which describes the a priori preference for solutions that is necessary to generalize. As a specific application, I outlined the search for implicit inductive bias at the contemporary example of deep learning. My work aims to uncover hidden implicit bias and ranges from inductive bias in a specific deep learning algorithm to a generic diagnostic tool for inductive bias in Bayesian inference.

Is it useful to know the inductive bias of an algorithm? It could be argued that the goal of learning is generalization, which evidently can be achieved without fully understanding the inductive bias of the machine learning algorithm. But nowadays we do not only care about the performance, but also about making our algorithms trustworthy and interpretable, which is something that black boxes cannot provide. While notions such as interpretability address the learned predictor, inductive bias addresses the learning algorithm itself. As discussed in the no free lunch theorem, no algorithm is inherently superior to another in general, only on specific problem instances. The inductive bias of an algorithm informs us about the real-world applications in which we can expect it to work. Even more importantly, inductive bias tells us when we can expect the algorithm to fail, which is essential in safety-critical applications. Reversely, the success of an algorithm can inform us about properties of the specific problem instance, because successful generalization implies that the problem instance is aligned with the inductive bias. As demonstrated in Chapter 2, a more immediate advantage of knowing the inductive bias is the ability to simplify an algorithm in presentation, implementation, and execution. This is achieved by removing every component that complicates the learning procedure without changing its bias.

How do we express inductive bias? Technically, the inductive bias of an algorithm is trivially available: if an algorithm learns the predictor $\hat{f}$ given a data set D, we can simply define the data-dependent complexity measure $\Omega_D(\hat{f}) = 0$ and $\Omega_D(f) = \infty$

for $f \neq \hat{f}$. It is then clearly true that the algorithm uses this complexity measure to resolve ambiguities, for example when multiple candidate solutions have the same loss. However, such tautological explanations à la "it is what it is" do not help us understand the inductive bias. This raises the question of what we consider a good explanation, that is, in what form do we want to represent the inductive bias? The framework of regularized risk minimization answers that question with data-independent complexity measures, or data-dependent measures with special properties such as convexity. In Bayesian inference, the inductive bias is described by the data-independent prior distribution. Unfortunately, both these frameworks can be too restrictive to describe the actual inductive bias of an algorithm as discussed for complexity measures in Section 1.5 and for prior distributions in Chapter 3. As in the original learning problem, we have to trade off the clarity of an explanation with its flexibility to capture the bias of complex algorithms. Additionally, it might be useful to find other formulations of inductive bias that are a better fit for the algorithms we use in practice.

Should we avoid black box algorithms? The discussion above seems to indicate that black boxes such as deep learning with unclear inductive bias should be avoided. Machine learning practices that are based on heuristics and built through trial and error are compared to alchemy, because it is not clearly understood why they work. On the other hand, the success of deep learning showed that such practices can drive innovation faster than rigorous theory, which still struggles to explain deep learning and only started to catch up (Sections 1.4 and 1.5). I believe that the best way to progress is through joint efforts of exploration through practice and explanation through theory.

Bibliography

L. A. Adamic and N. Glance. The political blogosphere and the 2004 US election: divided they blog. *3rd international workshop on Link discovery*, 2005.

M. Arjovsky, S. Chintala, and L. Bottou. Wasserstein generative adversarial networks. *International Conference on Machine Learning (ICML)*, 2017.

B. C. Arnold and S. J. Press. Compatible conditional distributions. *Journal of the American Statistical Association*, 84(405):152–156, 1989.

B. C. Arnold, E. Castillo, and J. M. Sarabia. Conditionally specified distributions: An introduction (with comments and a rejoinder by the authors). *Statistical Science*, 16 (3):249 – 274, 2001.

B. C. Arnold, E. Castillo, and J. M. Sarabia. Exact and near compatibility of discrete conditional distributions. *Computational statistics & data analysis*, 40(2):231–252, 2002.

S. Arora, N. Cohen, W. Hu, and Y. Luo. Implicit regularization in deep matrix factorization. *Neural Information Processing Systems (NeurIPS)*, 2019.

S. Azulay, E. Moroshko, M. S. Nacson, B. E. Woodworth, N. Srebro, A. Globerson, and D. Soudry. On the implicit bias of initialization shape: Beyond infinitesimal mirror descent. *International Conference on Machine Learning (ICML)*, 2021.

J. L. Ba, J. R. Kiros, and G. E. Hinton. Layer normalization. *arXiv preprint arXiv:1607.06450*, 2016.

Z. Bai and J. W. Silverstein. *Spectral analysis of large dimensional random matrices*, volume 20. Springer, 2010.

L. Baldesi, C. T. Butts, and A. Markopoulou. Spectral graph forge: Graph generation targeting modularity. *Conference on Computer Communications*, 2018.

R. Bardenet, A. Doucet, and C. Holmes. On Markov chain Monte Carlo methods for tall data. *Journal of Machine Learning Research (JMLR)*, 18(47):1–43, 2017.

P. L. Bartlett, P. M. Long, G. Lugosi, and A. Tsigler. Benign overfitting in linear regression. *National Academy of Sciences*, 117(48):30063–30070, 2020.

P. W. Battaglia, J. B. Hamrick, V. Bapst, A. Sanchez-Gonzalez, V. Zambaldi, M. Malinowski, A. Tacchetti, D. Raposo, A. Santoro, R. Faulkner, et al. Relational inductive biases, deep learning, and graph networks. *arXiv preprint arXiv:1806.01261*, 2018.

M. A. Beaumont. Approximate Bayesian computation. *Annual Review of Statistics and Its Application*, 6(1):379–403, 2019.

M. Belkin. Fit without fear: remarkable mathematical phenomena of deep learning through the prism of interpolation. *Acta Numerica*, 30:203–248, 2021.

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

M. Belkin, D. Hsu, S. Ma, and S. Mandal. Reconciling modern machine-learning practice and the classical bias-variance trade-off. *National Academy of Sciences*, 116(32): 15849–15854, 2019.

A. Bellot and M. van der Schaar. Deconfounded score method: Scoring DAGs with dense unobserved confounding. *arXiv preprint arXiv:2103.15106*, 2021.

C. M. Bishop. *Pattern Recognition and Machine Learning*. Springer, 2006.

D. M. Blei, A. Kucukelbir, and J. D. McAuliffe. Variational inference: A review for statisticians. *Journal of the American Statistical Association*, 112(518):859–877, 2017.

P. Bloebaum, D. Janzing, T. Washio, S. Shimizu, and B. Schoelkopf. Cause-effect inference by comparing regression errors. *Artificial Intelligence and Statistics (AISTATS)*, 2018.

A. Bojchevski, O. Shchur, D. Zügner, and S. Günnemann. NetGAN: Generating graphs via random walks. *International Conference on Machine Learning (ICML)*, 2018.

R. J. Bowden and D. A. Turkington. *Instrumental variables*. Cambridge University Press, 1990.

M. M. Bronstein, J. Bruna, Y. LeCun, A. Szlam, and P. Vandergheynst. Geometric deep learning: Going beyond Euclidean data. *IEEE Signal Processing Magazine*, 34 (4):18–42, 2017.

G. Brown, M. Bun, V. Feldman, A. Smith, and K. Talwar. When is memorization of irrelevant training data necessary for high-accuracy learning? *Symposium on Theory of Computing (STOC)*, page 123–132, 2021.

S. Bubeck and M. Sellke. A universal law of robustness via isoperimetry. *Neural Information Processing Systems (NeurIPS)*, 2021.

G. Casella and E. I. George. Explaining the Gibbs sampler. *The American Statistician*, 46(3):167–174, 1992.

S.-H. Chen and E. Ip. Behavior of the Gibbs sampler when conditional distributions are potentially incompatible. *Journal of Statistical Computation and Simulation*, 85: 1–10, 2014.

Y.-L. Chen, L. Minorics, and D. Janzing. Correcting confounding via random selection of background variables. *arXiv preprint arXiv:2202.02150*, 2022.

C. Cheng, J. Duchi, and R. Kuditipudi. Memorize to generalize: on the necessity of interpolation in high dimensional linear regression. *Computational Learning Theory (COLT)*, 2022.

L. Chizat, E. Oyallon, and F. Bach. On lazy training in differentiable programming. *Neural Information Processing Systems (NeurIPS)*, 2019.

S. R. Cook, A. Gelman, and D. B. Rubin. Validation of software for Bayesian models using posterior quantiles. *Journal of Computational and Graphical Statistics*, 15(3): 675–692, 2006.

J. Cornfield, W. Haenszel, E. C. Hammond, A. M. Lilienfeld, M. B. Shimkin, and E. L. Wynder. Smoking and lung cancer: recent evidence and a discussion of some questions*. *International Journal of Epidemiology*, 38(5):1175–1191, 2009.

R. Couillet and Z. Liao. *Random Matrix Methods for Machine Learning*. Cambridge University Press, 2022.

M. Cusumano-Towner and V. K. Mansinghka. Aide: An algorithm for measuring the accuracy of probabilistic inference algorithms. *Neural Information Processing Systems (NeurIPS)*, 2017.

A. Dauber, M. Feder, T. Koren, and R. Livni. Can implicit bias explain generalization? stochastic convex optimization as a case study. *Neural Information Processing Systems (NeurIPS)*, 2020.

E. Daxberger, A. Kristiadi, A. Immer, R. Eschenhagen, M. Bauer, and P. Hennig. Laplace redux-effortless Bayesian deep learning. *Neural Information Processing Systems (NeurIPS)*, 2021.

P. J. Diggle and R. J. Gratton. Monte Carlo methods of inference for implicit statistical models. *Journal of the Royal Statistical Society. Series B (Methodological)*, 46(2): 193–227, 1984.

P. Ding and T. J. VanderWeele. Sensitivity analysis without assumptions. *Epidemiology*, 27(3):368, 2016.

E. Dobriban and S. Wager. High-dimensional asymptotics of prediction: Ridge regression and classification. *The Annals of Statistics*, 2018.

J. Domke. An easy to interpret diagnostic for approximate inference: Symmetric divergence over simulations. *arXiv preprint arXiv:2103.01030*, 2021.

S. Du, J. Lee, H. Li, L. Wang, and X. Zhai. Gradient descent finds global minima of deep neural networks. *International Conference on Machine Learning (ICML)*, 2019.

S. S. Du, W. Hu, and J. D. Lee. Algorithmic regularization in learning deep homogeneous models: Layers are automatically balanced. *Neural Information Processing Systems (NeurIPS)*, 2018.

R. O. Duda, P. E. Hart, and D. G. Stork. *Pattern Classification (2nd Edition)*. Wiley-Interscience, 2000.

J. L. Elman. Finding structure in time. *Cognitive Science*, 14(2):179–211, 1990.

V. Feldman. Does learning require memorization? A short tale about a long tail. *Symposium on Theory of Computing (STOC)*, pages 954–959, 2020.

V. Feldman and C. Zhang. What neural networks memorize and why: Discovering the long tail via influence estimation. *Neural Information Processing Systems (NeurIPS)*, 2020.

L. Fenton. The sum of log-normal probability distributions in scatter transmission systems. *IRE Transactions on Communications Systems*, 8(1):57–67, 1960.

W. D. Flanders and M. J. Khoury. Indirect assessment of confounding: graphic description and limits on effect of adjusting for covariates. *Epidemiology*, pages 239–246, 1990.

K. Fukushima and S. Miyake. Neocognitron: A self-organizing neural network model for a mechanism of visual pattern recognition. *Competition and Cooperation in Neural Nets*, pages 267–285, 1982.

L. A. Gatys, A. S. Ecker, and M. Bethge. A neural algorithm of artistic style. *arXiv preprint arXiv:1508.06576*, 2015.

A. Gelman and D. B. Rubin. Inference from iterative simulation using multiple sequences. *Statistical Science*, 7(4):457–472, 1992.

S. Geman and D. Geman. Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images. *IEEE Transactions on pattern analysis and machine intelligence*, PAMI-6(6):721–741, 1984.

J. Geweke. Getting it right: Joint distribution tests of posterior simulators. *Journal of the American Statistical Association*, 99(467):799–804, 2004.

I. Ghosh and N. Balakrishnan. Study of incompatibility or near compatibility of bivariate discrete conditional probability distributions through divergence measures. *Journal of Statistical Computation and Simulation*, 85(1):117–130, 2015.

D. F. Gleich, L. Zhukov, and P. Berkhin. Fast parallel PageRank: A linear system approach. Technical report, Yahoo! Research Labs, 2004.

I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial nets. *Neural Information Processing Systems (NeurIPS)*, 2014.

J. Gorham and L. Mackey. Measuring sample quality with Stein's method. *Neural Information Processing Systems (NeurIPS)*, 2015.

J. Gorham and L. Mackey. Measuring sample quality with kernels. *International Conference on Machine Learning (ICML)*, 2017.

M. Gori, G. Monfardini, and F. Scarselli. A new model for learning in graph domains. *IEEE International Joint Conference on Neural Networks*, 2:729–734, 2005.

S. Gunasekar, B. E. Woodworth, S. Bhojanapalli, B. Neyshabur, and N. Srebro. Implicit regularization in matrix factorization. *Neural Information Processing Systems (NeurIPS)*, 2017.

S. Gunasekar, J. Lee, D. Soudry, and N. Srebro. Characterizing implicit bias in terms of optimization geometry. *International Conference on Machine Learning (ICML)*, 2018a.

S. Gunasekar, J. D. Lee, D. Soudry, and N. Srebro. Implicit bias of gradient descent on linear convolutional networks. *Neural Information Processing Systems (NeurIPS)*, 2018b.

T. Hastie, A. Montanari, S. Rosset, and R. J. Tibshirani. Surprises in high-dimensional ridgeless least squares interpolation. *The Annals of Statistics*, 50(2):949–986, 2022.

W. K. Hastings. Monte Carlo sampling methods using Markov chains and their applications. *Biometrika*, 57(1):97–109, 1970.

D. Heckerman, D. M. Chickering, C. Meek, R. Rounthwaite, and C. Kadie. Dependency networks for inference, collaborative filtering, and data visualization. *Journal of Machine Learning Research (JMLR)*, 1:49–75, 2001.

C. Heinze-Deml, J. Peters, and N. Meinshausen. Invariant causal prediction for nonlinear models. *Journal of Causal Inference*, 6(2), 2018.

G. E. Hinton and D. van Camp. Keeping the neural networks simple by minimizing the description length of the weights. *Computational Learning Theory (COLT)*, 1993.

S. Hochreiter and J. Schmidhuber. Long short-term memory. *Neural computation*, 1997.

M. D. Hoffman and A. Gelman. The no-u-turn sampler: adaptively setting path lengths in Hamiltonian Monte Carlo. *Journal of Machine Learning Research (JMLR)*, 15(47): 1593–1623, 2014.

M. D. Hoffman, D. M. Blei, C. Wang, and J. Paisley. Stochastic variational inference. *Journal of Machine Learning Research (JMLR)*, 14(4):1303–1347, 2013.

K. Hornik, M. Stinchcombe, and H. White. Multilayer feedforward networks are universal approximators. *Neural networks*, 2(5):359–366, 1989.

P. Hoyer, D. Janzing, J. M. Mooij, J. Peters, and B. Schölkopf. Nonlinear causal discovery with additive noise models. *Neural Information Processing Systems (NeurIPS)*, 2008a.

P. O. Hoyer, S. Shimizu, A. J. Kerminen, and M. Palviainen. Estimation of causal effects using linear non-gaussian causal models with hidden variables. *International Journal of Approximate Reasoning*, 49(2):362–378, 2008b.

J. Huggins, M. Kasprzak, T. Campbell, and T. Broderick. Validated variational inference via practical posterior error bounds. *Artificial Intelligence and Statistics (AISTATS)*, 2020.

R. A. Hughes, I. R. White, S. R. Seaman, J. R. Carpenter, K. Tilling, and J. A. Sterne. Joint modelling rationale for chained equations. *BMC medical research methodology*, 14(1):1–10, 2014.

G. W. Imbens and J. D. Angrist. Identification and estimation of local average treatment effects. *Econometrica*, 62(2):467–475, 1994.

S. Ioffe and C. Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *International Conference on Machine Learning (ICML)*, 2015.

A. Jacot, F. Gabriel, and C. Hongler. Neural tangent kernel: Convergence and generalization in neural networks. *Neural Information Processing Systems (NeurIPS)*, 2018.

S. Jadon. A survey of loss functions for semantic segmentation. *IEEE Conference on Computational Intelligence in Bioinformatics and Computational Biology (CIBCB)*, pages 1–7, 2020.

M. Jagadeesan, I. Razenshteyn, and S. Gunasekar. Inductive bias of multi-channel linear convolutional networks with bounded weight norm. *Computational Learning Theory (COLT)*, 2022.

A. Jalilifard, V. Caridá, A. Mansano, and R. Cristo. Can NetGAN be improved by short random walks originated from dense vertices? *arXiv preprint arXiv:1905.05298*, 2019.

D. Janzing. Causal regularization. *Neural Information Processing Systems (NeurIPS)*, 2019.

D. Janzing and B. Schölkopf. Causal inference using the algorithmic Markov condition. *IEEE Transactions on Information Theory*, 56(10):5168–5194, 2010.

D. Janzing and B. Schölkopf. Detecting confounding in multivariate linear models via spectral analysis. *Journal of Causal Inference*, 2017.

D. Janzing and B. Schölkopf. Detecting non-causal artifacts in multivariate linear regression models. *International Conference on Machine Learning (ICML)*, 2018.

D. Janzing, J. Peters, J. Mooij, and B. Schölkopf. Identifying confounders using additive noise models. *Uncertainty in Artificial Intelligence (UAI)*, pages 249–257, 2009.

D. Janzing, E. Sgouritsa, O. Stegle, J. Peters, and B. Schölkopf. Detecting low-complexity unobserved causes. *Uncertainty in Artificial Intelligence (UAI)*, page 383–391, 2011.

Z. Ji and M. Telgarsky. The implicit bias of gradient descent on nonseparable data. *Computational Learning Theory (COLT)*, 2019.

Y. Jiang, B. Neyshabur, H. Mobahi, D. Krishnan, and S. Bengio. Fantastic generalization measures and where to find them. *International Conference on Learning Representations (ICLR)*, 2020.

M. Jordan, Z. Ghahramani, T. Jaakkola, and L. Saul. An introduction to variational methods for graphical models. *Machine Learning*, 37:183–233, 1999.

C. Joshi and F. Ruggeri. Duality between approximate Bayesian methods and prior robustness. *arXiv preprint arXiv:2004.00796*, 2020.

D. Kaltenpoth and J. Vreeken. We are not your real parents: Telling causal from confounded using MDL. *SIAM International Conference on Data Mining*, pages 199–207, 2019.

A. Kammoun, R. Couillet, J. Najim, and M. Debbah. Performance of capacity inference methods under colored interference. *IEEE Trans. Inf. Theory*, 2011.

Y. Kano, S. Shimizu, et al. Causal inference using nonnormality. *International symposium on science of modeling, the 30th anniversary of the information criterion*, pages 261–270, 2003.

N. S. Keskar, D. Mudigere, J. Nocedal, M. Smelyanskiy, and P. T. P. Tang. On large-batch training for deep learning: Generalization gap and sharp minima. *International Conference on Learning Representations (ICLR)*, 2017.

D. Kingma and J. Ba. Adam: A method for stochastic optimization. *International Conference on Learning Representations*, 2014.

D. Kobak, J. Lomond, and B. 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 (JMLR)*, 21(169):1–16, 2020.

A. Kucukelbir, D. Tran, R. Ranganath, A. Gelman, and D. M. Blei. Automatic differentiation variational inference. *Journal of Machine Learning Research (JMLR)*, 18 (14):1–45, 2017.

K.-L. Kuo and Y. J. Wang. A simple algorithm for checking compatibility among discrete conditional distributions. *Computational Statistics & Data Analysis*, 55(8): 2457–2462, 2011.

K.-L. Kuo and Y. J. Wang. Pseudo-Gibbs sampler for discrete conditional distributions. *Annals of the Institute of Statistical Mathematics*, 71(1):93–105, 2019.

K.-L. Kuo, C.-C. Song, and T. J. Jiang. Exactly and almost compatible joint distributions for high-dimensional discrete conditional distributions. *Journal of Multivariate Analysis*, 157:115–123, 2017.

Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, and L. D. Jackel. Backpropagation Applied to Handwritten Zip Code Recognition. *Neural Computation*, 1(4):541–551, 1989.

Y. LeCun, K. Kavukcuoglu, and C. Farabet. Convolutional networks and applications in vision. *International Symposium on Circuits and Systems*, 2010.

Y. LeCun, Y. Bengio, and G. Hinton. Deep learning. *nature*, 521(7553):436–444, 2015.

O. Ledoit and S. Péché. Eigenvectors of some large sample covariance matrix ensembles. *Probability Theory and Related Fields*, 151(1):233–264, 2011.

J. Lemeire and D. Janzing. Replacing causal faithfulness with algorithmic independence of conditionals. *Minds and Machines*, 23(2):227–249, 2013.

V. Lempitsky, A. Vedaldi, and D. Ulyanov. Deep image prior. *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9446–9454, 2018.

Z. Li, Y. Luo, and K. Lyu. Towards resolving the implicit bias of gradient descent for matrix factorization: Greedy low-rank learning. *International Conference on Learning Representations (ICLR)*, 2021.

Z. C. Lipton. A critical review of recurrent neural networks for sequence learning. *CoRR*, abs/1506.00019, 2015.

F. Liu and L. Chan. Confounder detection in high-dimensional linear models using first moments of spectral measures. *Neural Computation*, 30, 2018.

J. R. Lloyd and Z. Ghahramani. Statistical model criticism using kernel two sample tests. *Neural Information Processing Systems (NeurIPS)*, 2015.

L. Lovász et al. Random walks on graphs: A survey. *Combinatorics, Paul erdos is eighty*, 1993.

D. J. C. MacKay. A practical Bayesian framework for backpropagation networks. *Neural Computation*, 4(3):448–472, 1992.

D. J. C. MacKay. *Information Theory, Inference & Learning Algorithms*. Cambridge University Press, 2002.

A. Marx and J. Vreeken. Telling cause from effect by local and global regression. *Knowledge and Information Systems*, 60(3):1277–1305, 2019.

A. K. McCallum, K. Nigam, J. Rennie, and K. Seymore. Automating the construction of internet portals with machine learning. *Information Retrieval*, 2000.

J. L. McClelland and D. E. Rumelhart. An interactive activation model of context effects in letter perception: I. an account of basic findings. *Psychological review*, 88 (5):375, 1981.

T. P. Minka. Expectation propagation for approximate Bayesian inference. *Uncertainty in Artificial Intelligence (UAI)*, 2001.

M. Molloy and B. Reed. A critical point for random graphs with a given degree sequence. *Random Structures & Algorithms*, 1995.

E. Moroshko, B. E. Woodworth, S. Gunasekar, J. D. Lee, N. Srebro, and D. Soudry. Implicit bias in deep linear classification: Initialization scale vs training accuracy. *Neural Information Processing Systems (NeurIPS)*, 33:22182–22193, 2020.

J. Muré. Optimal compromise between incompatible conditional probability distributions, with application to Objective Bayesian Kriging. *ESAIM: P&S*, 23:271–309, 2019.

V. Muthukumar, K. Vodrahalli, V. Subramanian, and A. Sahai. Harmless interpolation of noisy data in regression. *IEEE Journal on Selected Areas in Information Theory*, 1:67–83, 2020.

D. Navon. Forest before trees: The precedence of global features in visual perception. *Cognitive Psychology*, 9(3):353–383, 1977.

B. Neyshabur, S. Bhojanapalli, D. Mcallester, and N. Srebro. Exploring generalization in deep learning. *Neural Information Processing Systems (NeurIPS)*, 2017.

J. R. Norris and J. R. Norris. *Markov chains*. Cambridge University Press, 1998.

A. Orlitsky. Estimating and computing density based distance metrics. *International Conference on Machine learning (ICML)*, 2005.

J. Pearl. Causal inference in statistics: An overview. *Statistics surveys*, 2009a.

J. Pearl. *Causality: Models, Reasoning and Inference*. Cambridge University Press, 2009b.

J. Peters, P. Bühlmann, and N. Meinshausen. Causal inference by using invariant prediction: identification and confidence intervals. *Journal of the Royal Statistical Society. Series B (Statistical Methodology)*, 78(5):947–1012, 2016.

J. Peters, D. Janzing, and B. Schölkopf. *Elements of causal inference: foundations and learning algorithms*. The MIT Press, 2017.

J. M. Peña. Simple yet sharp sensitivity analysis for unmeasured confounding. *Journal of Causal Inference*, 10(1):1–17, 2022.

D. Phan, N. Pradhan, and M. Jankowiak. Composable effects for flexible and accelerated probabilistic programming in numpyro. *arXiv preprint arXiv:1912.11554*, 2019.

D. Prangle, M. G. B. Blum, G. Popovic, and S. A. Sisson. Diagnostic tools for approximate Bayesian computation using the coverage property. *Australian & New Zealand Journal of Statistics*, 56(4):309–329, 2014.

R. Ranganath, S. Gerrish, and D. Blei. Black box variational inference. *Artificial Intelligence and Statistics (AISTATS)*, 2014.

N. Razin and N. Cohen. Implicit regularization in deep learning may not be explainable by norms. *Neural Information Processing Systems (NeurIPS)*, 2020.

H. Reichenbach. *The direction of time*, volume 65. University of California Press, 1956.

O. E. Richardson. Loss as the inconsistency of a probabilistic dependency graph: Choose your model, not your loss function. *Artificial Intelligence and Statistics (AISTATS)*, 2022.

G. Rodrigues, D. Prangle, and S. Sisson. Recalibration: A post-processing method for approximate Bayesian computation. *Computational Statistics & Data Analysis*, 126: 53–66, 2018.

F. Rosenblatt. Principles of neurodynamics. perceptrons and the theory of brain mechanisms. Technical report, Cornell Aeronautical Lab Inc Buffalo NY, 1961.

R. A. Rossi and N. K. Ahmed. The network data repository with interactive graph analytics and visualization. *Association for the Advancement of Artificial Intelligence*, 2015.

V. Roy. Convergence diagnostics for Markov chain Monte Carlo. *Annual Review of Statistics and Its Application*, 7(1):387–412, 2020.

H. Rue, S. Martino, and N. Chopin. Approximate Bayesian inference for latent Gaussian models by using integrated nested Laplace approximations. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 71(2):319–392, 2009.

H. Rue, A. Riebler, S. H. Sørbye, J. B. Illian, D. P. Simpson, and F. K. Lindgren. Bayesian computing with inla: A review. *Annual Review of Statistics and Its Application*, 4(1):395–421, 2017.

P. Sen, G. Namata, M. Bilgic, L. Getoor, B. Galligher, and T. Eliassi-Rad. Collective classification in network data. *AI magazine*, 2008.

O. Shamir. The implicit bias of benign overfitting. *Computational Learning Theory (COLT)*, 178:448–478, 2022.

C. Shorten and T. M. Khoshgoftaar. A survey on image data augmentation for deep learning. *Journal of big data*, 6(1):1–48, 2019.

J. W. Silverstein and Z. Bai. On the empirical distribution of eigenvalues of a class of large dimensional random matrices. *Journal of Multivariate Analysis*, 54:175–192, 1995.

S. A. Sisson, Y. Fan, and M. Beaumont. *Handbook of approximate Bayesian computation*. CRC Press, 2018.

A. Sjölander. A note on a sensitivity analysis for unmeasured confounding, and the related e-value. *Journal of Causal Inference*, 8(1):229–248, 2020.

D. Soudry, E. Hoffer, M. S. Nacson, S. Gunasekar, and N. Srebro. The implicit bias of gradient descent on separable data. *Journal of Machine Learning Research (JMLR)*, 19(70):1–57, 2018.

D. J. Spiegelhalter and S. L. Lauritzen. Sequential updating of conditional probabilities on directed graphical structures. *Networks*, 20(5):579–605, 1990.

N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov. Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research (JMLR)*, 15(56):1929–1958, 2014.

X. Sun, D. Janzing, and B. Schölkopf. Causal inference by choosing graphs with most plausible Markov kernels. *International Symposium on Artificial Intelligence and Mathematics*, pages 1–11, 2006.

S. Talts, M. Betancourt, D. Simpson, A. Vehtari, and A. Gelman. Validating Bayesian inference algorithms with simulation-based calibration. *arXiv preprint arXiv:1804.06788*, 2018.

J. B. Tenenbaum, V. De Silva, and J. C. Langford. A global geometric framework for nonlinear dimensionality reduction. *science*, 2000.

E. C. Titchmarsh et al. *The theory of functions*. Oxford university press, 1939.

R. E. Turner and M. Sahani. Two problems with variational expectation maximisation for time-series models. *Bayesian Time series models*, pages 109–130, 2011.

S. Van Buuren, J. P. Brand, C. G. Groothuis-Oudshoorn, and D. B. Rubin. Fully conditional specification in multivariate imputation. *Journal of statistical computation and simulation*, 76(12):1049–1064, 2006.

T. J. VanderWeele and O. A. Arah. Bias formulas for sensitivity analysis of unmeasured confounding for general outcomes, treatments, and confounders. *Epidemiology*, pages 42–52, 2011.

T. J. VanderWeele and P. Ding. Sensitivity analysis in observational research: introducing the e-value. *Annals of internal medicine*, 167(4):268–274, 2017.

T. J. VanderWeele, P. Ding, and M. Mathur. Technical considerations in the use of the e-value. *Journal of Causal Inference*, 7(2):20180007, 2019.

L. C. Vankadara, L. Rendsburg, U. von Luxburg, and D. Ghoshdastidar. Interpolation and regularization for causal learning. *arXiv preprint arXiv:2202.09054*, 2022.

V. Vapnik. *The nature of statistical learning theory*. Springer science & business media, 1999.

G. Vardi and O. Shamir. Implicit regularization in relu networks with the square loss. *Computational Learning Theory (COLT)*, 2021.

A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. u. Kaiser, and I. Polosukhin. Attention is all you need. *Neural Information Processing Systems (NeurIPS)*, 2017.

Q. Wang, Y. Ma, K. Zhao, and Y. Tian. A comprehensive survey of loss functions in machine learning. *Annals of Data Science*, 9(2):187–212, 2022.

D. H. Wolpert. *The Mathematics of Generalization: The Proceedings of the SFI/CNLS Workshop on Formal Approaches to Supervised Learning*. Perseus Publishing, 1994.

B. Woodworth, S. Gunasekar, J. D. Lee, E. Moroshko, P. Savarese, I. Golan, D. Soudry, and N. Srebro. Kernel and rich regimes in overparametrized models. *Computational Learning Theory (COLT)*, 125:3635–3673, 2020.

Z. Wu, S. Pan, F. Chen, G. Long, C. Zhang, and P. S. Yu. A comprehensive survey on graph neural networks. *Neural Networks and Learning Systems*, 32(1):4–24, 2021.

A. J. Wyner, M. Olson, J. Bleich, and D. Mease. Explaining the success of adaboost and random forests as interpolating classifiers. *Journal of Machine Learning Research (JMLR)*, 18(48):1–33, 2017.

H. Xing, G. Nicholls, and J. (Kate) Lee. Distortion estimates for approximate Bayesian inference. *Uncertainty in Artificial Intelligence (UAI)*, 2020.

Y. Yao, A. Vehtari, D. Simpson, and A. Gelman. Yes, but did it work?: Evaluating variational inference. *International Conference on Machine Learning (ICML)*, 2018.

X. Yu, D. J. Nott, M.-N. Tran, and N. Klein. Assessment and adjustment of approximate inference algorithms using the law of total variance. *Journal of Computational and Graphical Statistics*, 0(0):1–14, 2021.

C. Yun, S. Krishnan, and H. Mobahi. A unifying view on implicit bias in training linear neural networks. *International Conference on Learning Representations (ICLR)*, 2021.

C. Zhang, S. Bengio, M. Hardt, B. Recht, and O. Vinyals. Understanding deep learning requires rethinking generalization. *International Conference on Learning Representations (ICLR)*, 2017.

C. Zhang, J. Butepage, H. Kjellstrom, and S. Mandt. Advances in variational inference. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, 41(08):2008–2026, 2019.

K. Zhang and A. Hyvärinen. On the identifiability of the post-nonlinear causal model. *Uncertainty in Artificial Intelligence (UAI)*, 2009.

Y. Zhou, H. Cheng, and J. X. Yu. Graph clustering based on structural/attribute similarities. *VLDB Endowment*, 2009.