Uncertainties of Latent Representations in Computer Vision

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

vorgelegt von
Michael Kirchhof
aus Hilden

Tübingen
2024

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

Tag der mündlichen Qualifikation:	17.06.2024
Dekan:	Prof. Dr. Thilo Stehle
1. Berichterstatter/-in:	Prof. Dr. Enkelejda Kasneci
2. Berichterstatter/-in:	Prof. Dr. Philipp Hennig

To whom it may concern.

Abstract

Uncertainty quantification is a key pillar of trustworthy machine learning. It enables safe reactions under unsafe inputs, like predicting only when the machine learning model detects sufficient evidence, discarding anomalous data, or emitting warnings when an error is likely to be inbound. This is particularly crucial in safety-critical areas like medical image classification or self-driving cars. Despite the plethora of proposed uncertainty quantification methods achieving increasingly higher scores on performance benchmarks, uncertainty estimates are often shied away from in practice. Many machine learning projects start from pretrained latent representations that come without uncertainty estimates. Uncertainties would need to be trained by practitioners on their own, which is notoriously difficult and resource-intense.

This thesis makes uncertainty estimates easily accessible by adding them to the latent representation vectors of pretrained computer vision models. Besides proposing approaches rooted in probability and decision theory, such as Monte-Carlo InfoNCE (MCInfoNCE) and loss prediction, we delve into both theoretical and empirical questions. We show that these unobservable uncertainties about unobservable latent representations are indeed provably correct. We also provide an uncertainty-aware representation learning (URL) benchmark to compare these unobservables against observable ground-truths. Finally, we compile our findings to pretrain lightweight representation uncertainties on large-scale computer vision models that transfer to unseen datasets in a zero-shot manner.

Our findings do not only advance the current theoretical understanding of uncertainties over latent variables, but also facilitate the access to uncertainty quantification for future researchers inside and outside the field. As downloadable starting points, our pretrained representation uncertainties enable a range of novel practical tasks for straightforward but trustworthy machine learning.

Zusammenfassung

Die Quantifizierung von Unsicherheiten ist ein Grundpfeiler des vertrauenswürdigen maschinellen Lernens. Sie ermöglicht sichere Reaktionen bei unsicheren Eingaben, wie etwa Vorhersagen nur dann zu treffen wenn ein künstlich intelligentes Modell genügend Anhaltspunkte findet, anomale Daten zu filtern oder Warnungen auszugeben wenn ein Fehler wahrscheinlich ist. Dies ist besonders in sicherheitskritischen Bereichen wie der Klassifizierung medizinischer Bilder oder bei selbstfahrenden Autos wichtig. Trotz der Fülle an publizierten Methoden zur Quantifizierung von Unsicherheiten, die in numerischen Vergleichen immer bessere Ergebnisse erzielen, werden Unsicherheitsschätzungen in der Praxis oft gescheut. Viele Projekte des maschinellen Lernens starten mit vortrainierten latenten Repräsentationen, die von sich aus keine Unsicherheitsschätzungen beinhalten. Die Unsicherheiten müssten von Anwendern selbst trainiert werden, was jedoch als kompliziert und ressourcenintensiv angesehen wird.

In dieser Doktorarbeit werden Unsicherheitsschätzer leichter zugänglich gemacht, indem sie zu den latenten Repräsentationsvektoren von vortrainierten Modellen für die Verarbeitung von Bilddaten hinzugefügt werden. Wir entwickeln Ansätze aus der Wahrscheinlichkeits- und Entscheidungstheorie, wie Monte-Carlo InfoNCE (MCInfoNCE) und die Schätzung von Vorhersagefehlern, und befassen uns sowohl mit mathematischen als auch empirischen Aspekten des Problems. Wir zeigen, dass diese unbeobachtbaren Unsicherheiten über unbeobachtbare latente Repräsentationen tatsächlich beweisbar korrekt sind. Wir stellen außerdem einen numerischen Leistungstest für Unsicherheiten über latente Repräsentationen vor (URL), um diese unbeobachtbaren Schätzungen mit beobachtbaren Vergleichswerten abzugleichen. Schließlich bündeln wir unsere Ergebnisse, um kostengünstige Unsicherheitsschätzer für die latenten Repräsentationen großer Modelle des computergestützten Sehens vorzutrainieren, die ohne weiteres Training auf neuen Datensätzen funktionieren.

Unsere Ergebnisse erweitern nicht nur das aktuelle theoretische Verständnis von Unsicherheiten über latente Variablen, sondern erleichtern auch den Zugang zur Quantifizierung von Unsicherheiten für zukünftige Forschung innerhalb und außerhalb des Feldes. Als herunterladbare Ausgangspunkte ermöglichen unsere vortrainierten Repräsentationsunsicherheiten eine Reihe neuartiger praktischer Anwendungen für unkompliziertes, aber vertrauenswürdiges maschinelles Lernen.

Acknowledgements

While the reader may rightfully consider this thesis a document, to me it resembles a chapter of my life. Though words alone can barely express my gratitude, I want to dedicate this page to those who accompanied me through the past three years.

First, I want to thank my supervisors Enkelejda Kasneci and Seong Joon Oh. You gave me the freedom to explore my interests and guided them into well-rounded projects, with an eye for both what's theoretically intriguing and what's practically needed. I want to thank you for your mentorship beyond science, shaping me as the person I now am. All the schools, conferences, and retreats that you let me discover gave me exciting insights into the worldwide machine learning community in these exciting times. In these terms, I would like to thank the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) for funding me and my projects under Germany's Excellence Strategy – EXC number 2064/1 – Project number 390727645. I also want to thank my third thesis advisory committee member, Wieland Brendel. Your rich feedback and deep exchange about my papers far exceeded what I hoped for. Finally, I want to thank Philipp Hennig for volunteering his time to be the independent reviewer of this thesis.

I thank my co-authors Bálint Mucsányi, Karsten Roth, Mark Collier, Tobias Leemann, Yao Rong, Elisa Nguyen, Alexander Rubinstein, and Gjergji Kasneci. I very much enjoyed the exchange and discussions, which defined what science feels like to me. I am also grateful to Bálint Mucsányi and Patrick Köhler for allowing me to supervise their master's theses. I hope I could teach you even just a fraction of what you taught me.

As evidenced by the anecdotes throughout the thesis, Tübingen was a place of collaboration to me. I particularly want to thank the International Max Planck Research School for Intelligent Systems (IMPRS-IS) for all the community events and retreats that gave space for casual conversations with hundreds of Ph.D. students working on cutting-edge papers across the whole spectrum of machine learning. This is what I dreamed of when I applied to the University of Tübingen.

Last, but most importantly, I would like to thank those that gave me joy in the best times and strength in the toughest times of my life. Above all, these are my parents Gabi Kirchhof and Harry Kirchhof and my brother Christian Kirchhof. Your unconditional support is an unwavering anchor in my life. This also holds for the best companions I could think of for this journey: Kajetan Wojtowicz, Lisa Haxel, Rachel Rapp, Torben Rapp, Patrik Reizinger, Frederic Bootz, and Florian Grieskamp. Thank you for the countless game evenings, daring outdoor adventures, and deep talks.

This document is not long enough to list all those that had an impact on me. Freely quoting Viktor Frankl, we radiate into our effects on others.[1] Rest assured that I will forever embody your effects – figuratively for your practiced values, and quite literally for all your delicious food!

[1] *Man's Search for Meaning*, Viktor Frankl, 1946, Beacon Press.

Contents

1 Introduction 13

1.1 Of Penguins and Uncertainties . . . 13

1.2 Related Work . . . 14

1.3 Research Questions . . . 18

1.4 Contributions and Outline . . . 18

1.5 List of Publications . . . 19

2 Probabilistic Representation Learning 21

2.1 Prologue . . . 21

2.2 Motivation . . . 21

2.3 Methods . . . 22

2.4 Core Results . . . 23

2.5 Discussion . . . 24

3 Probabilistic Embeddings are Provably Correct 25

3.1 Prologue . . . 25

3.2 Motivation . . . 25

3.3 Methods . . . 26

3.4 Core Results . . . 27

3.5 Discussion . . . 28

4 The URL Benchmark for Representation Uncertainties 31

4.1 Prologue . . . 31

4.2 Motivation . . . 31

4.3 Methods . . . 32

4.4 Core Results . . . 33

4.5 Discussion . . . 35

5 Pretrained Representation Uncertainties 37

5.1 Prologue . . . 37

5.2 Motivation . . . 37

5.3 Methods . . . 38

5.4 Core Results . . . 39

5.5 Discussion . . . 40

6 Discussion 43

6.1 Applications . . . 43

6.2 Specialized Uncertainties 46
6.3 Conclusion 47

Abbreviations **49**

List of Figures **51**

Bibliography **53**

A A non-isotropic probabilistic take on proxy-based deep metric learning **63**

B Probabilistic contrastive learning recovers the correct aleatoric uncertainty of ambiguous inputs **95**

C URL: A Representation Learning Benchmark for Transferable Uncertainty Estimates **117**

D Pretrained Visual Uncertainties **143**

Introduction 1

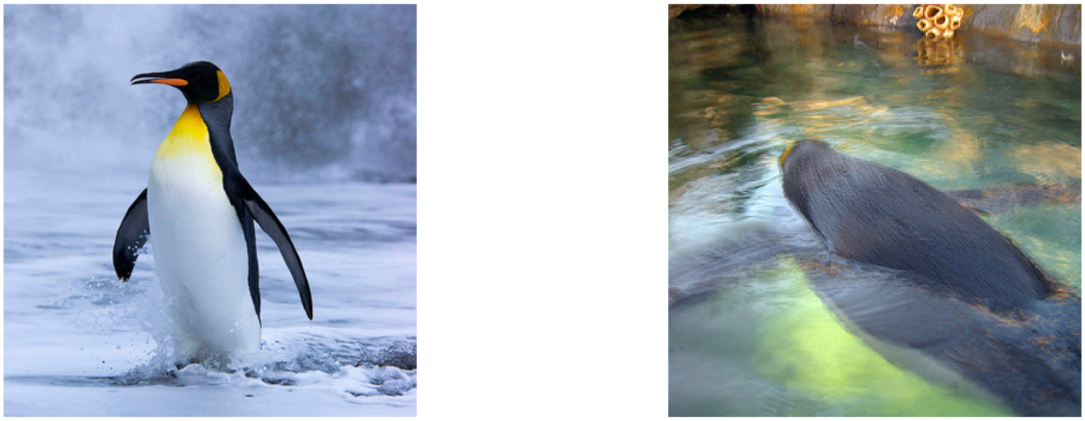
1.1 Of Penguins and Uncertainties

Any human will recognize the image in Figure 1a as a penguin and store it accordingly in their mental map of animals. Now consider Figure 1b. It could show a penguin, but also a seal, or maybe a beaver. The picture itself is uncertain, that is, it does not contain enough information to infer what it shows. The best any human can do is to store it somewhere in the region of aquatic animals in their mental map, flagged with a question mark.

Figure 1b is no exception and the problem can not be trained away, whether we deploy a computer vision model or a human expert. Beyer et al. (2020) show that humans disagree about the class of 29.9% of the images in the popular ImageNet-1k benchmark (Deng et al., 2009). This is even more pronounced when images are no high-quality photographs from the internet but automatically taken by magnetic resonance imaging scanners or surveillance cameras on animal farms, as studied by Schmarje et al. (2022). Even their best real-world dataset with only four classes to choose from has a disagreement rate of 92.2%. Vision inherently is and will remain ambiguous.

Current deep encoders in computer vision also have mental maps, their embedding spaces, where they store what they detect in images as representation vectors. But they lack the ability to express their uncertainty: Although Figure 1b is much more ambiguous than Figure 1a, both will be pinpointed to an exact representation vector in the embedding space. Any module that further processes these representations, for retrieving similar images or predicting the animal species, has no more sense of the ambiguity.

This thesis adds uncertainty estimates to representation vectors. This enhancement may seem nuanced at first glance, but it conceals an iceberg of challenges beneath the surface: Representations are latent variables, meaning that they are not observable in the real world and a computer vision model has to find them itself. Adding uncertainties, which are also not observable in the real world, to such already unobservable latent variables makes the problem even more complicated. And more still, we also have to compare these unobservables about unobservables against some notion of observable ground-truth in the real world to ensure their quality and good performance, which is a territory uncharted by state-of-the-art uncertainty benchmarks. But the reward of solving these challenges is high: Uncertainties added directly to representations will trickle down to all applications that start off from representations, e.g., of pretrained models.



(a) A clear image of a penguin.

(b) An ambiguous image of a penguin.

Figure 1: Images can be inherently ambiguous, making it necessary to quantify their uncertainty. Both images are from the ImageNet-1k benchmark dataset (Deng et al., 2009).

Before diving into details, we give a teaser of our results in Figure 2. This is the embedding space of six dog breeds, where more uncertain representations are larger and more transparent. The plot makes it easy to understand which images are more ambiguous and likely to be misclassified because they reveal too few information. This enables computer vision models to automatically treat uncertain images differently, e.g., by abstaining from classifying them until a user inputs a more clear image, enabling a more trustworthy deployment of machine learning models in practice (Mucsányi et al., 2023). Notably, these uncertainties are provided zero-shot. The model was trained on a different dataset and the uncertainties are generalized from the large pretraining corpora we scaled our approach to. This means that practitioners can make direct use of the findings of this thesis by downloading our pretrained uncertainty models and running them on their data.

Let us now return to the start of our expedition, formally defining representation learning and uncertainties in computer vision as well as our precise research questions, which culminated in the development of these general-purpose representation uncertainties.

1.2 Related Work

1.2.1 Pretrained Representations

We first establish notation that will reoccur throughout the thesis. Let x denote an input from the input space $\mathcal{X}$ and y an output from the output space $\mathcal{Y}$. The task of any machine learning model is to fit a function $f : \mathcal{X} \rightarrow \mathcal{Y}$ that predicts y from x. In computer vision, these are commonly an image x and a class label y. To predict a label from an unstructured information source like an image, modern deep learning architectures first

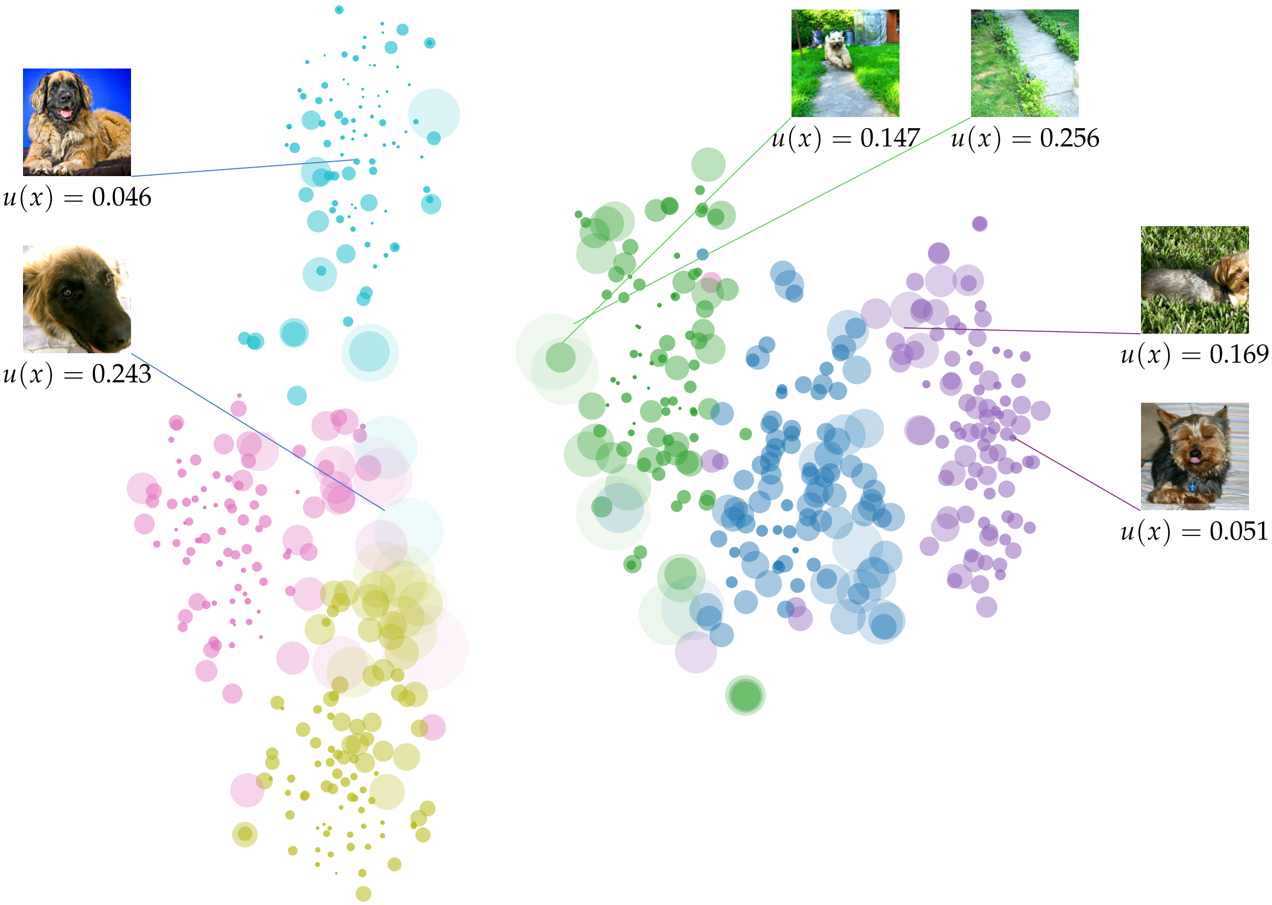


Figure 2: Each point is the representation of an Oxford Pets image (Parkhi et al., 2012). The size of each dot visualizes the uncertainty $u(x)$ of the representation, calculated by our approach in Chapter 5. This makes it easier to detect images that are naturally ambiguous (large and transparent). Figure cited from the original paper (Kirchhof et al., 2024).

extract features from the image. That is, they have an encoder component $e : \mathcal{X} \to \mathcal{Z}$ and a subsequent classifier $c : \mathcal{Z} \to \mathcal{Y}$ with $f = c \circ e$. The goal of representation learning (Bengio et al., 2013) is to learn a versatile encoder e such that the representations z in the embedding spaces $\mathcal{Z}$ in the middle represent the content of the image in an abstract manner, as a high dimensional latent vector.

Since training the encoder is often the most time and data consuming part of training a deep learning architecture, the representations are often trained in advance on large image datasets with versatile classes to learn diverse features. Most modern computer vision projects start off from such pretrained models. Pretrained representations enable, e.g., to search for semantically similar images, called retrieval (Chang and Fu, 1979; Liu et al., 2021; Jush et al., 2023; Douze et al., 2024), or to re-use the encoder and learn new tasks quicker by finetuning on a small dataset in a few-shot or even zero-shot manner (Weiss et al., 2016; Goyal et al., 2023; Ramesh et al., 2021).

The key to well usable pretrained representations is that semantically similar images should lay close to one another in the embedding space. Contrastive learning approaches (Chopra et al., 2005; Schroff et al., 2015; Hadsell et al., 2006; Chen et al., 2020; Grill et al., 2020; Radford et al., 2021; Chen and He, 2021) formulate this directly as an objective function when training the encoder. As an example, tuples of images cropped from the same underlying image are encouraged to be close to one another (Hadsell et al., 2006) or they are encouraged to be close to one another and far from other images (Schroff et al., 2015). Besides these self-supervised approaches, modern pretraining approaches also return to traditional supervised learning, where the supervision signal that tells if two images are similar is whether they have the same class label. The key here is that the class labels are diverse enough, such as on the ImageNet-1k dataset (Deng et al., 2009) that comprises 1.2 million images of 1000 classes or on ImageNet-21k (Deng et al., 2009) that comprises 14.2 million images of 21.8 thousand classes.

These advances in pretrained representations reveal important desiderata for our desired representation uncertainties: If representation uncertainties are output along with pretrained representations, we need to pretrain them on similar scales. We also need to ensure that our representation uncertainties capture uncertainties of the general image content that the representation summarizes, not just uncertainty in terms of the (pre-)training task. With these properties in mind, let us review the current state uncertainty estimation approaches in computer vision.

1.2.2 Large-scale Uncertainties in Computer Vision

Uncertainty estimation adds a second task to the model. In addition to the estimate for y it also has to output an uncertainty estimate $u(x)$, $u : \mathcal{X} \to \mathcal{U}$, sometimes in the form of a probability $\mathcal{U} = [0, 1]$ or more generally any scalar value with $\mathcal{U} \subseteq \mathbb{R}$. These uncertainty estimates are a key prerequisite to deploy models in safety-critical areas like medical imaging or self-driving cars (Gulshan et al., 2016; Carannante et al., 2021; Kurz et al., 2022; Franchi et al., 2022) where we want to predict only if we are certain. Uncertainties are also fundamental ingredients of anomaly detection (Chalapathy and Chawla, 2019) and active learning (Settles, 2009; Nguyen et al., 2022).

First attempts to bring uncertainty into deep learning and computer vision stem from Bayesian roots (Bernardo and Smith, 2009). A prominent example is the Laplace approximation (Mackay, 1992) that approximates a Gaussian around the network's parameters. This allows to sample multiple output vectors per input, which can be processed into scalar uncertainty estimates $u(x)$. The issue is that these approximate Bayesian approaches are hard to scale to deep architectures, with work on scaling going on to this date (Ritter et al., 2018; Daxberger et al., 2021; Deng et al., 2022). Later works thus yield these samples more directly: Deep ensembles (Lakshminarayanan et al., 2017) train multiple networks to output multiple vectors, and Gal and Ghahramani (2016) activate random Dropout

(Srivastava et al., 2014) at inference time to produce multiple slightly different outputs. These approaches are scalable to arbitrarily deep architectures, but their runtime (and memory) costs still scale linearly in the number of samples. Anyhow, they mark the start of a trend: Reducing the computational hurdle of uncertainty estimation to enable a widespread application in practice.

A recent step towards low-cost uncertainties is to output uncertainty estimates $u(x)$ directly during the forward pass of the model, so called deterministic methods (Postels et al., 2022; Haußmann et al., 2020). This is implemented by adding modules to the architecture that output specialized uncertainties $u(x)$. These modules are trained for specialized tasks (Mucsányi et al., 2024). For example, to detect out-of-distribution inputs (Galil et al., 2023a), deterministic methods estimate the data density in the model's latent space (Lee et al., 2018; Van Amersfoort et al., 2020; Mukhoti et al., 2023). To estimate correctness of prediction, they predict the model's own loss at any given sample (Yoo and Kweon, 2019; Cui et al., 2023; Lahlou et al., 2023; Laves et al., 2020). Such specialized approaches are also relevant for the recent strive for decomposed or disentangled uncertainties (Wimmer et al., 2023; Bengs et al., 2023; Gruber et al., 2023; Valdenegro-Toro and Mori, 2022; Depeweg et al., 2018). We contribute to these disentanglement efforts in Chapter 5.

So how could one build a specialized uncertainty estimate for representations? One strain of research are probabilistic embeddings (Oh et al., 2019; Collier et al., 2023; Kim et al., 2023; Nakamura et al., 2023). They add an auxiliary output head that estimates a variance parameter for each representation, resulting in a distribution over all possible latents that an input could show. While probabilistic embeddings have shown increased performance (Karpukhin et al., 2022) and qualitatively sensible outputs (Oh et al., 2019; Scott et al., 2019), their theoretical underpinning and evaluation metrics are still in their infancy. We contribute to these efforts in Chapters 2 to 4.

The last gap to bridge uncertainty estimates with pretrained representations is their transferability. Initial works (Guo et al., 2017) but also current large-scale undertakings (Dehghani et al., 2023) often consider calibration only on the dataset the model was trained on. The largest distribution shifts that uncertainties are evaluated on are corrupted versions of the train dataset (Ovadia et al., 2019; Tran et al., 2022). At a certain level of corruption, let alone on new datasets with new classes, images are commonly considered out-of-distribution, so that the evaluation protocol is just to achieve high uncertainties on these samples compared to in-distribution ones (Park et al., 2023; Galil et al., 2023a; Postels et al., 2022; Ovadia et al., 2019). This is a reasonable goal, but we want to take the generalization of uncertainty estimators one step further. Similar to representation learning, we want uncertainties to work within completely new datasets, where uncertainties should not be generally high but differ between the unseen samples. Also similar to representation learning, this requires novel benchmarking metrics, see Chapter 4, and large-scale pretrained models developed along these metrics, see Chapter 5.

1.3 Research Questions

The main goal of this thesis is to provide uncertainty estimates along with representation vectors, merging the current enhancements in representation learning with those in uncertainty estimation. As we have outlined before, this is intended to make uncertainties transferable to new datasets and tasks and thus easier to use. To this end, we need to develop new approaches, gain a theoretical understanding about representation uncertainties and how to benchmark them, and finally scale our findings to match current pretrained representation models. We take on these challenges one by one in the following chapters, guided by the following research questions (RQs).

RQ1. Which methods can provide uncertainties about representations?

RQ2. What do these uncertainties reflect theoretically or in the real-world?

RQ3. How to quantify how good an uncertainty about a representation is?

RQ4. Can we scale the uncertainties to large models and large pretraining corpora?

1.4 Contributions and Outline

We answer these questions in four chapters, each resembling one paper. We summarize their main findings and contributions to the scientific community below.

Chapter 2 takes on RQ1, providing first methods that add uncertainties to representations in a subfield of representation learning, deep metric learning. In particular, we utilize the probabilistic embeddings framework where we predict a mean and a variance for each representation. We expand these works by implementing distribution-to-distribution distance functions and generalizing previous distributions to more flexible covariance structures. These enhancements make deep metric learning probabilistic and we find that they increase the performance. We take a first look at RQ2 to analyze why the uncertainties lead to higher performance.

Chapter 3 provides deeper theoretical insights into RQ2. We derive a theoretical framework to define representation uncertainties formally and to investigate in which sense they are correct. We find that they can be seen as the posteriors of a lossy image-generating process, generalizing previous theoretical results of nonlinear independent component analysis (Zimmermann et al., 2021; Reizinger et al., 2022). We also provide a new method to learn representation uncertainties in self-supervised, adding to RQ1. We verify that its representation uncertainties are indeed correct in the theoretical sense and, in the practical sense of RQ3, correlated with human uncertainties.

Chapter 4 provides the community with the first benchmark to measure correctness of representation uncertainties at scale and under distribution shifts, answering RQ3. This allows representation learning researchers to enhance their benchmarking code with metrics for uncertainties in four lines of code. We verify our benchmark with comparisons to the human and other real-world uncertainties studied in the previous chapter to broaden our understanding of RQ2. We reimplement both of our approaches from RQ1 and add multiple ones from literature. We scale them all to ImageNet-1k and use a pretraining-like train and test framework, paving the way for RQ4.

Chapter 5 uses this benchmark to develop pretrained representation uncertainties at the scale of ImageNet-21k and large Vision Transformer, aiming for RQ4. This compiles the findings of all previous chapters and research questions into one downloadable model for future researchers. In the process, we fix a gradient conflict that deteriorated the performance of deterministic uncertainty approaches in the literature. We also study the behaviour of our uncertainties and find that they provide aleatoric uncertainties devoid from epistemic uncertainties, contributing to RQ2 and to the recent efforts in uncertainty disentanglement (Wimmer et al., 2023; Mucsányi et al., 2024).

Finally, in Chapter 6 we outline the applications that representation uncertainties enable and discuss how our findings shape future directions in uncertainty quantification.

1.5 List of Publications

1.5.1 Publications Relevant to this Thesis

This thesis comprises four main publications that I have published as first author in the last three years. All papers are summarized and discussed in their corresponding chapters in the main text, and appended to the thesis in their full form.

Michael Kirchhof, Karsten Roth, Zeynep Akata, and Enkelejda Kasneci. A non-isotropic probabilistic take on proxy-based deep metric learning. *European Conference on Computer Vision (ECCV)*, 2022.

Michael Kirchhof, Enkelejda Kasneci, and Seong Joon Oh. Probabilistic contrastive learning recovers the correct aleatoric uncertainty of ambiguous inputs. *International Conference on Machine Learning (ICML)*, 2023.

Michael Kirchhof, Bálint Mucsányi, Seong Joon Oh, and Enkelejda Kasneci. URL: A representation learning benchmark for transferable uncertainty estimates. *Neural Information Processing Systems Track on Datasets and Benchmarks (NeurIPS D&B)*, 2023.

Michael Kirchhof, Mark Collier, Seong Joon Oh, and Enkelejda Kasneci. Pretrained visual uncertainties. arXiv preprint arXiv:2402.16569, 2024. Under submission.

1.5.2 Further Publications

Besides these main works, I have been involved in three major side projects during my time as a Ph.D. student, all centered around representations and uncertainty.

> Tobias Leemann, **Michael Kirchhof**, Yao Rong, Enkelejda Kasneci, and Gjergji Kasneci. When are post-hoc conceptual explanations identifiable? *Uncertainty in Artificial Intelligence (UAI)*, 2023.
>
> Bálint Mucsányi, **Michael Kirchhof**, Elisa Nguyen, Alexander Rubinstein, and Seong Joon Oh. Trustworthy machine learning. arXiv preprint arXiv:2310.08215, 2023.
>
> Bálint Mucsányi, **Michael Kirchhof**, and Seong Joon Oh. Benchmarking uncertainty disentanglement: Specialized uncertainties for specialized tasks. arXiv preprint arXiv:2402.19460, 2024. Under submission.

Probabilistic Representation Learning 2

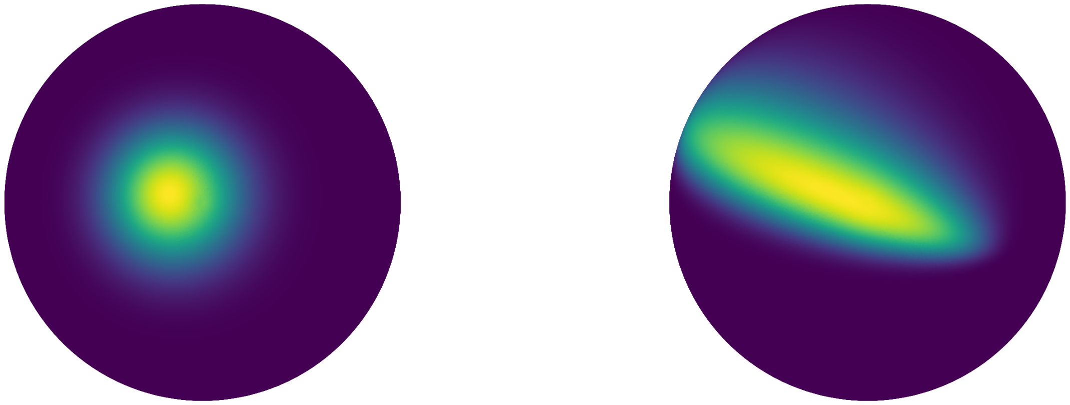
Michael Kirchhof, Karsten Roth, Zeynep Akata, and Enkelejda Kasneci. A non-isotropic probabilistic take on proxy-based deep metric learning. *European Conference on Computer Vision (ECCV)*, 2022.

2.1 Prologue

"That probabilistic approach seems to work, I'm right now at 66% Recall@1 on CUB.", I wrote to Karsten Roth, at that time a Ph.D. student specializing in representation learning. I had just sent him an initial implementation of what happens if we change vector representations to distributional or probabilistic ones. What had happened was that it outperformed most baselines in his recent benchmark. He responded within seconds, with a scientist's mixture of excitation and caution, "I'd have time for a meeting later today? Let's just double check the code.". The code was fine and by evening we had set up our collaboration. In combining our strenghts, we expanded the initial idea mathematically and gained more evidence empirically. We presented the results at a computer vision conference, ECCV, that would have impacts on the next chapters of this dissertation.

2.2 Motivation

We develop our uncertainties for representations from classical representation learning. The goal here is to encode images into vectors, such that images of, e.g., the same class, have similar vector representations. This is a mandatory property for retrieval systems (Sohn, 2016; Brattoli et al., 2020; Douze et al., 2024). One sub-field of this is deep metric learning (Roth et al., 2020). It investigates which distance function between the representations one should use to train the encoder. We find that simple ones like cosine distance do not account for uncertainties in the images, despite the field having argued that this was an intended feature to ensure all images were treated the same (Ranjan et al., 2017). We, along with concurrent works (Scott et al., 2021), question this and argue that uncertainties are informative features that support the training. In this chapter, we represent images as distributions over possible latents instead of single vectors, so called probabilistic embeddings. We show how to calculate distances between them, and how much and why this improves deep metric learning.



(a) von Mises Fisher (vMF)

(b) non-isotropic von Mises-Fisher (nivMF)

Figure 3: Densities of a vMF and a non-isotropic vMF distributions on a three-dimensional unit-sphere. Purple is a low and yellow a high density. Figure adapted from the original paper (Kirchhof et al., 2022).

2.3 Methods

The goal of deep metric learning is to learn representations $e(x)$ for each image such that similar images are placed close to one another and dissimilar ones far from another in a model's representation space $\mathcal{Z}$. Similarity is usually defined as belonging to the same class or being two crops from the same image. These representations are learned by loss functions that measure the distance between representations and push similar ones closer to one another and dissimilar ones away from each other. To reduce the noise in this process, ProxyNCA and ProxyNCA++ (Movshovitz-Attias et al., 2017; Teh et al., 2020) propose to use proxies for each class, so that each image is pushed closer to the proxy of its class. The contrastive loss function is

$$\mathcal{L}_{\text{NCA++}} = \log \frac{\exp\left(s\left(\frac{e(x)}{\|e(x)\|}, \mathrm{p}^*\right)/t\right)}{\sum_{c=1}^{C} \exp\left(s\left(\frac{e(x)}{\|e(x)\|}, \mathrm{p}_c\right)/t\right)}, \tag{2.1}$$

where p^* is the true proxy (i.e., class) of x, $t > 0$ is a temperature, and $p_c, c = 1, \dots, C$, are all C possible classes. In practice, the representations $e(x)$ are often normalized to unit length, so the representation space $\mathcal{Z}$ is a unit sphere and the similarity function s is a cosine similarity.

Our key idea is to allow uncertainties about what an image represents, e.g., if it is blurry or an information-losing crop. For this, we represent images as distributions $\zeta(x)$ over all possible latents, so called probabilistic embeddings. In particular, we use von Mises-Fisher (vMF) distributions (Fisher, 1953; Mardia and Jupp, 2009) over the unit-sphere $\mathcal{Z}$, as

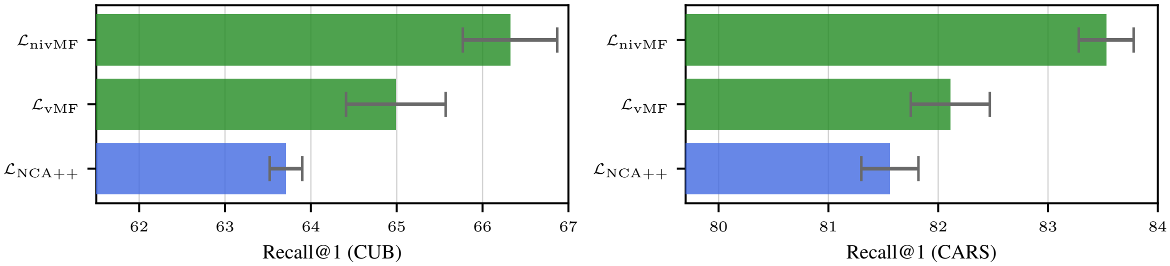


Figure 4: Probabilistic embeddings (green) lead to better retrieval performance than deterministic ones (blue). Bars show the standard deviation across five seeds. Figure adapted from the original paper (Kirchhof et al., 2022).

shown in Figure 3. For proxies, we develop non-isotropic von Mises-Fisher (nivMF) distributions ρ. They allow class distributions to have non-unit covariances. To measure the distribution-to-distribution similarity between probabilistic embeddings and proxies, we use the expected likelihood kernel (ELK, Jebara and Kondor, 2003). These changes result in the uncertainty-aware contrastive loss function

$$\mathcal{L}_{\text{nivMF}} = \log \frac{\exp(s(\zeta(x), \rho^*)/t)}{\sum_{c=1}^{C} \exp(s(\zeta(x), \rho_c)/t)}. \tag{2.2}$$

We implement this by using auxiliary learnable variables for the proxy means and covariances. For the probabilistic embeddings of images, we use the typical (normalized) representations $\frac{e(x)}{\|e(x)\|}$ as the mean value of the $\zeta(x)$ distribution. The concentration (inverse variance) parameter is set to the pre-normalization representation norms, i.e., $\|e(x)\|$, following Li et al. (2021). This utilizes that the representation norm is empirically related to certainty, namely how many class characteristic features can be detected in an image. We dicuss this further in the main paper.

2.4 Core Results

2.4.1 Probabilistic Embeddings Improve Retrieval Performance

The primary objective of deep metric learning is to learn a well-structured embedding space. Representations of similar images should be close to one another, enabling retrieval. This is measured via the Recall@1: If we compute mean representations for each image in the test dataset, how often is the nearest neighbor of each representation in the same class? This percentage is computed on the dataset the model was trained on, but on a withheld set of classes. This induces a small domain shift to ensure the representations generalize beyond the training classes.

Figure 4 shows that training probabilistic embeddings with $\mathcal{L}_{\text{nivMF}}$ leads to a higher Recall@1 than the vector representations of $\mathcal{L}_{\text{NCA++}}$ on two representation learning benchmark datasets, CUB-200 (Wah et al., 2011) and CARS-196 (Krause et al., 2013). As a step in-between, $\mathcal{L}_{\text{vMF}}$ uses vMF distributions for both images and classes, showing that both enhancements, probabilistic embeddings and non-isotropy, increase retrieval performance. The main paper presents similar advantages when adding probabilistic embeddings to more complicated contrastive loss functions.

2.4.2 Uncertainties Re-weight the Gradients

The above experiment does not use the learned uncertainties during testing. It only measures the Recall@1, i.e., nearest (mean) representation in terms of cosine similarity. So training with uncertainties has helped learn better representations, but how?

In the main paper, we analyze our probabilistic loss analytically. We find that the gradient that each image has on the representations is scaled up by its certainty. More certain images receive a higher weight during training than uncertain images. This reduces the impact of samples that are low-quality or potentially mislabeled. We provide more details and comparisons to the deterministic $\mathcal{L}_{\text{NCA++}}$ in the main paper in the appendix.

2.5 Discussion

This work centered around RQ1, finding a method to add uncertainties to representations, namely probabilistic embeddings. Our main finding is that uncertainties are not just an end unto themselves, but help learn better representations. This work also contributed fundamentals that we will see reoccurring in the next chapters, such as the non-isotropic vMF distribution or a corrected approximation to the vMF normalizing constant, excluded here but detailed in the main paper. We have also touched upon RQ2, gaining a first understanding of how uncertainties benefit representation learning.

But the main limitation is that we have evaluated the learned uncertainties only indirectly. They helped learn representations with higher retrieval performance, but we have not evaluated how correct the uncertainties are in and by themselves. In fact, one could argue that they are also trained only indirectly, since they are parametrized by the representation vector norms which other side effects during training could have influenced. We investigate these two open points in the next chapter to verify and expand our understanding of representation uncertainties.

3 Probabilistic Embeddings are Provably Correct

Michael Kirchhof, Enkelejda Kasneci, and Seong Joon Oh. Probabilistic contrastive learning recovers the correct aleatoric uncertainty of ambiguous inputs. *International Conference on Machine Learning (ICML)*, 2023.

3.1 Prologue

"But how do you know that these variances are *correct*?". It was at a poster session at ECCV where I presented the previous paper, and I had just encountered a mind-reader (or my mind was very easy to read). This question turned out to be not just in my mind, but to be haunting the field ever since probabilistic embeddings, or even variational auto-encoders, were invented. In parallel, a new researcher came to Tübingen: Seong Joon Oh. I knew his name. "Aren't you the author of hedged instance embeddings, the first paper on probabilistic embeddings?", I asked. He confirmed. And he also confirmed that he had also been looking for a mathematical answer to the upper question. We compared our notes and so started hours-long discussions of potential proof techniques, thought experiments, and scrutiny of potential loopholes. My coworkers may remember me sitting in the office for days, weeks, and months on end without a laptop, only with countless scribbled papers and a pencil. We succeeded eventually, and the chapter below summarizes our mathematical formalization of the question, as well as its answer.

3.2 Motivation

The previous chapter introduced probabilistic embeddings as a way to represent uncertainties in representation spaces. And they indeed work in the sense that they improve performance. But what is it that their variance parameters capture? Are they indeed the *correct* uncertainties (and if yes, in which sense)? To establish a ground for mathematical arguments, we first need a formal framework. We generalize the non-linear independent component analysis framework of Zimmermann et al. (2021) to formalize data-generating processes that lose information while generating images, where the amount of lost information equals the uncertainty that the probabilistic embeddings have to resemble. The

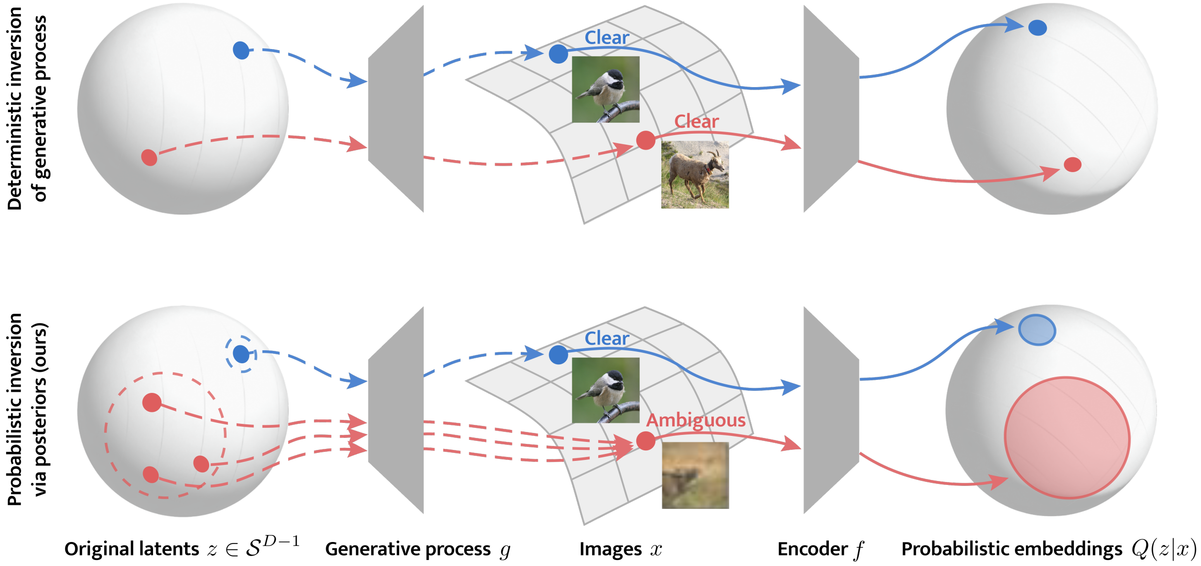

Deterministic inversion of generative process
Probabilistic inversion via posteriors (ours)
Clear
Clear
Clear
Ambiguous
Original latents $z \in \mathcal{S}^{D-1}$
Generative process g
Images x
Encoder f
Probabilistic embeddings $Q(z|x)$

Figure 5: Images are created from unknown latent vectors by a data-generating process. Deterministic image representations intend to rediscover this vector (top). When the data-generating process is probabilistic (bottom) and creates ambiguous images, it loses information about the latent vectors, so that several ones could have created the image. Probabilistic embeddings recover this posterior, which we prove for MCInfoNCE. Figure cited from the original paper (Kirchhof et al., 2023a).

challenge here is that the amount of lost information is also a lost information – we only have access to the final image without any further supervision on how uncertain it is. Strikingly, we find a loss function called MCInfoNCE whose probabilistic embeddings are provably correct: Their variances exactly reflect the amount of lost information, while being learned solely from self-supervision.

3.3 Methods

To be able to discuss any notion of correctness, let us first formalize how images are generated. Following Zimmermann et al. (2021), we assume that some data-generating process turns latent vectors into images, as depicted in Figure 5. In mathematical terms, a function $g : \mathcal{Z} \to \mathcal{X}$ maps the latent z to an image x. To add uncertainties to the process, this mapping is not deterministic. The same latent could be mapped to different images, blurred, cropped, or pixelated in different ways. In statistical terms, g is no more a function with one output per input, but a likelihood $P(X|Z)$. This likelihood is even more complicated than the already complicated generative process g. The trick is that we are not actually interested in the generator $P(X|Z)$ – we are only interested in reconstructing z from x, i.e., in its posterior $P(Z|X)$. This posterior describes which latents z could have generated the image x. If more information about z is lost during the generation of x and x could match several possible z, the posterior becomes wider.

This is what probabilistic embeddings $Q(Z|X)$ are designed to represent. Consequently, we can define correctness for them: Probabilistic embeddings $Q(Z|X)$ (including their uncertainty parameters or variances) are *correct* iff they are equal to the posteriors $P(Z|X)$ of the generative process. What is left to show is that some loss function is minimized by probabilistic embedding estimates only if they are equal to the true posteriors.

To this end, we introduce the MCInfoNCE loss

$$\mathcal{L}_{\text{MCInfoNCE}} = -\log \underset{z \sim Q(z|x)}{\mathbb{E}} \; \underset{z^+ \sim Q(z^+|x^+)}{\mathbb{E}} \; \underset{z_m^- \sim Q(z_m^-|x_m^-)}{\mathbb{E}} \left(\frac{e^{\kappa_{\text{pos}} z^\top z^+}}{\frac{1}{M} e^{\kappa_{\text{pos}} z^\top z^+} + \frac{1}{M} \sum_{m=1}^{M} e^{\kappa_{\text{pos}} z^\top z_m^-}} \right) . \tag{3.1}$$

The innermost part is a self-supervised InfoNCE loss (Oord et al., 2018) that trains the representation z of an image to be closer to a positive partner z^+ (a crop of the same image) than to negatives z_m^- (other images in the batch). This whole inner term is then evaluated not over predicted deterministic representations z but over representations z drawn from the predicted probabilistic embeddings, usually 4 to 16 samples. We implement the probabilistic embeddings by vMF distributions whose variances are learned by an MLP head. This adjustment to turn InfoNCE into the probabilistic MCInfoNCE is enough to guarantee the above identifiability condition, as we show below.

3.4 Core Results

3.4.1 MCInfoNCE Learns the Correct Posterior

The main result of this paper is a proof that the only minimizer of $\mathcal{L}_{\text{MCInfoNCE}}$ are probabilistic embeddings $Q(Z|X)$ that are equal to the true posteriors of the generative process, up to a general rotation of the whole embedding space $\mathcal{Z}$. This has some technical assumptions like that the probabilistic embedding distribution must be the same family as the true posterior, in this case vMF distributions, for otherwise it is impossible to exactly match it. We refer to the appendix for the full statement, conditions, and proof. This proof shows that the variances of probabilistic embeddings are not just training artifacts, but theoretically grounded.

3.4.2 The Correctness is Robust to Violations of Assumptions

We empirically verify this proof in a controlled experiment where a generator network with a randomly initialized posterior produces ambiguous data. We train a probabilistic embedding encoder using MCInfoNCE and find that its probabilistic embeddings are indeed equal to the generator posterior. This is robust to violations of the assumptions, like a different distribution family, a too low or too high dimensional embedding space, or even a generator without uncertainty, in which case MCInfoNCE correctly converges to Dirac probabilistic embeddings, that is, deterministic embeddings. We further test the vMF loss from the previous chapter and find that it also leads to correct posteriors, showing that several probabilistic embedding approaches learn correct uncertainties. However, this is no trivial property, as other losses like hedged instance embeddings (HIB, Oh et al., 2019) do not provide the correct posteriors.

3.4.3 The Learned Uncertainties are Aleatoric Uncertainties

The theoretical formulation of the data-generating process hinted at the idea that these uncertainties are intrinsic to the image and that even the best model, the true posterior, could not reduce them. This is known as aleatoric uncertainty (Hüllermeier and Waegeman, 2021). To investigate this experimentally, we apply MCInfoNCE to CIFAR-10H (Peterson et al., 2019), an image dataset with around 50 annotations per image. The entropy of these annotations serves as a proxy for the irreducible aleatoric uncertainty. We find that our probabilistic embeddings' uncertainties are indeed correlated to those human ones. Similarly, they are correlated to the amount of aleatoric uncertainty we synthetically induce in images by cropping them, thereby losing information. This is first evidence that we can learn the aleatoric uncertainty of images and their representations.

3.5 Discussion

This work focussed on RQ2, understanding what our uncertainties about latents represent. To the best of my knowledge, it is the first paper to find that uncertainties in latent spaces are not just theoretical artifacts of variational training but have a real-world justification and show consistent behaviour. One detail that underlines this is that the uncertainties are trained from a randomly initialized MLP without any prior bias that could explain away its behaviour. This is an issue in the previous chapter (and the literature it followed (Scott et al., 2021; Ko et al., 2021; Ranjan et al., 2017)), where the representation vector norm we used to parametrize the uncertainties is nowadays suspected to be a mere fragment of cross-entropy training (Kang et al., 2023). Beyond RQ2, this chapter also added to RQ1 by giving a new approach to learn uncertainties about latent representations, this time self-supervised, and opened ways for RQ3 by pioneering evaluations that test the uncertainties about unobservable latents against observable ground-truths.

One limitation is that this work is limited to vMF posteriors. An extension to different exponential families as in Zimmermann et al. (2021) or even mixture densities would have been possible because the proof's arguments still hold: 1) MCInfoNCE is a cross-entropy that when optimized equalizes a certain expected positivity score between the generative process (and its posterior embeddings) and the predicted embeddings. This proof does not use the vMF assumption. And 2), this expected positivity can only be equal to the generative process's expected positivity if all embeddings match the true posteriors. The uniqueness argument of 2) would still hold for other posterior families, except that permutations of mixture components and other invariances in distribution parameters could add technical corner-cases. Formalizing these corner cases would have been very time-consuming without adding interesting proof techniques or novel understanding, so we decided to present only the core result on vMFs.

A second limitation is the scale of the experiments. Our first experiment was necessarily a toy experiment, because we needed full control over the data-generating process which is unknown in real-world data. The second experiment on real-world data, however, could have been on a larger scale. MCInfoNCE is scalable because it only adds a lightweight MLP head and 16 Monte-Carlo samples at a late model layer with near diminishing runtime and memory costs. The limiting factor here is that only CIFAR-10H (or datasets of similar scale (Schmarje et al., 2022)) provide sensible ground-truths to compare our uncertainties against. At the time of publication, the literature lacked metrics and benchmarks to evaluate representation uncertainties, which hindered their development and scaling. This changed with the paper we present in the next chapter.

The URL Benchmark for Representation Uncertainties 4

Michael Kirchhof, Bálint Mucsányi, Seong Joon Oh, and Enkelejda Kasneci. URL: A representation learning benchmark for transferable uncertainty estimates. *Neural Information Processing Systems Track on Datasets and Benchmarks (NeurIPS D&B)*, 2023.

4.1 Prologue

"That's neat, but does it scale?", inquired Enkelejda. She did not mean MCInfoNCE, for scaling it to a larger dataset and architecture would still fit on consumer-grade GPUs. Instead, the challenge in scaling the previous results was the evaluation protocol. What metric could we compare the learned uncertainties against if human uncertainty ground-truths are not available? I started forging and comparing several metrics. After weeks of tinkering, there finally was one metric to rule them all, one metric to find the best methods, one metric to scale them and in benchmarks develop them.[1] This metric, the R-AUROC, would allow to benchmark arbitrary representation uncertainty methods on arbitrarily large datasets – if only I could implement them by the rapidly approaching NeurIPS deadline. I turned to Bálint Mucsányi, a student visiting the lab and inter alia an excellent code engineer, and asked "Do you want to write a NeurIPS paper?".

4.2 Motivation

The previous chapters have demonstrated the promises of representation uncertainties. But further progress can only be enabled by a large-scale benchmark. The key to such a benchmark is a metric that quantifies the performance of representation uncertainties and is 1) scalable, 2) easy to implement, but 3) hard enough to be of longer term utility. Human annotations, as in the previous chapter, are not scalable as they have to be recollected for each dataset. Interventional metrics, like checking if uncertainties increase when an image is cropped or deteriorated, can be easily cheated by an overspecialized approach and are already saturated. In this section, we find a metric that fulfills all above criteria, and is even

[1] Inspired by *The Lord of the Rings, The Fellowship of the Ring*, J.R.R. Tolkien, 1954, George Allen and Unwin.

correlated with the human annotation gold-standard. We find it by broadening the view away from specific solutions and towards the problem that representation uncertainties address from a decision theory perspective.

4.3 Methods

We derive our metric, the R-AUROC, by reconsidering the problem from a decision theory perspective. In principle, uncertainties reflect the loss we expect when making a decision. In classification, when we give the decision "dog" with probability 80%, we quantify how high of a 0/1 correctness loss we expect. We can evaluate our uncertainty estimate of 80% by comparing it to the actual 0/1 correctness on test data. In representation learning, our decision is the representation vector and a popular loss is the Recall@1. The Recall@1 measures if, when we embed all test samples, each representation's next neighbour is in the same class. This is also a 0/1 loss. So, when we give an uncertainty estimate about our decision, the representation, we evaluate whether it is predictive of this 0/1 correctness. To quantify this, we use the area under the ROC curve (AUROC) that tells if the uncertainties are predictive of the binary outcome variable. We name this the representation AUROC (R-AUROC). The R-AUROC allows evaluating a broad range of approaches, including ones that give a variance estimate $u(x) \in \mathbb{R}$ instead of a probability $u(x) \in [0, 1]$. It can be evaluated on any classification dataset without new annotations, overcoming the previous hurdle, and can be added to existing representation learning benchmarks in four lines of code, thereby taking the practical hurdle for the field.

The R-AUROC has another advantage inherited from representation learning: We do not need to know the classes at train time. They are added to the Recall@1, and hence the R-AUROC, at test time. This allows testing the representation correctness not just on seen but also on unseen datasets. We leverage this to test the transferability of uncertainty estimates on distribution shifts beyond previous benchmarks on robustness to corruptions (Galil et al., 2023a; Ovadia et al., 2019). We train uncertainty estimators on ImageNet-1k (Deng et al., 2009) and evaluate them on three zero-shot datasets using the R-AUROC. This allow judging which approaches learn a notion of uncertainty that is transferable, paving the way for pretrained uncertainties.

The remaining details of the benchmark protocol are specified in the appendix. The core idea is to train eleven uncertainty estimators from the probabilistic embeddings from Chapters 2 and 3 to ensembles, determine their optimal hyperparameters via Bayesian optimization on a validation set, and test them via the zero-shot R-AUROC. To ensure a fair comparison, we reimplement all approaches as an extension of the `timm` (Wightman, 2019) library. To move towards scalability, another catalyst for future pretrained uncertainties, we use both ResNet 50 (He et al., 2016) and medium-sized Vision Transformers (ViT Medium, Dosovitskiy et al., 2021) as model backbones. Together, this comprises the uncertainty-aware representation learning (URL) benchmark.

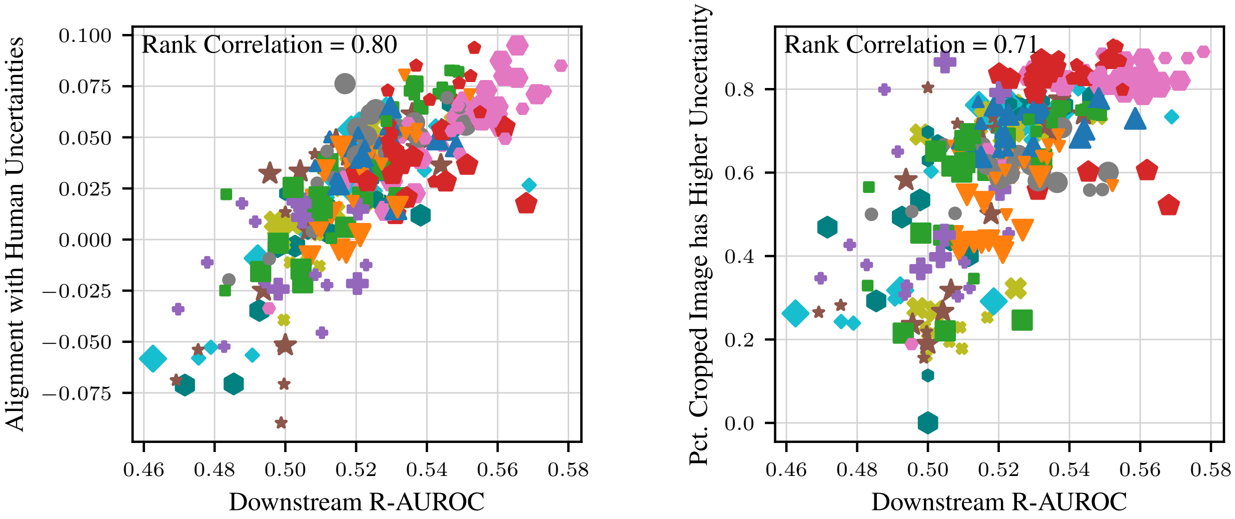


Figure 6: Dots represent all models we train with all approaches, hyperparameters, and backbones. Models with a higher R-AUROC reflect human uncertainties better (left) and behave better under uncertainty inducing transforms like cropping (right). This supports the R-AUROC empirically. Figure cited from the original paper (Kirchhof et al., 2023b).

4.4 Core Results

4.4.1 The R-AUROC Metric Correlates with Gold Standard Metrics

Before we start, we verify the integrity of our novel R-AUROC metric empirically by comparing it to existing uncertainty evaluation metrics. First, we use the gold standard from the previous section. That is, besides the R-AUROC, we track how well correlated the uncertainty estimates are with human annotator disagreements on five datasets (Schmarje et al., 2022), including the previous CIFAR-10H. Figure 6 shows that these two metrics are highly correlated (rank correlation = 0.80) across all approaches, backbones, hyperparameters, and seeds we used in the benchmark. This means that whenever a model has a high R-AUROC, it also tends to score high on the gold standard human annotator metric (which is unavailable in most datasets). In the plot, even their random performance levels, 0.5 for the R-AUROC and 0 for rank correlation with human uncertainties, coincide. The same holds for an interventional metric that checks how often a smaller cropped version of an image receives a higher uncertainty estimate than its original (Figure 6, right). Last, the R-AUROC is also highly correlated with the widely used classification AUROC, when the latter is available on the seen classes of the ImageNet-1k validation set (see appendix). These experiments demonstrate that the R-AUROC judges uncertainty estimates consistent with previous gold standards, while being simpler to compute and available on arbitrary, even unseen, classification datasets.

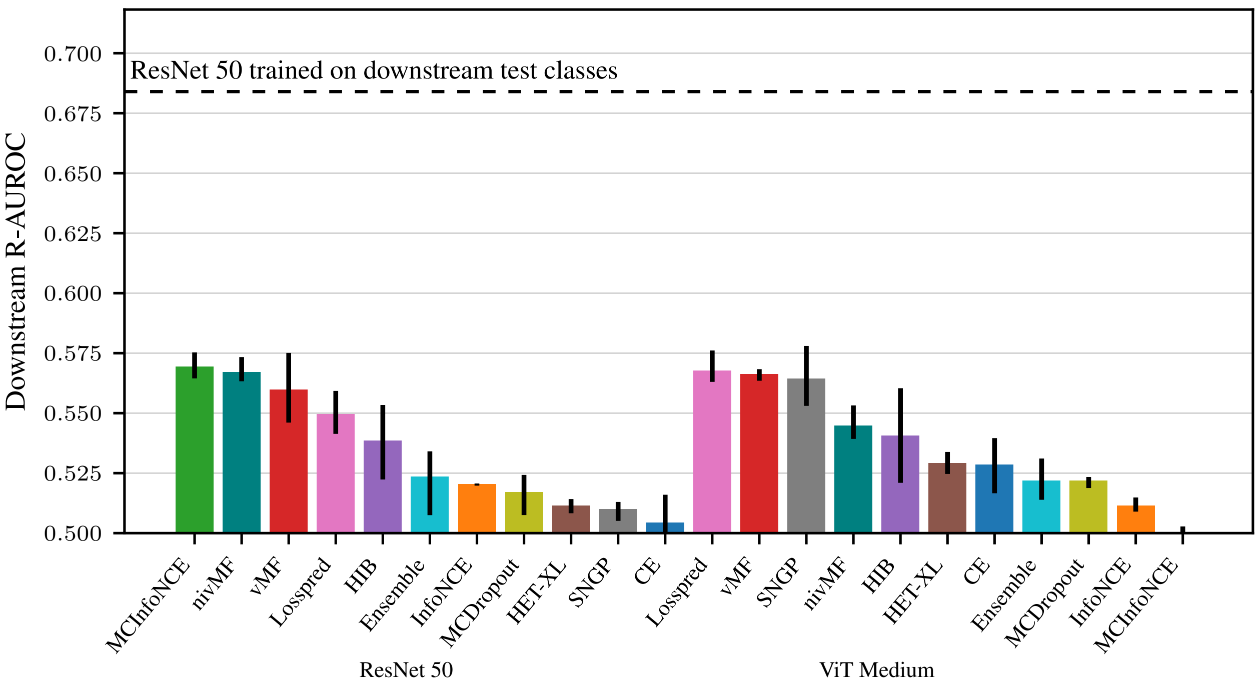


Figure 7: Methods from Chapters 2 and 3, MCInfoNCE, nivMF, and vMF, give among the best transferable uncertainties. Loss prediction also stands out, which we further investigate in Chapter 5. Bars indicate minimum and maximum performance across three seeds. Figure adapted from the original paper (Kirchhof et al., 2023b).

4.4.2 Approaches from Previous Chapters are Among the Best

We now use the R-AUROC to evaluate the methods from the previous chapters, MCInfoNCE, nivMF, and vMF, on a large scale and compare them to contemporary methods. Figure 7 shows that they are the best three approaches on ResNets and among the best on ViTs. A second approach, loss prediction, which we imported in this paper from regression literature (Upadhyay et al., 2023b; Levi et al., 2022; Laves et al., 2020; Yoo and Kweon, 2019), also shows stable performance across both models. The plot additionally highlights that the R-AUROC is far from being saturated: As a reference, we trained a ResNet 50 with cross-entropy loss on the downstream datasets to obtain an upper bound on the performance. This is a loose bound since the zero-shot approaches do not know the precise downstream classes and can not use different uncertainty estimators on each downstream dataset, but it shows that there is room for improvements. In Chapter 5, we reduce this gap by using URL to develop a new state-of-the-art.

4.4.3 Uncertainty Quantification Sometimes Degrades the Main Task

Figure 7 reveals another detail: MCInfoNCE is performant on ResNet 50, but not on ViT Medium. This is not a bug in implementation, but the result of a conflict of the intertwined representation learning and uncertainty estimation tasks. Although MCInfoNCE has one joint optimum, in practice the predicted (mean) representations and the predicted uncertainties drive the backbone into different gradient directions, and here the representatons' gradients were orders of magnitude stronger. This conflict is not exclusive to MCInfoNCE. In the main paper, we find that 15 of the 22 approaches have a trade-off when optimizing for Recall@1 versus for R-AUROC. We solve this in Chapter 5.

4.5 Discussion

This paper answers RQ3 by providing a both theoretically funded and empirically well behaving metric to evaluate representation uncertainties. We developed it with RQ4 in mind, ensuring that it can be scaled to ImageNet and beyond. We also found that an approach originating from regression, loss prediction, achieves strong performance, adding to the methods sought after in RQ1. It takes a less variational, more direct approach at learning our desired representation uncertainties. This is why we further investigate unlocking its full potential in the next chapter, with a special emphasis on the gradient conflict we uncovered in this benchmark.

Pretrained Representation Uncertainties 5

Michael Kirchhof, Mark Collier, Seong Joon Oh, and Enkelejda Kasneci. Pretrained visual uncertainties. arXiv preprint arXiv:2402.16569, 2024. Under submission.

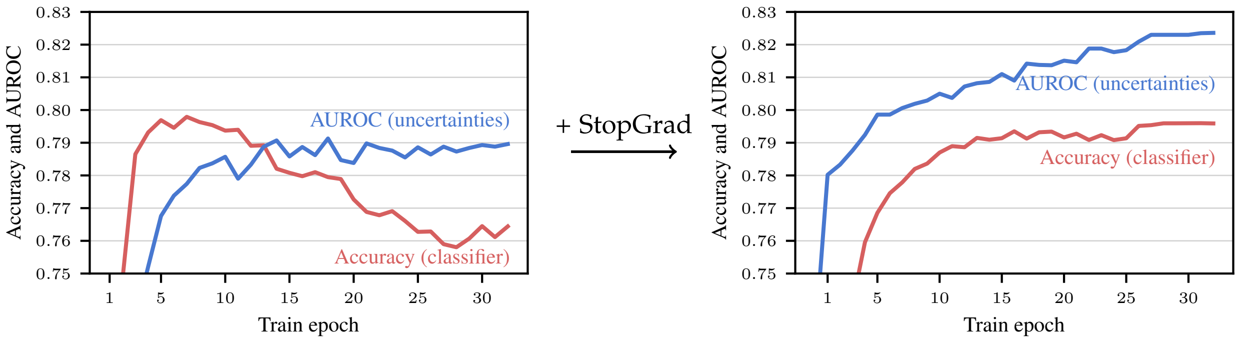
5.1 Prologue

"We need to talk!", I said, excitedly. I had just met Mark Collier at ICML. He had been working on the same problem of scalable uncertainties, had come to the same solution, probabilistic embeddings (Collier et al., 2023), and, I figured, now faced the same hurdle. He had a look at my preliminary analysis of the gradient conflicts. A 15 minutes coffee break became a 1 hour lunch break became regular meetings with Enkelejda and Joon. We were determined to scale our previous efforts up, while cutting away any complexity that did not lead to measurable improvements on the URL benchmark. Ultimately, this chapter compiles the findings on all above research questions into one downloadable plug-and-play model.

5.2 Motivation

The previous chapters have shown that our representation uncertainties are scalable and learn transferable notions of uncertainty. The last challenge is to scale them to large pretraining datasets, so that they can be deployed in a zero-shot plug-and-play manner by downstream practitioners. In particular, the pretrained model's representation uncertainties should (i) not interfere with the main pretraining or downstream task, (ii) generalize to zero-shot downstream datasets, (iii) flexibly adjust to any (downstream) task, (iv) have minimal compute overhead, and (v) converge stably to ensure scalability.

The main remaining hurdles are desiderata (i) and (v) because of the gradient conflict we discovered in Chapter 4. As portrayed in Figure 8a for loss prediction, where uncertainty estimation and representation learning are two distinct losses, the uncertainty objective hurts the performance of the pretrained model's main objective, transferable representations, and vice versa. This is because the gradients flowing back from both task heads attempt to change the backbone in interfering directions. This was so far avoided by



(a) Vanilla Loss Prediction

(b) Ours: Loss Prediction + StopGrad

Figure 8: There is a conflict between the uncertainty and the classification objective when pretraining on ImageNet-1k, deteriorating both performances. A StopGrad resolves this conflict, enabling stable and scalable training. Figure cited from the original paper (Kirchhof et al., 2024).

stopping early, roughly around epoch 12 in the figure. However, early stopping prohibits training on large pretraining corpora. This chapter presents our solution to solve the conflict, scale up the training, and, finally, provide pretrained representation uncertainties for computer vision, as we set out to find in this thesis.

5.3 Methods

We decide to base our model on the loss prediction method from Chapter 4 over the equally well performing probabilistic embeddings, which we discuss further in the discussion below. Loss prediction stems from a decision theory perspective. The (in-)correctness of any task is defined by its loss function. Hence, to provide uncertainty estimates $u(x) \in \mathbb{R}$ that predict incorrectness, we predict the (gradient-detached) loss at each input (Yoo and Kweon, 2019). As in the previous two chapters, this uncertainty estimation is realized by a lightweight MLP head added to a pretrained model. To a practitioner, this results in a simple dual-output API.

```
embedding, uncertainty = pretrained_model(input)
```

A big hurdle is the gradient conflict. Although we have experimented with techniques like PCGrad (Yu et al., 2020) to resolve it, the best performing and simplest solution is to place a `StopGrad` between the uncertainty MLP head and the model backbone. This strictly ensures the non-interference principle (i) and improves not only the main objective, but also the uncertainty performance, as shown in Figure 8b.

The last challenge was to train on large pretraining corpora, here ImageNet-21k (Deng et al., 2009), with large Vision Transformer backbones under limited compute. The solution is also enabled by StopGrad: Since StopGrad ensures that the backbone and classifier head are completely independent from influences of the uncertainty module, their training is orthogonal. We first load a pretrained checkpoint for the backbone (and classifier), and then cache the representations $e(x)$ throughout the whole training process once (all epochs including their random augmentations). Then we train the uncertainty head, which only needs to load the representations as inputs and the class labels as targets from disk. This increases the training throughput by a factor of 180x, enabling to train the uncertainty head of a ViT-Large for seven ImageNet-21k epochs (92 million samples) in 2:26 hours on a single V100 GPU, as opposed to 18 days when loading images x from disk.

We report more possible methodological enhancements in the main paper, but as negative results. None of them substantially increases the performance on the URL benchmark. We remove them to maintain the simplest possible approach.

5.4 Core Results

5.4.1 Pretrained Visual Uncertainties Transfer Across Datasets

The performance of our enhanced pretrained uncertainties exceeds that of the previous approaches on the URL benchmark, even when we use the smaller ImageNet-1k dataset for pretraining. In fact, the datasets in the URL benchmark (CUB 200 Wah et al. (2011), SOP (Song et al., 2016), and CARS 196 (Krause et al., 2013)) are among the hardest due to their fine-grained and thus highly specialized classification task. Figure 9 shows that our pretrained uncertainties generalize to other natural image datasets, including those from the visual task adaptation benchmark (Zhai et al., 2020). This shows that our pretrained uncertainties behave as expected from a pretrained model, spanning the domain of the natural images pretraining dataset.

5.4.2 Pretrained Uncertainties Represent Aleatoric Uncertainties

When providing uncertainties, it is inevitable to specify which kind of uncertainties these are, commonly epistemic or aleatoric (Hüllermeier and Waegeman, 2021). Epistemic denotes uncertainties about the correct choice of model parameters on unseen inputs, which can be reduced by collecting more similar inputs. Aleatoric are uncertainties in the data itself, e.g. a blurred or pixelated image, which are irreducible even with an expert or a Bayes-optimal model. We hypothesize that our pretrained visual uncertainties capture aleatoric uncertainty, without epistemic uncertainties.

We find three pieces of evidence for this in the paper. First, ImageNet images where humans report ambiguity (Beyer et al., 2020) receive a higher pretrained uncertainty estimate than images where they agree on a Dirac label, similar to Chapters 3 and 4.

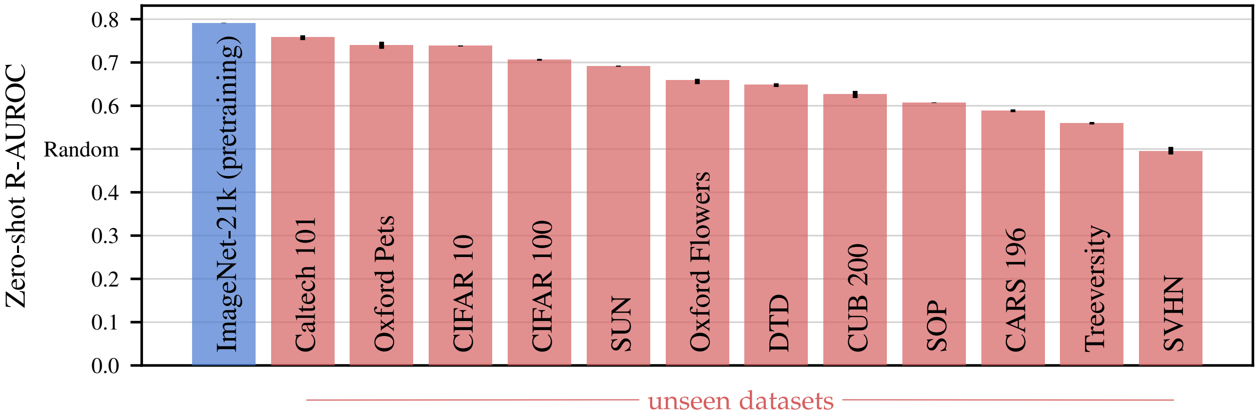


Figure 9: Pretrained uncertainties transfer to various downstream datasets, as measured by the R-AUROC. Bars indicate minimum and maximum performance across three seeds. Figure cited from the original paper (Kirchhof et al., 2024).

Second, if we intervene on the images by cropping, but also by noising, blurring, or overlaying with grey boxes, uncertainties increase with the strength of intervention. Third, we find that uncertainty estimates on unseen datasets follow the same distribution as on the seen pretraining dataset, indicating the absense of epistemic influences.

These findings support our hypothesis that pretrained uncertainties model aleatoric uncertainty exclusively. This is a positive trait for a pretrained model, since it is intended to be deployed on unseen data where generally high epistemic uncertainty could drown out any aleatoric signal. It also provides one of the first methods that can disentangle the two uncertainties, which has been a recent effort in the field because it enables novel applications (Wimmer et al., 2023; Mucsányi et al., 2024).

5.5 Discussion

This work focussed on RQ4, overcoming remaining challenges to enable scaling, benefiting from the approaches and benchmarks we've built through the previous chapters and research questions. But it also adds new understanding about which uncertainties our representation uncertainties resemble, thereby contributing to RQ2.

We made two major design choices in developing our pretrained visual uncertainties: Using loss prediction as a starting point, and applying `StopGrad`. Both have viable alternatives, which we discuss in the following two paragraphs.

We made our decision about the approach to base pretrained uncertainties from a problem-oriented perspective by introducing five desiderata meaningful to future practitioners. Both loss prediction and probabilistic embeddings have shown strong empirical performance in Chapter 4, have theoretical foundations, and, with `StopGrad`, ensure non-interference, fulfilling desiderata (i), (ii), and (v). The biggest differences lie in the effort downstream users would have to make to adjust the pretrained model to their task of choice. In Chapter 3, we have seen that a simple blueprint can turn deterministic losses like InfoNCE into probabilistic ones like MCInfoNCE, maintaining their original properties and adding guarantees about the uncertainties. We are confident that this holds for further losses, but in comparison loss prediction is guaranteed by construction to adapt to any loss a downstream user may insert. As for compute, both methods output a scalar uncertainty $u(x) \in \mathbb{R}$ via an MLP head at inference time. However, at train time, MCInfoNCE requires Monte-Carlo samples whereas loss prediction uses the already computed loss value. This difference is small as sampling only happens in a late layer, but practitioners may be discouraged when remembering the large computational hurdle of sampling-based approaches like deep ensembles or MCDropout (Mucsányi et al., 2024). Thus, we anticipate that pretrained uncertainties based on loss prediction will be more readily accepted outside the uncertainty quantification field.

The decision for `StopGrad` was based on Occam's razor. Instead of using `StopGrad` to ensure that the uncertainty training does not interfere with the backbone, we could have first trained the backbone and classifier head and then frozen it before training the uncertainty head. This stagewise training would have been functionally the same, but `StopGrad` unloads this implementation hurdle from practitioners. Further, it enables training both the main task head and the uncertainty head at the same time (as in Figure 8), providing a fail-safe mechanism for future users. Second, we could have implemented a gradient disentanglement approach like PCGrad to overcome the interferences. However, besides not working empirically in preliminary experiments, this would complicate training and add a dependency to be tuned. We encourage future researchers to reassess this point when the multi-task learning community finds new, robust algorithms.

Discussion 6

Uncertainties are often thought of as probabilities over output classes or intervals of the target variable in regression. These uncertainties are specific to each individual task. To provide uncertatinties that are more independent of the task, we attached uncertainties to representations. Chapter 2 demonstrated how to achieve this by simple adjustments to representation learning. Chapter 3 showed that these uncertainties about latent representations indeed have a provable notion of correctness. Chapter 4 provided a benchmark to quantify how practically correct different methods for representation uncertainties are. Chapter 5 brought these findings to a large scale and developed a pretrained model whose representation uncertainties transfer across datasets.

Besides this transferability, representation uncertainties also enable novel applications, which we outline below. Further, we comment on how our pretrained uncertainties are a starting point for specialized uncertainties, in which we see fruitful ends in future of uncertainty quantification research.

6.1 Applications

Representation uncertainties add a new dimension to representations that opens up novel applications. Having provided a downloadable model for representation uncertainties in Chapter 5, we expect future research to explore the multitude of applications that representation uncertainties enable. We outline some applications below, some of which we already investigated in the main papers in the appendix, whereas others are given as inspiration for future researchers.

We start with a traditional application of uncertainties, selective prediction (El-Yaniv et al., 2010; Tran et al., 2022; Galil et al., 2023b). In selective prediction, the estimated uncertainty of each input is compared to a threshold and if the uncertainty is too high, we refuse to predict on this input. Figure 10 shows that when increasing this threshold and rejecting more of the samples that the model considers uncertain, the accuracy on the remaining samples in fact increases. While machine learning may not be able to handle all inputs, this allows giving automated decisions at least for certain ones at a high accuracy. Remaining samples can, e.g., be asked to be re-taken or handed over to humans for inspection, as proposed by Tran et al. (2022).

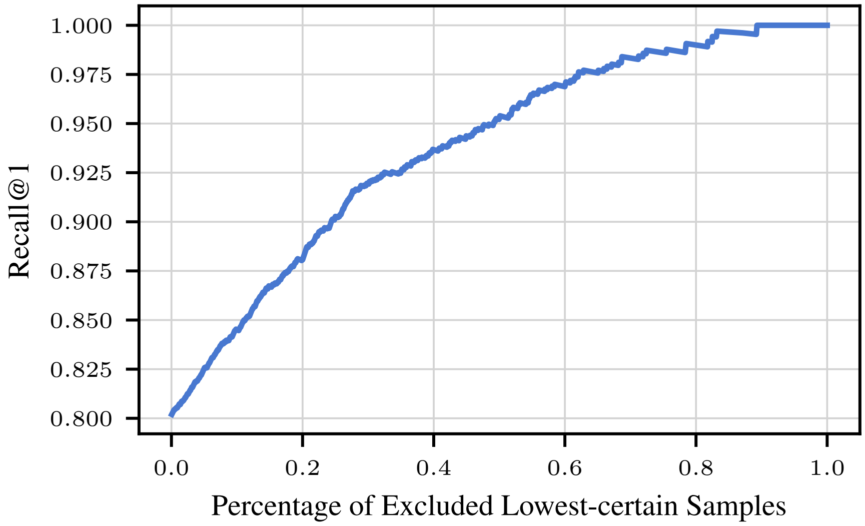


Figure 10: When the model rejects inputs where it predicts a high uncertainty, it can achieve a high accuracy on the remainder of the data. This abstained prediction enables deploying models in situations with a high desired accuracy. Performance of MCInfoNCE trained on CIFAR-10. Figure cited from the original paper (Kirchhof et al., 2023a).

The paradigm of selective prediction can also be applied to retrieval. In retrieval, the user inputs an image (or in multimodal settings, a text (Chun et al., 2021; Upadhyay et al., 2023a)) and we output matching images from our database by comparing the representations of inputs and the database. Uncertainties can be added in two ways: We can reject inputs that are uncertain, and we can remove images with high representation uncertainties from our database to prevent matching them to any input. In the main paper of Chapter 5, we show that these enhancements decrease the retrieval error, both on databases the model was trained on (10% reduction each) and on databases where it gives zero-shot uncertainties (14% and 17% each). This marks low-hanging performance improvements and paves the way for safer retrieval.

If one is hesitant to reject user queries, one can also react more softly to ambiguous retrieval inputs. Figure 11 shows how we utilize the representation uncertainty, here probabilistic embeddings over the latent space from Chapter 3, to flexibly adjust how many possible matches we return to the user. If an input is clear, like the car in the top example, we return only a single sample, also a car. If the input is ambiguous, like in the lower examples, we return multiple matches, spanning images from several possible classes. This simple way to visualize the uncertainty is concurrently explored by Upadhyay et al. (2023a) and reminds of the current advances in conformal prediction (Angelopoulos and Bates, 2022) where one outputs the set of all possible class labels to cover the true class with high probability. This similarity is no coincidence: Conformal prediction builds and calibrates these sets on the basis of score functions that indicate the

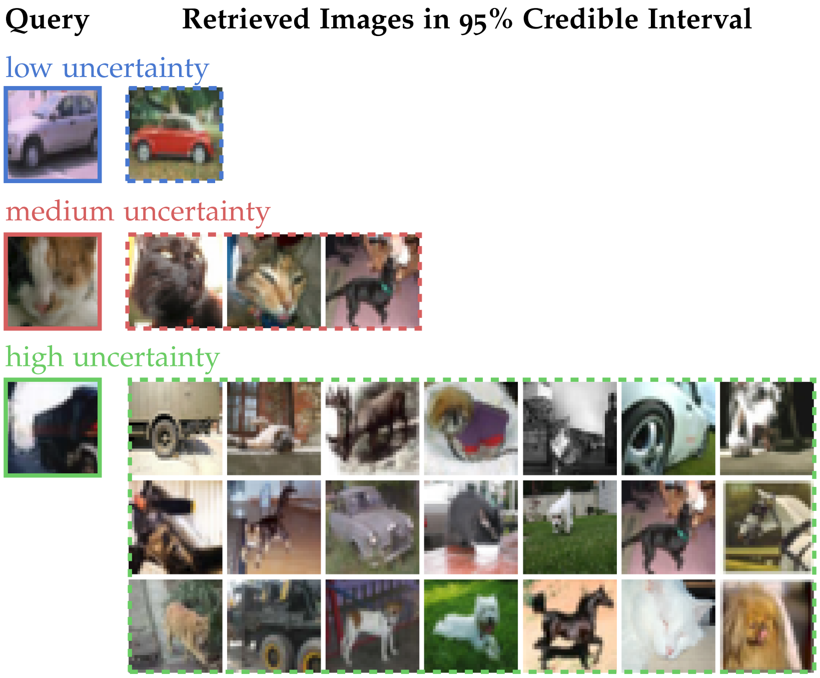


Figure 11: When a user inputs an image whose representation is uncertain, we retrieve multiple images that may match the input. The size of the output set depends on the ambiguity of the input. Here, it is the 95% highest density region of the input's probabilistic embedding, learned by MCInfoNCE on CIFAR-10. Figure adapted from the original paper (Kirchhof et al., 2023a).

uncertainty of every possible event. Our (pretrained) representation uncertainties are such score functions, enabling future advancements in zero-shot conformal prediction.

Another area that can benefit from our representation uncertainties is active learning. The most recent approaches (Mindermann et al., 2022; Lahlou et al., 2023) seek samples that are not learned yet but of high quality. In other words, samples that have a high epistemic but low aleatoric uncertainty. To this end, they require estimators for aleatoric uncertainty that are not influenced by epistemic uncertainty. As we have seen in Chapter 5, our pretrained representation uncertainties are among the first approaches to fulfill these criteria, simplifying active learning endeavours.

A similar strain of literature is dataset curation and handling noisy training signals (Ortiz-Jimenez et al., 2023; Marion et al., 2023; Sachdeva et al., 2024; Evans et al., 2024). This challenge gained new interest with the current paradigm of using web-crawled, uncurated data to train large models (Schuhmann et al., 2021; Tran et al., 2022). Recent approaches find that removing low-quality data improves performance. Pretrained representation uncertainties capture precisely this, inputs with a generally low quality, and can be computed on the spot even for new data, enabling future use as dataset curators.

Last, representation uncertainties can be used in any approach that uses representations to visualize datasets, such as clustering (van der Maaten and Hinton, 2008; McInnes et al., 2018). We have already seen in Figure 2 how uncertainties enhance these plots to communicate uncertainties to practitioners, allowing to understand and debug datasets for more trustworthy machine learning.

6.2 Specialized Uncertainties

Throughout the thesis, the reader may have noted an increasing abstraction of our uncertainties. Whereas previous approaches commonly define uncertainties as, e.g., classification probabilities, Chapters 2 and 3 first abstract uncertainties away from the classification task and towards uncertainties about representations in general, independent of the specific task. Chapter 4 generalizes this further by going from probabilistic embeddings towards any uncertainty estimator that provides uncertainties about representations. Last, we settle on a loss-based interpretation of uncertainties. If uncertainties aim to estimate how wrong we are then the loss quantifies what wrongness precisely means in the task the practitioner is handling. This is the paradigm that our pretrained uncertainties use in Chapter 5, with the intention that the pretrained uncertainties will adjust to the practitioner's loss once they are finetuned on downstream data.

This shows our main vision: Specializing uncertainties to individual tasks (Franchi et al., 2022; Mucsányi et al., 2024). This is a pragmatic generalization of the recent efforts in uncertainty disentanglement. Here, the field is currently moving from one-fits-all predictive uncertainty values (Gal and Ghahramani, 2016; Lakshminarayanan et al., 2017) to disentangled aleatoric and epistemic uncertainties (Hüllermeier and Waegeman, 2021; Valdenegro-Toro and Mori, 2022; Wimmer et al., 2023; Mucsányi et al., 2024). One unsolved issue in this framework is that epistemic uncertainty remains only vaguely defined (Der Kiureghian and Ditlevsen, 2009; Jürgens et al., 2024). We expect that a loss-based view will move uncertainty estimation forward by making it more explicit and more specialized to the tasks practitioners intend to solve with it.

As examples, aleatoric uncertainty in classification becomes the remaining cross-entropy loss of the trained classifier. Epistemic uncertainty for outlier detection becomes a 0/1 loss of a binary OOD classification task. Density estimation is predicting a log likelihood loss. If there are further tasks a practitioner wants to use uncertainties for, they do not have to be fitted into the epistemic-aleatoric dichotomy, but can be defined as a precise task to be optimized by the uncertainty module. This makes uncertainty estimation more pragmatic and more explicitly optimizable since loss prediction is, in essence, just another regression task. We anticipate that this specialization enabled by abstraction will both simplify and unify future works on uncertainty estimation.

6.3 Conclusion

This thesis brought uncertainties in computer vision to the layer of representations. This has the advantage that they can be pretrained on a large scale and then transferred to new datasets and tasks. Besides these practical advances, we also explored the theoretical foundation of uncertainties about latents and how to benchmark them. We compiled all these theoretical and practical findings into one downloadable model in order to facilitate uncertainty quantification for researchers inside and, importantly, outside the field. This demonstrates our vision for the future of uncertainty quantification: We encourage researchers from inside the field to shape their sophisticated methods and findings into pragmatic answers to the pragmatic questions practitioners outside the field face. We expect that this will enable a widespread application of uncertainties, making trustworthy machine learning the norm.

List of Abbreviations

API	Application Programming Interface
AUROC	Area Under the Receiver Operating Characteristic, measure
Caltech 101	Caltech 101 dataset (Fei-Fei et al., 2004)
CARS	Stanford Cars 196 dataset (Krause et al., 2013)
CE	Class Entropy, uncertainty estimator
CIFAR-10	Canadian Institute For Advanced Research dataset (Krizhevsky, 2009)
CIFAR-100	Canadian Institute For Advanced Research dataset (Krizhevsky, 2009)
CUB	Caltech-UCSD Birds-200-2011 dataset (Wah et al., 2011)
DTD	Decribable Textures dataset (Cimpoi et al., 2014)
ECCV	European Conference on Computer Vision
ELK	Expected Likelihood Kernel (Jebara and Kondor, 2003)
GPU	Graphics Processing Unit
HET-XL	Large Heteroscedastic Classifier (Collier et al., 2023)
HIB	Hedged Instance Embeddings (Oh et al., 2019)
ICML	International Conference on Machine Learning
InfoNCE	Info Noise Contrastive Estimation loss (Oord et al., 2018)
Losspred	Loss prediction (Kirchhof et al., 2023b)
MCDropout	Monte-Carlo Dropout (Gal and Ghahramani, 2016)
MCInfoNCE	Monte-Carlo InfoNCE (Kirchhof et al., 2023a)
MLP	Multi-layer Perceptron
NeurIPS	Neural Information Processing Systems conference
nivMF	non-isotropic von Mises-Fisher distribution (Kirchhof et al., 2022)
Oxford Flowers	102 Category Flower dataset (Nilsback and Zisserman, 2008)
Oxford Pets	Oxford-IIIT Pet dataset (Parkhi et al., 2012)
PCGrad	Projecting Conflicting Gradients (Yu et al., 2020)
ProxyNCA	Proxy Noise Contrastive Estimation (Movshovitz-Attias et al., 2017)
R-AUROC	Representation AUROC, measure (Kirchhof et al., 2023b)

ResNet	Residual Neural Network (He et al., 2016)
RQ	Research Question
SNGP	Spectral-normalized Neural Gaussian Process (Liu et al., 2020)
SOP	Stanford Online Products dataset (Song et al., 2016)
StopGrad	Gradient Stopping module
SUN	Scene Recognition Benchmark Database (Xiao et al., 2010)
SVHN	Street View House Numbers dataset (Netzer et al., 2011)
Treeversity	Treeversity dataset, single label (Schmarje et al., 2022)
URL	Uncertainty-aware Representation Learning benchmark (Kirchhof et al., 2023b)
ViT	Vision Transformer (Dosovitskiy et al., 2021)
vMF	von Mises Fisher distribution (Fisher, 1953)
VTAB	Visual Task Adaptation Benchmark, dataset collection (Zhai et al., 2020)

List of Figures

1 Images can be inherently ambiguous, making it necessary to quantify their uncertainty. Both images are from the ImageNet-1k benchmark dataset (Deng et al., 2009). 14

2 Each point is the representation of an Oxford Pets image (Parkhi et al., 2012). The size of each dot visualizes the uncertainty $u(x)$ of the representation, calculated by our approach in Chapter 5. This makes it easier to detect images that are naturally ambiguous (large and transparent). Figure cited from the original paper (Kirchhof et al., 2024). 15

3 Densities of a vMF and a non-isotropic vMF distributions on a three-dimensional unit-sphere. Purple is a low and yellow a high density. Figure adapted from the original paper (Kirchhof et al., 2022). 22

4 Probabilistic embeddings (green) lead to better retrieval performance than deterministic ones (blue). Bars show the standard deviation across five seeds. Figure adapted from the original paper (Kirchhof et al., 2022). 23

5 Images are created from unknown latent vectors by a data-generating process. Deterministic image representations intend to rediscover this vector (top). When the data-generating process is probabilistic (bottom) and creates ambiguous images, it loses information about the latent vectors, so that several ones could have created the image. Probabilistic embeddings recover this posterior, which we prove for MCInfoNCE. Figure cited from the original paper (Kirchhof et al., 2023a). 26

6 Dots represent all models we train with all approaches, hyperparameters, and backbones. Models with a higher R-AUROC reflect human uncertainties better (left) and behave better under uncertainty inducing transforms like cropping (right). This supports the R-AUROC empirically. Figure cited from the original paper (Kirchhof et al., 2023b). 33

7 Methods from Chapters 2 and 3, MCInfoNCE, nivMF, and vMF, give among the best transferable uncertainties. Loss prediction also stands out, which we further investigate in Chapter 5. Bars indicate minimum and maximum performance across three seeds. Figure adapted from the original paper (Kirchhof et al., 2023b). 34

8 There is a conflict between the uncertainty and the classification objective when pretraining on ImageNet-1k, deteriorating both performances. A StopGrad resolves this conflict, enabling stable and scalable training. Figure cited from the original paper (Kirchhof et al., 2024). 38

9 Pretrained uncertainties transfer to various downstream datasets, as measured by the R-AUROC. Bars indicate minimum and maximum performance across three seeds. Figure cited from the original paper (Kirchhof et al., 2024). 40
10 When the model rejects inputs where it predicts a high uncertainty, it can achieve a high accuracy on the remainder of the data. This abstained prediction enables deploying models in situations with a high desired accuracy. Performance of MCInfoNCE trained on CIFAR-10. Figure cited from the original paper (Kirchhof et al., 2023a). 44
11 When a user inputs an image whose representation is uncertain, we retrieve multiple images that may match the input. The size of the output set depends on the ambiguity of the input. Here, it is the 95% highest density region of the input's probabilistic embedding, learned by MCInfoNCE on CIFAR-10. Figure adapted from the original paper (Kirchhof et al., 2023a). . 45

Bibliography

Anastasios N. Angelopoulos and Stephen Bates. A gentle introduction to conformal prediction and distribution-free uncertainty quantification. *arXiv preprint arXiv:2107.07511*, 2022. Cited on page 44.

Yoshua Bengio, Aaron Courville, and Pascal Vincent. Representation learning: A review and new perspectives. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35 (8):1798–1828, 2013. Cited on page 15.

Viktor Bengs, Eyke Hüllermeier, and Willem Waegeman. On second-order scoring rules for epistemic uncertainty quantification. In *International Conference on Machine Learning (ICML)*, 2023. Cited on page 17.

José M Bernardo and Adrian FM Smith. *Bayesian theory*, volume 405. John Wiley & Sons, 2009. Cited on page 16.

Lucas Beyer, Olivier J Hénaff, Alexander Kolesnikov, Xiaohua Zhai, and Aäron van den Oord. Are we done with ImageNet? *arXiv preprint arXiv:2006.07159*, 2020. Cited on pages 13 and 39.

Biagio Brattoli, Joseph Tighe, Fedor Zhdanov, Pietro Perona, and Krzysztof Chalupka. Rethinking zero-shot video classification: End-to-end training for realistic applications. In *Computer Vision and Pattern Recognition (CVPR)*, 2020. Cited on page 21.

Giuseppina Carannante, Dimah Dera, Nidhal C Bouaynaya, Ghulam Rasool, and Hassan M Fathallah-Shaykh. Trustworthy medical segmentation with uncertainty estimation. *arXiv preprint arXiv:2111.05978*, 2021. Cited on page 16.

Raghavendra Chalapathy and Sanjay Chawla. Deep learning for anomaly detection: A survey. *arXiv preprint arXiv:1901.03407*, 2019. Cited on page 16.

Ning-San Chang and King Sun Fu. A relational database system for images. *Technical Report TR-EE 79-28*, 1979. Cited on page 15.

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *Proceedings of the 37th International Conference on Machine Learning (ICML)*, 2020. Cited on page 16.

Xinlei Chen and Kaiming He. Exploring simple siamese representation learning. In *Computer Vision and Pattern Recognition (CVPR)*, 2021. Cited on page 16.

Sumit Chopra, Raia Hadsell, and Yann LeCun. Learning a similarity metric discriminatively, with application to face verification. In *Computer Vision and Pattern Recognition (CVPR)*, 2005. Cited on page 16.

Sanghyuk Chun, Seong Joon Oh, Rafael Sampaio De Rezende, Yannis Kalantidis, and Diane Larlus. Probabilistic embeddings for cross-modal retrieval. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021. Cited on page 44.

M. Cimpoi, S. Maji, I. Kokkinos, S. Mohamed, and A. Vedaldi. Describing textures in the wild. In *Computer Vision and Pattern Recognition (CVPR)*, 2014. Cited on page 49.

Mark Collier, Rodolphe Jenatton, Basil Mustafa, Neil Houlsby, Jesse Berent, and Effrosyni Kokiopoulou. Massively scaling heteroscedastic classifiers. *arXiv preprint arXiv:2301.12860*, 2023. Cited on pages 17, 37, and 49.

Peng Cui, Dan Zhang, Zhijie Deng, Yinpeng Dong, and Jun Zhu. Learning sample difficulty from pre-trained models for reliable prediction. In *Neural Information Processing Systems (NeurIPS)*, 2023. Cited on page 17.

Erik Daxberger, Agustinus Kristiadi, Alexander Immer, Runa Eschenhagen, Matthias Bauer, and Philipp Hennig. Laplace redux-effortless bayesian deep learning. *Neural Information Processing Systems (NeurIPS)*, 2021. Cited on page 16.

Mostafa Dehghani, Josip Djolonga, Basil Mustafa, Piotr Padlewski, Jonathan Heek, Justin Gilmer, Andreas Peter Steiner, Mathilde Caron, Robert Geirhos, Ibrahim Alabdulmohsin, et al. Scaling vision transformers to 22 billion parameters. In *International Conference on Machine Learning (ICML)*, 2023. Cited on page 17.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. ImageNet: A large-scale hierarchical image database. In *Computer Vision and Pattern Recognition (CVPR)*, 2009. Cited on pages 13, 14, 16, 32, 38, and 51.

Zhijie Deng, Feng Zhou, and Jun Zhu. Accelerated linearized laplace approximation for bayesian deep learning. *Neural Information Processing Systems (NeurIPS)*, 2022. Cited on page 16.

Stefan Depeweg, Jose-Miguel Hernandez-Lobato, Finale Doshi-Velez, and Steffen Udluft. Decomposition of uncertainty in bayesian deep learning for efficient and risk-sensitive learning. In *International Conference on Machine Learning (ICML)*, 2018. Cited on page 17.

Armen Der Kiureghian and Ove Ditlevsen. Aleatory or epistemic? does it matter? *Structural safety*, 31(2):105–112, 2009. Cited on page 46.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations (ICLR)*, 2021. Cited on pages 32 and 50.

Matthijs Douze, Alexandr Guzhva, Chengqi Deng, Jeff Johnson, Gergely Szilvasy, Pierre-Emmanuel Mazaré, Maria Lomeli, Lucas Hosseini, and Hervé Jégou. The Faiss library. *arXiv preprint arXiv:2401.08281*, 2024. Cited on pages 15 and 21.

Ran El-Yaniv et al. On the foundations of noise-free selective classification. *Journal of Machine Learning Research*, 11(5), 2010. Cited on page 43.

Talfan Evans, Shreya Pathak, Hamza Merzic, Jonathan Schwarz, Ryutaro Tanno, and Olivier J Henaff. Bad students make great teachers: Active learning accelerates large-scale visual understanding. *arXiv preprint arXiv:2312.05328*, 2024. Cited on page 45.

Li Fei-Fei, Rob Fergus, and Pietro Perona. Learning generative visual models from few training examples: An incremental Bayesian approach tested on 101 object categories. *Computer Vision and Pattern Recognition Workshop (CVPRW)*, 2004. Cited on page 49.

Ronald Aylmer Fisher. Dispersion on a sphere. *Proceedings of the Royal Society of London. Series A. Mathematical and Physical Sciences*, 217(1130), 1953. Cited on pages 22 and 50.

Gianni Franchi, Xuanlong Yu, Andrei Bursuc, Angel Tena, Rémi Kazmierczak, Séverine Dubuisson, Emanuel Aldea, and David Filliat. Muad: Multiple uncertainties for autonomous driving, a benchmark for multiple uncertainty types and tasks. *British Machine Vision Conference (BMVC)*, 2022. Cited on pages 16 and 46.

Yarin Gal and Zoubin Ghahramani. Dropout as a Bayesian approximation: Representing model uncertainty in deep learning. In *International Conference on Machine Learning (ICML)*, 2016. Cited on pages 16, 46, and 49.

Ido Galil, Mohammed Dabbah, and Ran El-Yaniv. A framework for benchmarking class-out-of-distribution detection and its application to ImageNet. In *The Eleventh International Conference on Learning Representations (ICLR)*, 2023a. Cited on pages 17 and 32.

Ido Galil, Mohammed Dabbah, and Ran El-Yaniv. What can we learn from the selective prediction and uncertainty estimation performance of 523 ImageNet classifiers? In *International Conference on Learning Representations (ICLR)*, 2023b. Cited on page 43.

Sachin Goyal, Ananya Kumar, Sankalp Garg, Zico Kolter, and Aditi Raghunathan. Finetune like you pretrain: Improved finetuning of zero-shot vision models. In *Computer Vision and Pattern Recognition (CVPR)*, 2023. Cited on page 15.

Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Guo, Mohammad Gheshlaghi Azar, et al. Bootstrap your own latent-a new approach to self-supervised learning. *Neural Information Processing Systems (NeurIPS)*, 2020. Cited on page 16.

Cornelia Gruber, Patrick Oliver Schenk, Malte Schierholz, Frauke Kreuter, and Göran Kauermann. Sources of uncertainty in machine learning–a statisticians' view. *arXiv preprint arXiv:2305.16703*, 2023. Cited on page 17.

Varun Gulshan, Lily Peng, Marc Coram, Martin C Stumpe, Derek Wu, Arunachalam Narayanaswamy, Subhashini Venugopalan, Kasumi Widner, Tom Madams, Jorge Cuadros, et al. Development and validation of a deep learning algorithm for detection of diabetic retinopathy in retinal fundus photographs. *JAMA*, 316(22):2402–2410, 2016. Cited on page 16.

Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q Weinberger. On calibration of modern neural networks. In *International Conference on Machine Learning (ICML)*, 2017. Cited on page 17.

Raia Hadsell, Sumit Chopra, and Yann LeCun. Dimensionality reduction by learning an invariant mapping. In *Computer Vision and Pattern Recognition (CVPR)*, 2006. Cited on page 16.

Manuel Haußmann, Fred A Hamprecht, and Melih Kandemir. Sampling-free variational inference of bayesian neural networks by variance backpropagation. In *Uncertainty in Artificial Intelligence (UAI)*, 2020. Cited on page 17.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Computer Vision and Pattern Recognition (CVPR)*, 2016. Cited on pages 32 and 50.

Eyke Hüllermeier and Willem Waegeman. Aleatoric and epistemic uncertainty in machine learning: An introduction to concepts and methods. *Machine Learning*, 110:457–506, 2021. Cited on pages 28, 39, and 46.

Tony Jebara and Risi Kondor. Bhattacharyya and expected likelihood kernels. In *Learning Theory and Kernel Machines*. 2003. Cited on pages 23 and 49.

Mira Jürgens, Nis Meinert, Viktor Bengs, Eyke Hüllermeier, and Willem Waegeman. Is epistemic uncertainty faithfully represented by evidential deep learning methods? *arXiv preprint arXiv:2402.09056*, 2024. Cited on page 46.

Farnaz Khun Jush, Tuan Truong, Steffen Vogler, and Matthias Lenga. Medical image retrieval using pretrained embeddings. *arXiv preprint arXiv:2311.13547*, 2023. Cited on page 15.

Katie Kang, Amrith Setlur, Claire Tomlin, and Sergey Levine. Deep neural networks tend to extrapolate predictably. *arXiv preprint arXiv:2310.00873*, 2023. Cited on page 28.

Ivan Karpukhin, Stanislav Dereka, and Sergey Kolesnikov. Probabilistic embeddings revisited. *arXiv preprint arXiv:2202.06768*, 2022. Cited on page 17.

Eunji Kim, Dahuin Jung, Sangha Park, Siwon Kim, and Sungroh Yoon. Probabilistic concept bottleneck models. In *International Conference on Machine Learning (ICML)*, 2023. Cited on page 17.

Michael Kirchhof, Karsten Roth, Zeynep Akata, and Enkelejda Kasneci. A non-isotropic probabilistic take on proxy-based deep metric learning. In *European Conference on Computer Vision (ECCV)*, 2022. Cited on pages 22, 23, 49, and 51.

Michael Kirchhof, Enkelejda Kasneci, and Seong Joon Oh. Probabilistic contrastive learning recovers the correct aleatoric uncertainty of ambiguous inputs. *International Conference on Machine Learning (ICML)*, 2023a. Cited on pages 26, 44, 45, 49, 51, and 52.

Michael Kirchhof, Bálint Mucsányi, Seong Joon Oh, and Enkelejda Kasneci. Url: A representation learning benchmark for transferable uncertainty estimates. *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks*, 2023b. Cited on pages 33, 34, 49, 50, and 51.

Michael Kirchhof, Mark Collier, Seong Joon Oh, and Enkelejda Kasneci. Pretrained visual uncertainties. *arXiv preprint arXiv:2402.16569*, 2024. Cited on pages 15, 38, 40, 51, and 52.

Byungsoo Ko, Geonmo Gu, and Han-Gyu Kim. Learning with memory-based virtual classes for deep metric learning. In *International Conference on Computer Vision (ICCV)*, 2021. Cited on page 28.

Jonathan Krause, Michael Stark, Jia Deng, and Li Fei-Fei. 3D object representations for fine-grained categorization. In *Conference on Computer Vision and Pattern Recognition (CVPR) Workshop*, 2013. Cited on pages 24, 39, and 49.

Alex Krizhevsky. Learning multiple layers of features from tiny images. Technical report, 2009. Cited on page 49.

Alexander Kurz, Katja Hauser, Hendrik Alexander Mehrtens, Eva Krieghoff-Henning, Achim Hekler, Jakob Nikolas Kather, Stefan Fröhling, Christof von Kalle, and Titus Josef Brinker. Uncertainty estimation in medical image classification: systematic review. *JMIR Medical Informatics*, 10(8):e36427, 2022. Cited on page 16.

Salem Lahlou, Moksh Jain, Hadi Nekoei, Victor I Butoi, Paul Bertin, Jarrid Rector-Brooks, Maksym Korablyov, and Yoshua Bengio. DEUP: Direct epistemic uncertainty prediction. *Transactions on Machine Learning Research (TMLR)*, 2023. ISSN 2835-8856. Cited on pages 17 and 45.

Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. Simple and scalable predictive uncertainty estimation using deep ensembles. *Advances in Neural Information Processing Systems (NeurIPS)*, 2017. Cited on pages 16 and 46.

Max-Heinrich Laves, Sontje Ihler, Jacob F Fast, Lüder A Kahrs, and Tobias Ortmaier. Well-calibrated regression uncertainty in medical imaging with deep learning. In *Medical Imaging with Deep Learning*, pages 393–412. PMLR, 2020. Cited on pages 17 and 34.

Kimin Lee, Kibok Lee, Honglak Lee, and Jinwoo Shin. A simple unified framework for detecting out-of-distribution samples and adversarial attacks. *Neural Information Processing Systems (NeurIPS)*, 2018. Cited on page 17.

Dan Levi, Liran Gispan, Niv Giladi, and Ethan Fetaya. Evaluating and calibrating uncertainty prediction in regression tasks. *Sensors*, 22(15):5540, 2022. Cited on page 34.

Shen Li, Jianqing Xu, Xiaqing Xu, Pengcheng Shen, Shaoxin Li, and Bryan Hooi. Spherical confidence learning for face recognition. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021. Cited on page 23.

Jeremiah Liu, Zi Lin, Shreyas Padhy, Dustin Tran, Tania Bedrax Weiss, and Balaji Lakshminarayanan. Simple and principled uncertainty estimation with deterministic deep learning via distance awareness. *Advances in Neural Information Processing Systems (NeurIPS)*, 2020. Cited on page 50.

Zheyuan Liu, Cristian Rodriguez-Opazo, Damien Teney, and Stephen Gould. Image retrieval on real-life images with pre-trained vision-and-language models. In *International Conference on Computer Vision (ICCV)*, 2021. Cited on page 15.

David John Cameron Mackay. *Bayesian methods for adaptive models, PhD thesis*. California Institute of Technology, 1992. Cited on page 16.

Kanti V Mardia and Peter E Jupp. *Directional statistics*. Wiley Series in Probability and Statistics. John Wiley & Sons, 2009. Cited on page 22.

Max Marion, Ahmet Üstün, Luiza Pozzobon, Alex Wang, Marzieh Fadaee, and Sara Hooker. When less is more: Investigating data pruning for pretraining llms at scale. *arXiv preprint arXiv:2309.04564*, 2023. Cited on page 45.

Leland McInnes, John Healy, Nathaniel Saul, and Lukas Großberger. UMAP: Uniform manifold approximation and projection. *Journal of Open Source Software*, 3(29):861, 2018. doi: 10.21105/joss.00861. URL `https://doi.org/10.21105/joss.00861`. Cited on page 46.

Sören Mindermann, Jan M Brauner, Muhammed T Razzak, Mrinank Sharma, Andreas Kirsch, Winnie Xu, Benedikt Höltgen, Aidan N Gomez, Adrien Morisot, Sebastian Farquhar, et al. Prioritized training on points that are learnable, worth learning, and not yet learnt. In *International Conference on Machine Learning (ICML)*, 2022. Cited on page 45.

Yair Movshovitz-Attias, Alexander Toshev, Thomas K Leung, Sergey Ioffe, and Saurabh Singh. No fuss distance metric learning using proxies. In *International Conference on Computer Vision (ICCV)*, 2017. Cited on pages 22 and 49.

Bálint Mucsányi, Michael Kirchhof, Elisa Nguyen, Alexander Rubinstein, and Seong Joon Oh. Trustworthy machine learning. *arXiv preprint arXiv:2310.08215*, 2023. Cited on page 14.

Bálint Mucsányi, Michael Kirchhof, and Seong Joon Oh. Benchmarking uncertainty disentanglement: Specialized uncertainties for specialized tasks. *arXiv preprint arXiv:2402.19460*, 2024. Cited on pages 17, 19, 40, 41, and 46.

Jishnu Mukhoti, Andreas Kirsch, Joost van Amersfoort, Philip HS Torr, and Yarin Gal. Deep deterministic uncertainty: A new simple baseline. In *Computer Vision and Pattern Recognition (CVPR)*, 2023. Cited on page 17.

Hiroki Nakamura, Masashi Okada, and Tadahiro Taniguchi. Representation uncertainty in self-supervised learning as variational inference. In *International Conference on Computer Vision (ICCV)*, 2023. Cited on page 17.

Yuval Netzer, Tao Wang, Adam Coates, Alessandro Bissacco, Bo Wu, and Andrew Y Ng. Reading digits in natural images with unsupervised feature learning. 2011. Cited on page 50.

Vu-Linh Nguyen, Mohammad Hossein Shaker, and Eyke Hüllermeier. How to measure uncertainty in uncertainty sampling for active learning. *Machine Learning*, 111(1):89–122, 2022. Cited on page 16.

M-E. Nilsback and A. Zisserman. Automated flower classification over a large number of classes. In *Proceedings of the Indian Conference on Computer Vision, Graphics and Image Processing*, 2008. Cited on page 49.

Seong Joon Oh, Andrew C. Gallagher, Kevin P. Murphy, Florian Schroff, Jiyan Pan, and Joseph Roth. Modeling uncertainty with hedged instance embeddings. In *International Conference on Learning Representations (ICLR)*, 2019. Cited on pages 17, 28, and 49.

Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018. Cited on pages 27 and 49.

Guillermo Ortiz-Jimenez, Mark Collier, Anant Nawalgaria, Alexander Nicholas D'Amour, Jesse Berent, Rodolphe Jenatton, and Efi Kokiopoulou. When does privileged information explain away label noise? In *International Conference on Machine Learning (ICML)*, 2023. Cited on page 45.

Yaniv Ovadia, Emily Fertig, Jie Ren, Zachary Nado, David Sculley, Sebastian Nowozin, Joshua Dillon, Balaji Lakshminarayanan, and Jasper Snoek. Can you trust your model's uncertainty? Evaluating predictive uncertainty under dataset shift. *Neural Information Processing Systems (NeurIPS)*, 2019. Cited on pages 17 and 32.

Jaewoo Park, Jacky Chen Long Chai, Jaeho Yoon, and Andrew Beng Jin Teoh. Understanding the feature norm for out-of-distribution detection. In *International Conference on Computer Vision (ICCV)*, 2023. Cited on page 17.

O. M. Parkhi, A. Vedaldi, A. Zisserman, and C. V. Jawahar. Cats and dogs. In *Computer Vision and Pattern Recognition (CVPR)*, 2012. Cited on pages 15, 49, and 51.

Joshua C Peterson, Ruairidh M Battleday, Thomas L Griffiths, and Olga Russakovsky. Human uncertainty makes classification more robust. In *International Conference on Computer Vision (CVPR)*, pages 9617–9626, 2019. Cited on page 28.

Janis Postels, Mattia Segù, Tao Sun, Luca Daniel Sieber, Luc Van Gool, Fisher Yu, and Federico Tombari. On the practicality of deterministic epistemic uncertainty. In *International Conference on Machine Learning (ICML)*, 2022. Cited on page 17.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International Conference on Machine Learning (ICML)*, 2021. Cited on page 16.

Aditya Ramesh, Mikhail Pavlov, Gabriel Goh, Scott Gray, Chelsea Voss, Alec Radford, Mark Chen, and Ilya Sutskever. Zero-shot text-to-image generation. In *International Conference on Machine Learning (ICML)*, 2021. Cited on page 15.

Rajeev Ranjan, Carlos D Castillo, and Rama Chellappa. L2-constrained softmax loss for discriminative face verification. *arXiv preprint arXiv:1703.09507*, 2017. Cited on pages 21 and 28.

Patrik Reizinger, Yash Sharma, Matthias Bethge, Bernhard Schölkopf, Ferenc Huszár, and Wieland Brendel. Jacobian-based causal discovery with nonlinear ica. *Transactions on Machine Learning Research (TMLR)*, 2022. Cited on page 18.

Hippolyt Ritter, Aleksandar Botev, and David Barber. A scalable laplace approximation for neural networks. In *International Conference on Representation Learning (ICLR)*, 2018. Cited on page 16.

Karsten Roth, Timo Milbich, Samarth Sinha, Prateek Gupta, Bjorn Ommer, and Joseph Paul Cohen. Revisiting training strategies and generalization performance in deep metric learning. In *International Conference on Machine Learning (ICML)*, 2020. Cited on page 21.

Noveen Sachdeva, Benjamin Coleman, Wang-Cheng Kang, Jianmo Ni, Lichan Hong, Ed H Chi, James Caverlee, Julian McAuley, and Derek Zhiyuan Cheng. How to train data-efficient llms. *arXiv preprint arXiv:2402.09668*, 2024. Cited on page 45.

Lars Schmarje, Vasco Grossmann, Claudius Zelenka, Sabine Dippel, Rainer Kiko, Mariusz Oszust, Matti Pastell, Jenny Stracke, Anna Valros, Nina Volkmann, et al. Is one annotation enough? A data-centric image classification benchmark for noisy and ambiguous label estimation. *arXiv preprint arXiv:2207.06214*, 2022. Cited on pages 13, 29, 33, and 50.

Florian Schroff, Dmitry Kalenichenko, and James Philbin. Facenet: A unified embedding for face recognition and clustering. In *Computer Vision and Pattern Recognition (CVPR)*, 2015. Cited on page 16.

Christoph Schuhmann, Richard Vencu, Romain Beaumont, Robert Kaczmarczyk, Clayton Mullis, Aarush Katta, Theo Coombes, Jenia Jitsev, and Aran Komatsuzaki. Laion-400m: Open dataset of clip-filtered 400 million image-text pairs. *arXiv preprint arXiv:2111.02114*, 2021. Cited on page 45.

Tyler R Scott, Karl Ridgeway, and Michael C Mozer. Stochastic prototype embeddings. *International Conference on Machine Learning (ICML) Workshop on Uncertainty and Robustness in Deep Learning*, 2019. Cited on page 17.

Tyler R Scott, Andrew C Gallagher, and Michael C Mozer. von Mises-Fisher loss: An exploration of embedding geometries for supervised learning. In *International Conference on Computer Vision (ICCV)*, 2021. Cited on pages 21 and 28.

Burr Settles. Active learning literature survey. *University of Wisconsin-Madison Department of Computer Sciences*, 2009. Cited on page 16.

Kihyuk Sohn. Improved deep metric learning with multi-class n-pair loss objective. *Neural Information Processing Systems (NeurIPS)*, 2016. Cited on page 21.

Hyun Oh Song, Yu Xiang, Stefanie Jegelka, and Silvio Savarese. Deep metric learning via lifted structured feature embedding. In *Computer Vision and Pattern Recognition (CVPR)*, 2016. Cited on pages 39 and 50.

Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research (JMLR)*, 15(1):1929–1958, 2014. Cited on page 17.

Eu Wern Teh, Terrance DeVries, and Graham W Taylor. ProxyNCA++: Revisiting and revitalizing proxy neighborhood component analysis. In *European Conference on Computer Vision (ECCV)*, pages 448–464, 2020. Cited on page 22.

Dustin Tran, Jeremiah Liu, Michael W Dusenberry, Du Phan, Mark Collier, Jie Ren, Kehang Han, Zi Wang, Zelda Mariet, Huiyi Hu, et al. Plex: Towards reliability using pretrained large model extensions. *arXiv preprint arXiv:2207.07411*, 2022. Cited on pages 17, 43, and 45.

Uddeshya Upadhyay, Shyamgopal Karthik, Massimiliano Mancini, and Zeynep Akata. ProbVLM: Probabilistic adapter for frozen vison-language models. In *International Conference on Computer Vision (ICCV)*, 2023a. Cited on page 44.

Uddeshya Upadhyay, Jae Myung Kim, Cordelia Schmidt, Bernhard Schölkopf, and Zeynep Akata. Posterior annealing: Fast calibrated uncertainty for regression. *arXiv preprint arXiv:2302.11012*, 2023b. Cited on page 34.

Matias Valdenegro-Toro and Daniel Saromo Mori. A deeper look into aleatoric and epistemic uncertainty disentanglement. In *Computer Vision and Pattern Recognition Workshops (CVPRW)*, 2022. Cited on pages 17 and 46.

Joost Van Amersfoort, Lewis Smith, Yee Whye Teh, and Yarin Gal. Uncertainty estimation using a single deep deterministic neural network. In *International Conference on Machine Learning (ICML)*, 2020. Cited on page 17.

Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-SNE. *Journal of Machine Learning Research (JMLR)*, 9(86):2579–2605, 2008. Cited on page 46.

C. Wah, S. Branson, P. Welinder, P. Perona, and S. Belongie. The Caltech-UCSD birds-200-2011 dataset. Technical Report CNS-TR-2011-001, California Institute of Technology, 2011. Cited on pages 24, 39, and 49.

Karl Weiss, Taghi M Khoshgoftaar, and DingDing Wang. A survey of transfer learning. *Journal of Big data*, 3(1):1–40, 2016. Cited on page 15.

Ross Wightman. PyTorch image models. *GitHub repository*, 2019. doi: 10.5281/zenodo.4414861. Cited on page 32.

Lisa Wimmer, Yusuf Sale, Paul Hofman, Bernd Bischl, and Eyke Hüllermeier. Quantifying aleatoric and epistemic uncertainty in machine learning: Are conditional entropy and mutual information appropriate measures? In *Uncertainty in Artificial Intelligence (UAI)*, 2023. Cited on pages 17, 19, 40, and 46.

J. Xiao, J. Hays, K. A. Ehinger, A. Oliva, and A. Torralba. Sun database: Large-scale scene recognition from abbey to zoo. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2010. Cited on page 50.

Donggeun Yoo and In So Kweon. Learning loss for active learning. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. Cited on pages 17, 34, and 38.

Tianhe Yu, Saurabh Kumar, Abhishek Gupta, Sergey Levine, Karol Hausman, and Chelsea Finn. Gradient surgery for multi-task learning. *Neural Information Processing Systems (NeurIPS)*, 2020. Cited on pages 38 and 49.

Xiaohua Zhai, Joan Puigcerver, Alexander Kolesnikov, Pierre Ruyssen, Carlos Riquelme, Mario Lucic, Josip Djolonga, Andre Susano Pinto, Maxim Neumann, Alexey Dosovitskiy, Lucas Beyer, Olivier Bachem, Michael Tschannen, Marcin Michalski, Olivier Bousquet, Sylvain Gelly, and Neil Houlsby. A large-scale study of representation learning with the visual task adaptation benchmark. *arXiv preprint arXiv:1910.04867*, 2020. Cited on pages 39 and 50.

Roland S. Zimmermann, Yash Sharma, Steffen Schneider, Matthias Bethge, and Wieland Brendel. Contrastive learning inverts the data generating process. In *Proceedings of the 38th International Conference on Machine Learning (ICML)*, 2021. Cited on pages 18, 25, 26, and 28.

A
A non-isotropic probabilistic take on proxy-based deep metric learning

This appendix contains the full paper and appendix discussed in Chapter 2, reproduced with permission.

Michael Kirchhof, Karsten Roth, Zeynep Akata, and Enkelejda Kasneci. A non-isotropic probabilistic take on proxy-based deep metric learning, 2022. First published in: Avidan, S., Brostow, G., Cissé, M., Farinella, G.M., Hassner, T. (eds) Computer Vision – ECCV 2022. ECCV 2022. Lecture Notes in Computer Science, vol 13686, pp 435–454, by Springer Nature. DOI: `https://doi.org/10.1007/978-3-031-19809-0_25`. Reproduced with permission from Springer Nature.

A Non-isotropic Probabilistic Take on Proxy-based Deep Metric Learning

Michael Kirchhof(✉), Karsten Roth, Zeynep Akata, and Enkelejda Kasneci

University of Tübingen, Tübingen, Germany
michael.kirchhof@uni-tuebingen.de

Abstract. Proxy-based Deep Metric Learning (DML) learns deep representations by embedding images close to their class representatives (*proxies*), commonly with respect to the angle between them. However, this disregards the embedding norm, which can carry additional beneficial context such as class- or image-intrinsic uncertainty. In addition, proxy-based DML struggles to learn class-internal structures. To address both issues at once, we introduce non-isotropic probabilistic proxy-based DML. We model images as directional von Mises-Fisher (vMF) distributions on the hypersphere that can reflect image-intrinsic uncertainties. Further, we derive non-isotropic von Mises-Fisher (nivMF) distributions for class proxies to better represent complex class-specific variances. To measure the proxy-to-image distance between these models, we develop and investigate multiple distribution-to-point and distribution-to-distribution metrics. Each framework choice is motivated by a set of ablational studies, which showcase beneficial properties of our probabilistic approach to proxy-based DML, such as uncertainty-awareness, better behaved gradients during training, and overall improved generalization performance. The latter is especially reflected in the competitive performance on the standard DML benchmarks, where our approach compares favourably, suggesting that existing proxy-based DML can significantly benefit from a more probabilistic treatment. Code is available at http://github.com/ExplainableML/Probabilistic_Deep_Metric_Learning.

Keywords: Deep metric learning · von Mises-Fisher · Non-isotropy · Probablistic embeddings · Uncertainty

1 Introduction

Understanding and encoding visual similarity is a key concept that drives applications ranging from image (video) retrieval [3,27,60,65,70] to clustering [1] and

M. kirchhof and K. Roth—Equal contribution.

Supplementary Information The online version contains supplementary material available at https://doi.org/10.1007/978-3-031-19809-0_25.

S. Avidan et al. (Eds.): ECCV 2022, LNCS 13686, pp. 435–454, 2022.
https://doi.org/10.1007/978-3-031-19809-0_25

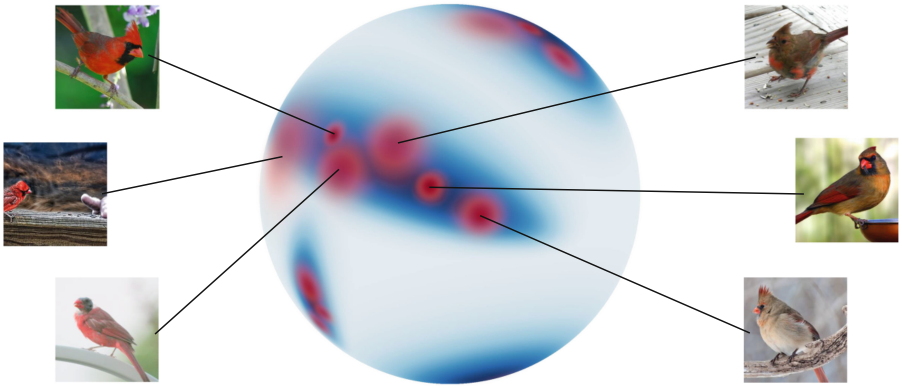


Fig. 1. Class proxy distributions (blue (Color figure online)) and image distributions (red) embedded on the 3D unit sphere. The central proxy has a non-isotropic variance, so it can represent the high variance in body color between male (left) and female (right) cardinals and the low variance in their beak shape (top to bottom). Ambiguous images (e.g. middle left) have higher variance than images that clearly show class-discriminating features (top left, middle right). Best viewed in color.

face re-identification [9, 20, 33, 56]. Most commonly, approaches leverage Deep Metric Learning (DML) [40, 52, 56, 60, 70] to reformulate visual similarity learning into a surrogate, contrastive representation learning problem: Here, a deep network is tasked to embed images such that a simple predefined distance metric over pairs of embeddings represents their actual semantic relations. Similar contrastive learning is used for representation learning tasks s.a. supervised image classification [25] or self-supervised learning [5, 18]. Common DML approaches are formulated as ranking tasks over data tuples (e.g. pairs [15], triplets [56] or quadruplets [6]) of similar and dissimilar samples. Unfortunately, the complexity of sampling such tuples grows exponentially with the tuples size [70]. This has motivated recent advances in DML to focus on *proxy-based* approaches, where the similar samples are summarized into learnable proxy representations [40, 47] against which the sample embeddings are contrasted.

While this allows for fast convergence and reliable generalization, drawbacks may arise both in the treatment of proxies and samples: Firstly, the deterministic treatment of sample representations does not offer any degrees of freedom to address ambiguities and uncertainty (e.g., an image of a bird covered by branches). Secondly, isotropic distance scores between proxy-sample pairs (e.g., cosine similarity) provide only limited tools for the network to derive the similarity of samples within a class, as the distance to each proxy alone is insufficient to resolve relative sample placements around a proxy. This hinders class-specific variance and substructures to be successfully accounted for, which have been shown to notably benefit downstream generalization performance [37, 52].

To address these issues, we propose a *probabilistic* interpretation of proxy-based DML. Driven by the fact that modern DML consistently operates on hyperspherical (i.e., normalized) representations [52, 70], we derive hyperspherical von-Mises Fisher (vMF) distributions for each sample. A sample embedding's direction controls the placement on the hypersphere, and therefore its semantic content, and its norm parametrizes the certainty of the distribution. In conjunction, we also treat class proxies probabilistically, but through *non-isotropic* vMF distributions. This enforces the distributional prior over each class proxy to explicitly account for different, non-isotropic distributions, capturing more complex class-specific sample distributions (c.f. Figure 1). As this moves the DML training from point-based to distributional comparisons, we merge both components into a sound setup by motivating distribution-to-distribution matching metrics based on probabilistic product kernels. Our full framework is supported through an extensive set of derivations and experimental ablations that showcase and support how the extension to probabilistic proxy-based DML offers significant improvements, with competitive performance across the standard DML benchmarks – CUB200-2011 [64], CARS196 [30], and Stanford Online Products [42] – even when compared to much more complex training methods.

Overall, our contributions can be summarized as: **(1)** We propose and derive a novel probabilistic interpretation of proxy-based DML to account for sample and class ambiguities by reformulating the standard proxy-based metric learning approach to a distributional one on the hypersphere. **(2)** We extend the vMF model to a non-isotropical one for each class proxy to better incorporate and address intra-class substructures for better generalization. **(3)** We introduce various distribution-to-distribution metrics for DML and contrast them to traditional point-to-point metrics. **(4)** We support our proposed framework through various derivational and experimental ablations showcasing how a distributional treatment can positively impact the learned representation spaces. **(5)** Finally, we benchmark against standard DML approaches and provide further significant experimental support for our probabilistic approach to proxy-based DML.

2 Related Work

Deep Metric Learning comprises several conceptually different approaches. Firstly, one can define ranking tasks over data tuples such as pairs [15,70], triplets [56], quadruplets [6] or higher-order variants [42,60,67]. An underlying network then learns to solve each tuple presented by learning a representation space in which distances between embeddings correctly reflect their respective semantics/labelling. However, as the sizes of presented tuples increase, so does the tuple space each ranking task is sampled from, resulting in notable redundancy and impacted convergence behaviour [52,56,70]. As a result, a secondary branch evolved focusing on heuristics which target ranking tuples fulfilling a set of predefined [56,67,70,71] or learned [16,50] criteria. In a similar vein, DML research has also tried to address the sampling complexity issue through the replacement of tuple components with learned concept representations denoted as *proxies*, with some approaches leveraging proxies in a classification-style setting [9,73] or in a ranking fashion, where each sample is contrasted against a respective proxy [26,40,47,63]. Finally, benefits have also been found in orthogonal extensions and fundamental improvements to the general DML training pipeline, through various different approaches such as the usage of adversarial training [10], synthetic samples [32,75], higher-order or curvilinear metric learning [4,21], feature mining for ranking [37,38,49] or proxy-based [54] approaches, a breakdown of the overall metric space into subspaces [43,44,55], orthogonal modalities [53] or knowledge distillation [51]. Our proposed probabilistic proxy-based DML falls into this line of work, but is orthogonal to these other approaches, as these extensions can be applied in a method-agnostic fashion. In particular, we extend proxy-based DML by specifically accounting for sample and class ambiguity through a distributional treatment of samples and proxies, and by utilizing non-isotropic proxy distributions to encourage more complex intra-class distributions around each proxy, which has been shown to be beneficial for generalization [39,52,54].

Probabilistic Embeddings. Various approaches to DML can already be framed from a more probabilistic standpoint, where softmax-based approaches on the basis of cosine similarities [9,63,73] can be seen as analytical class posteriors if each class assumes a von Mises-Fisher (vMF) distribution [45,74]. While these methods implicitly model classes as vMFs, probabilistic embedding approaches further model each sample as a distribution in the embedding space [31,57,58]. This allows the model to express uncertainty when images are ambiguous. Recent works argue that this ambiguity is captured in the image embedding's norm [31,48,57]: [57] argues that the embedding of an image that shows many class-discriminative features of one class consists of several vectors that all point in the same direction, resulting in a higher norm. On this basis, [31,57] pioneered the use of embedding direction and norm to model each image as a vMF distribution, in particular for supervised classification. Utilizing vMF distributions, we are the first to introduce a full probabilistic proxy-based DML framework, yielding distribution-to-distribution metrics. Additionally, we propose a non-isotropic vMF for proxy distributions, which allows us to represent richer class structures in the embedding space beneficial to generalization [37,52].

3 Non-isotropic Probabilistic Proxy-based DML

3.1 A Probabilistic Interpretation of Proxy-based DML

In this section, we extend the common DML framework to a probabilistic one. Fundamentally, DML aims to find embedding functions $e : \mathcal{X} \rightarrow \mathcal{E}$ from image $\mathcal{X} \subset \mathbb{R}^{H \times W \times 3}$ to M-dimensional metric embedding spaces $\mathcal{E} \subset \mathbb{R}^M$ such that a distance function $d : \mathcal{E} \times \mathcal{E} \rightarrow \mathbb{R}$ between embeddings $z_1 = e(x_1)$ and $z_2 = e(x_2)$ of images $x_1, x_2 \in \mathcal{X}$ reflects the semantic relation between them. The embedding space $\mathcal{E}$ is chosen to be the M-dimensional unit hypersphere $\mathcal{E} = \mathcal{S}^{M-1}$, i.e. $\|z\| = 1$. While an euclidean $\mathcal{E}$ might appear more natural, recent works in DML [26,51,52,67,70] and other contrastive learning domains like self-supervised learning [5,8,18,66] have seen significant benefits in a directional treatment through normalization of embeddings to the unit hypersphere. This can in parts be attributed to better scaling with increased embedding dimensions [68] and semantic information being mostly directionally encoded [48]. To learn the respective embedding space $\mathcal{E}$, DML commonly employs ranking objectives over sample tuples. Based on the class assignments for each sample, an embedding network is tasked to minimize distances between same-class samples while maximizing them when classes differ. More recently, proxy-based approaches [26,40,47,63] directly model the class assignments by introducing class representatives during training – the proxies $\mathrm{p} \in \mathcal{S}^{M-1}$. These are contrasted against the sample embeddings $e(x) = z$ using a NCA-like [14] formulation (ProxyNCA, [40]), which was slightly modified by [63] as a softmax-loss

$$\mathcal{L}_{\text{NCA++}} = \log \frac{\exp(-d(\mathrm{p}^*, z)/t)}{\sum_{c=1}^{C} \exp(-d(\mathrm{p}_c, z)/t)} \,. \tag{1}$$

Here, p^* denotes the ground-truth proxy associated with z, t a temperature, and d a distance metric, most commonly the negative cosine similarity $d = -s$ with $s(p_c, z) = (p_c z)/(\|p_c\|\|z\|)$. This implies a problematic assumption: Since only angles between samples and proxies are leveraged, class-specific distribution variances around each proxy cannot be accounted for. Second, the deterministic underlying network e induces a Dirac delta distribution over sample representations [59]. This treats all the input data the same regardless of the level of ambiguity, not accounting for sample-specific uncertainties.

Therefore, we suggest to represent samples and proxies as random variables Z and P with densities ζ and ρ on $\mathcal{S}^{M-1}$, which allows both samples and proxies to carry uncertainty context to address sample ambiguity while encouraging to account for more complex class distributions. This converts the above loss to

$$\mathcal{L} = \log \frac{\exp(-d(\rho^*, \zeta)/t)}{\sum_{c=1}^{C} \exp(-d(\rho_c, \zeta)/t)}. \tag{2}$$

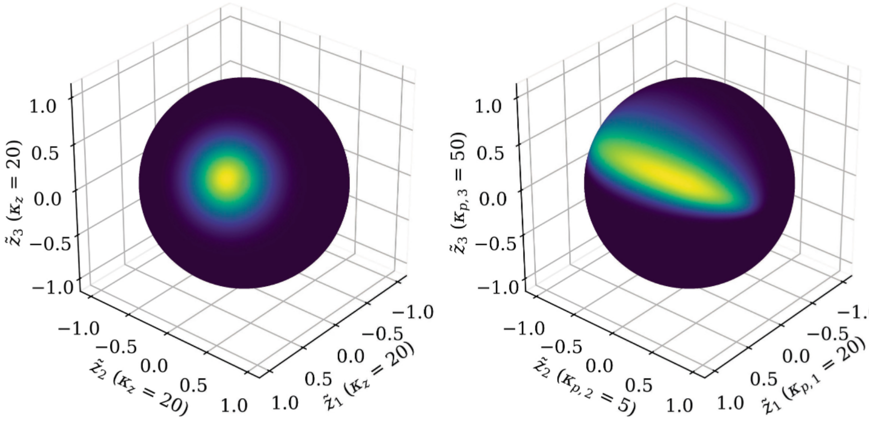
Below in Sect. 3.2, we discuss how precisely ρ and ζ are parametrized, and in Sect. 3.3, we find a $d(\cdot, \cdot)$ suitable for distribution-to-distribution matching .

3.2 Probabilistic Sample and Proxy Representations

Sample Representations. A common distribution on $\mathcal{S}^{M-1}$ is the von Mises-Fisher (vMF) distribution [13,35,79]. It parametrizes the sample distribution ζ by a direction vector $\mu_z \in \mathcal{S}^{M-1}$ that points towards the mode of the distribution and a concentration parameter $\kappa_z \in \mathbb{R}_{\geq 0}$ that controls the spread around the mode, where a higher κ_z yields a sharper distribution. The density ζ of a vMF-distributed sample $Z \sim \text{vMF}(\mu_z, \kappa_z)$ at a point $\tilde{z} \in \mathcal{S}^{M-1}$ is

$$\zeta(\tilde{z}) = C_M(\kappa_z) \exp\left(\kappa_z \, s(\tilde{z}, \mu_z)\right). \tag{3}$$

C_M is the normalizing function which we approximate in high-dimensions (see Supp. ??). The advantage of the vMF is a duality to the un-normalized image embeddings $z = e(x) \in \mathbb{R}^M$: The natural parameter of the vMF is $\nu_z = \kappa_z \mu_z \in \mathbb{R}^M$, such that if we set $\mu_z = \frac{z}{\|z\|}$ and $\kappa_z = \|z\|$, the embedding norm gives the vMF concentration without needing to explicitly predict it (as necessary for normal distribution [7,58]). This is further motivated by recent findings indicating that CNNs encode the amount of visible class discriminative features in the norm of the embedding (e.g. [57]). We validate this assumption in Sect. 4.4.



(a) vMF, $\kappa_z = 20$

(b) nivMF, $\kappa_p = (20, 5, 50)$

Fig. 2. Densities of (a) vMF and (b) non-isotropic vMF distributions on $\mathcal{S}^2$. The density is proportional to the color gradient from violet (zero) to yellow (high).

Proxy Representations. It is possible to analogously treat the proxy distributions ρ as vMF distributions with parameters $\nu_\rho = \kappa_\rho \mu_\rho$. However, a limiting factor owed to the simplicity of the vMF is its isotropy: The vMF is equivariant in all directions as shown in Fig. 2a. Proxies, however, need to account for more complex class distributions, i.e., non-isotropic ones (c.f. Figure 2b). Generalized families of vMF distributions, such as Fisher-Bingham or Kent distributions [24,35,36], are able to capture non-isotropy. However, they use covariance matrices with a quadratic number of parameters and constraints on their eigenvectors. This complicates their training via gradient descent, especially in high dimensions. Hence, we propose a low-parameter vMF extension called non-isotropic von Mises-Fisher distribution (nivMF). Just like the vMF, the M-dimensional nivMF of a proxy p is parametrized by a direction $\mu_p \in \mathcal{S}^{M-1}$, but its concentration is described by a concentration *matrix* $K_p \in \mathbb{R}^{(M \times M)}$. To reduce its parameters, we assume $K_p = \text{diag}(\kappa_p) = \text{diag}(\kappa_{p,1}, \ldots, \kappa_{p,M})$ to be a diagonal matrix where $\kappa_{p,m} > 0, m = 1, \ldots, M$, gives the concentration per dimension. They are treated as learnable parameters (see Supp.??). Then, we define the density ρ of a nivMF distributed proxy P $\sim$ nivMF(μ_p, K_p) at a point $\tilde{z} \in \mathcal{S}^{M-1}$ as

$$\rho = f_P(\tilde{z}) := C_M(\|K_p \mu_p\|)\, D(K_p)\, \exp\left(\|K_p \mu_p\|\, s(K_p \tilde{z}, K_p \mu_p)\right). \tag{4}$$

with vMF normalizer C_M, and D approximating an additional normalizing constant (see Supp. ??). Intuitively, the nivMF is obtained from a vMF by a change-of-variable transformation: The unit sphere is stretched into an ellipsoid with axis lengths $\kappa_m, m = 1, \ldots, M$, before the angle to the mode μ_p is measured. Thus, distances of $\tilde{z}$ to μ_p along dimensions with high concentrations are emphasized and distances along dimensions with low concentrations are weighted less. In effect, the M-dim K_p is projected onto the $(M-1)$-dim tangential plane of μ_p and controls the density's spherical shape (see Fig. 2b). The remaining concentration projected on the μ_p-axis, i.e., $\|K_p \mu_p\|$, controls the density's peakedness, analogously to the κ parameter from a standard vMF. Thus, when $K_p = cI_M$ is the identity matrix scaled by some $c > 0$, the nivMF simplifies to a vMF (up to a constant due to an approximation, see Supp. ??).

3.3 Comparing Distributions Instead of Points

As proxies and images are no longer modeled as points but as distributions, we present several distribution-to-distribution metrics (in the sense of distance functions d in DML – formally, they are no metrics as they don't fulfill the triangle inequality) and contrast them to traditional distribution-to-point metrics.

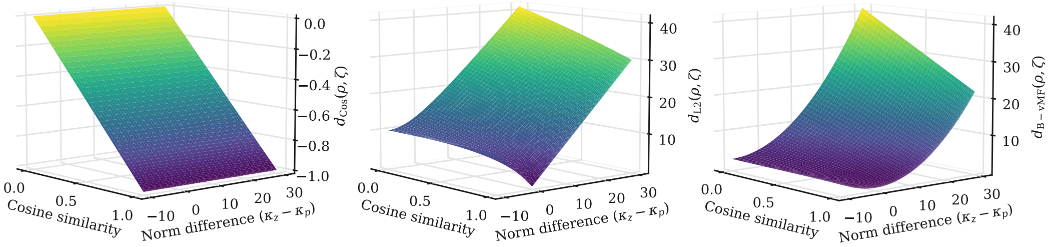
Distribution-to-Distribution Metrics. Probability product kernels (PPK) [22] are a family of metrics to compare two distributions ρ and ζ by the product of their densities. One member of this family is the expected likelihood kernel (or mutual likelihood score [58]). Although there is no analytical solution for nivMFs, we can derive a Monte-Carlo approximation

$$d_{\text{EL-nivMF}}(\rho, \zeta) := -\log\left(\int_{\mathcal{E}} \rho(a) d\zeta(a)\right) \approx -\log\left(\frac{1}{N} \sum_{\substack{i=1,\ldots,N \\ z_i \sim \zeta}} \rho(z_i)\right), \tag{5}$$

where N is the number of samples. Similar to [57], we empirically found that a low number of samples ($N = 5$) is sufficient. We use [8] to sample from ζ.

The expected likelihood kernel is advantageous since it is easily Monte-Carlo approximated, but there are other distribution-to-distribution metrics we would like to survey. Hence, we derive them under a vMF assumption for ρ, where they have analytical solutions (see Supp. ??). Namely, these are an analogous expected likelihood kernel $d_{\text{EL-vMF}}$, a related PPK kernel $d_{\text{B-vMF}}$, and a Kullback-Leibler distance $d_{\text{KL-vMF}}$. All three implicitly use the norm of the image embeddings in their calculations to respect the ambiguity, but differ in performance (see Sect. 4.3).

Distribution-to-point Metrics. Classical metrics like the cosine distance of the loss in Eq. 1 implicitly assume a distribution for each proxy and evaluate its log-likelihood at each sample. Hence, we will refer to them as distribution-to-point metrics. E.g., the cosine metric used in Eq. 1 is equivalent to the log-likelihood of the normalized sample embedding under vMF-distributed proxies with equal concentration values [17], i.e., $d_{\text{Cos}}(\rho, \zeta) := -s(\mu_p, \mu_z) =$



(a) Cosine distance (b) L_2 distance (c) Bhattacharyya distance

Fig. 3. Distances of a sample embedding to a vMF-distributed proxy with norm $\kappa_p = 10$. (a) and (b) treat the sample as a point and (c) as a vMF distribution.

$-\log(\rho(\mu_z))$. Another common example is the L2-distance $d_{\mathrm{L2}}(\rho,\zeta) := (\nu_p - \nu_z)^2 = -\log(\rho(\nu_z))$ which is obtained by an equivariance normal distribution assumption for ρ. We analogously define $d_{\mathrm{nivMF}}(\rho,\zeta) := -\log(\rho(\mu_z))$ under a nivMF assumption for ρ to benchmark it against the $d_{\text{EL-nivMF}}$ distance.

3.4 Probabilistic Proxy-based Deep Metric Learning

Utilizing distributional proxies ρ, distributional sample presentations ζ and the Monte-Carlo approximated Expected Likelihood Kernel $d_{\text{EL-nivMF}}(\rho,\zeta)$ we can fill in Eq. 2 and define the probabilistic extension to proxy-based DML, precisely of the basic ProxyNCA ([63]), as

$$\mathcal{L}^{\text{EL-nivMF}}_{\text{NCA++}} = \log \frac{\exp(-d_{\text{EL-nivMF}}(\rho^*,\zeta)/t)}{\sum_{c=1}^{C} \exp(-d_{\text{EL-nivMF}}(\rho_c,\zeta)/t)}\,. \tag{6}$$

While this can be used as standalone loss, it can also probabilistically enhance other proxy-based objectives $\mathcal{L}_{\text{Proxy-DML}}$, such as ProxyAnchor [26]. For easy usage in practice, we thus also propose using it as a regularizer via

$$\mathcal{L}^{\text{NCA++}}_{\text{joint}} = \mathcal{L}^{\text{EL-nivMF}}_{\text{NCA++}}(\rho,\zeta) + \omega \cdot \mathcal{L}_{\text{Proxy-DML}}(\mu_\rho,\mu_\zeta) \tag{7}$$

with regularization scale ω. Crucially, μ_ρ and μ_ζ of the proxy and sample distributions are shared parameters with the non-probabilistic objective's proxies. This ensures alignment between the two learned representations spaces. The scaling ω balances the orthogonal benefits of the two approaches: An increasing ω highlights the non-probabilistic objective that encourages a better global alignment of distribution modes, and a decreasing ω yields a continuously more distributional treatment. For the remainder of this work, we use **EL-nivMF** for the standalone probabilistic extension of ProxyNCA (Eq. 6), and *PANC*+**EL-nivMF** for the probabilistically regularized ProxyAnchor (Eq. 7).

3.5 How Uncertainty-awareness Impacts Training

Before the experimental evaluation, we provide an insight into *how* incorporating uncertainty into the training benefits it. For this, we take a closer look at the norms of sample embeddings that, by duality, yield the concentration κ_z of ζ.

Uncertainty as Sample-wise Temperature. Fig. 3 displays two distribution-to-point and one distribution-to-distribution metric with regard to the difference in norms and directions. We use the isotropic $d_{\text{B-vMF}}$ as a representative for distribution-to-distribution metrics since it has an analytical solution. While d_{Cos} ignores the difference in norms, d_{L2} and the similar, yet smoother, $d_{\text{B-vMF}}$ incorporate it as an sample-wise temperature: The larger the norm of the sample gets, the steeper the metrics rise with increasing cosine distance. Thus, when comparing a sample to several proxies of roughly the same norm, their distances to the sample will be more uniform when the sample embedding norm is low and become more contrasted when it is high. In other words, ambiguous images produce more similar logits across all proxies and thus flatter class posterior distributions whereas highly certain images produce sharp posteriors.

Uncertainty as Gradients Scale. κ_z has another influence on the training: Differentiating the losses $\mathcal{L}^{\text{Cos}}_{NCA++}$ and $\mathcal{L}^{\text{L2}}_{\text{NCA++}}$, obtained when using the norm-agnostic d_{Cos} or the norm-aware d_{L2} as distance functions in Eq. 1, w.r.t. the cosine similarity between μ_z and μ_p (as in [26]) reveals (see Supp. ??)

$$\frac{\delta \mathcal{L}^{\text{Cos}}_{NCA++}}{\delta \cos(\mu_p, \mu_z)} = \begin{cases} \frac{1}{t}\left(-1 + \frac{\exp(-d_{\text{Cos}}(\rho^*,\zeta)/t)}{\sum_{c=1}^{C}\exp(-d_{\text{Cos}}(\rho_c,\zeta)/t)}\right) & \text{if } p = p^* \\ \frac{1}{t}\frac{\exp(-d_{\text{Cos}}(\rho^*,\zeta)/t)}{\sum_{c=1}^{C}\exp(-d_{\text{Cos}}(\rho_c,\zeta)/t)} & \text{else} \end{cases} \tag{8}$$

$$\frac{\delta \mathcal{L}^{\text{L2}}_{\text{NCA++}}}{\delta \cos(\mu_p, \mu_z)} = \begin{cases} \frac{2\kappa_p\kappa_z}{t}\left(-1 + \frac{\exp(-d_{\text{L2}}(\rho^*,\zeta)/t)}{\sum_{c=1}^{C}\exp(-d_{\text{L2}}(\rho_c,\zeta)/t)}\right) & \text{if } p = p^* \\ \frac{2\kappa_p\kappa_z}{t}\frac{\exp(-d_{\text{L2}}(\rho^*,\zeta)/t)}{\sum_{c=1}^{C}\exp(-d_{\text{L2}}(\rho_c,\zeta)/t)} & \text{else} \end{cases}, \tag{9}$$

where p^* denotes the ground-truth class. Besides the sample-wise temperature in d_{L2}, the gradients differ in that the gradient of $\mathcal{L}^{\text{L2}}_{\text{NCA++}}$ scales proportionally to κ_z. This means that in batch-wise gradient descent, samples with a high embedding norm are pulled towards ground-truth proxies and pushed away from others stronger than samples with low norm. In other words, the impact of an image on the structuring process of the embedding space depends on its ambiguity. This holds similarly for the distribution-to-distribution metrics, but is harder to derive than for d_{L2}. This analysis unveils that using the Euclidean d_{L2} distance is adequate during training albeit switching to the hyperspherical d_{Cos} at retrieval-time, as it can be seen as a simple approximation to the uncertainty-aware training of hyperspherical distribution-to-distribution metrics.

4 Experiments

We now detail the experiments (Sect. 4.1) that benchmark our method (Sect. 4.2), before surveying different distr.-to-distr. metrics (Sect. 4.3) and the role of the norm (Sect. 4.4).

4.1 Experimental Details

Implementations. All experiments use PyTorch [46]. We follow standard DML protocols by leveraging ImageNet-pretrained ResNet50 [19] and Inception-V1 networks with Batch-Normalization [62] as encoders. Their weights are taken from torchvision [34] and timm [69]. To further ensure standardized training, we built upon the code and standardized DML protocols proposed in [52], using the Adam optimizer [28], a learning rate of 10^{-5} and weight decay of $4 \cdot 10^{-3}$. In the more open state-of-the-art comparison (Table 2), we additionally use step-wise learning rate scheduling. To ensure comparability and access to fast similarity search methods, all test-time retrieval uses cosine distances. To sample from vMF-distributions, we make use of [8] and respective implementations. Further details on our method and hyperparameters are provided in Supp. ??. All experiments were run on NVIDIA 2080Ti GPUs with 12GB VRAM.

Datasets. We benchmark on three standard datasets: CUB200-2011 [64] (has a 100/100 split of train and test bird classes with 11,788 images in total), CARS196 [30] (contains a 98/98 split of car classes and 16,185 images), and Stanford Online Products (SOP) [42] (covers 22,634 product categories and 120,053 images).

Table 1. We re-run various strong benchmarks in the *standardized comparison* setting of [52]. We find strong improvements both when enhancing simple ProxyNCA towards probabilistic DML (**EL-nivMF**) and when using our approach as a regularizer on top of more versatile approaches (*PANC* + **EL-nivMF**).

Benchmarks→	CUB200-2011		CARS196		SOP	
Approaches ↓	R@1	mAP@1000	R@1	mAP@1000	R@1	mAP@1000
Sample-based Baselines.						
Margin [70]	62.9 ± 0.4	32.7 ± 0.3	80.1 ± 0.2	32.7 ± 0.4	78.4 ± 0.1	46.8 ± 0.1
Multisimilarity [67]	62.8 ± 0.2	31.1 ± 0.3	81.6 ± 0.3	31.7 ± 0.1	76.0 ± 0.1	43.3 ± 0.1
Standard versus Probabilistic.						
ProxyNCA [40, 63]	63.2 ± 0.2	33.4 ± 0.1	78.8 ± 0.2	31.9 ± 0.2	76.2 ± 0.1	43.0 ± 0.1
EL-nivMF	64.8 ± 0.4	34.3 ± 0.3	82.1 ± 0.3	33.4 ± 0.2	76.6 ± 0.2	43.3 ± 0.1
Probabilistic DML as Regularization.						
ProxyAnchor (*PANC*, [26])	64.4 ± 0.3	33.2 ± 0.3	82.4 ± 0.4	34.2 ± 0.3	78.0 ± 0.1	45.5 ± 0.1
PANC + **EL-nivMF**	66.5 ± 0.3	35.3 ± 0.1	83.6 ± 0.2	35.1 ± 0.1	78.2 ± 0.1	45.6 ± 0.1

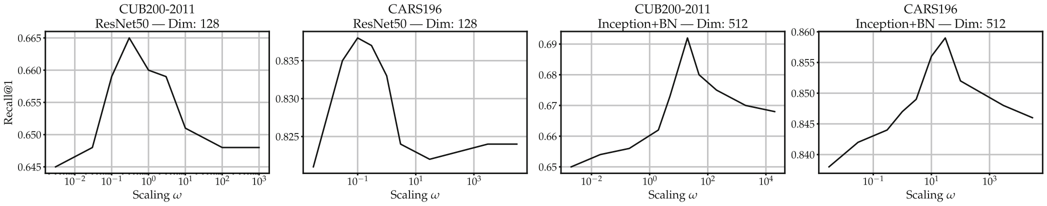


Fig. 4. *Probabilistic regularization as a function of the scaling factor* ω. We find a notable benefit when accounting for both orthogonal enhancements, i.e., the more probabilistic treatment (*decreasing* ω) and the better global alignment of the proxy distribution modes (*increasing* ω).

4.2 Quantitative Evaluation of Probabilistic Proxy-Based DML

Standardized Comparison. We first follow protocols proposed in [52], which suggest comparisons under equal pipeline and implementation settings (and no learning rate scheduling) to determine the true benefits of a proposed method, unbiased by external covariates. Particularily, we thus compare the standard ProxyNCA (see Eq. 1) against our proposed **EL-nivMF** extension of ProxyNCA that includes sample and proxy distributions with distribution-to-distribution metrics during training. We further apply **EL-nivMF** as a probabilistic regularizer on top of the strong, but hyperparameter-heavy ProxyAnchor objective. Here, we only optimize the scaling ω. Finally, we rerun the two strongest sample-based methods used in [52]. In all cases, Table 1 shows significant improvements in performance and outperforms the sample-based methods. First, converting from standard to probabilistic proxy-based DML (ProxyNCA → **EL-nivMF**) increases R@1 on CUB200-2011 by 1.6*pp*, 3.3*pp* on Cars196 and 0.4*pp* on SOP. This highlights the benefits of accounting for uncertainty and explicitly encouraging non-isotropic intra-class variance. However, due to the large number of proxies and low number of samples per class, on SOP benefits are limited when compared to datasets such as CUB200-2011 and CARS196, as the estimation of our proxy distributions becomes noticeably noisier. When using **EL-nivMF** as probabilistic regularization, we find boosts of over 2.1*pp* and 1.2*pp* on CUB200-2011 and CARS196, respectively, with expected smaller improvements of 0.2*pp* on SOP. Generally however, the consistent improvements, whether as a standalone objective or as a regularization method, highlight the versatility of a probabilistic take on DML, and offer a strong proof-of-concept for future DML research to built upon.

Impact of Different Scaling Factors. ω. Figure 4 showcases the generalization performance as a function of the scaling weight ω (see Eq. 7). Higher ω denotes a more non-probabilistic treatment to the point of ignoring the distributional aspects and returning to the auxiliary ProxyAnchor loss [26]. Lower ω indicates a higher emphasis on distributional treatment of proxies (and samples). Across benchmarks and backbones, the best performance is reached with an ω that is neither high nor 0. Thus, the results highlight that our probabilistic proxy-based DML helps the better global realignment of each proxy distribution mode via ProxyAnchor, and vice-versa. Overall, R@1 increases up to 4*pp* at the most suitable scaling choice. This optimum is reached robustly in a large area around the peak (note the logarithmic x-axes).

Comparison Against SOTA. After these strictly standardized comparisons, we now compare the combination of ProxyAnchor and **EL-nivMF**, which performed best in the previous study to the larger DML literature. The hyperparameters and pipeline components (e.g., learning rate, weight decay) differ between the approaches, and so the comparison should be taken with a grain of salt [41,52,57], but we still separate by the backbones and embedding dimensionalities, which are identified as the largest factors of variation [52]. Accounting

Table 2. *Comparison to Literature*, separated by backbones and embedding dimensions. **Bold** denotes best results for a respective Backbone/Dim. subset, **bold** the overall best. Results show that our probabilistically regularized ProxyAnchor method matches or beats previous, in parts notably more complex state-of-the-art methods.

Benchmarks →			CUB200 [64]			CARS196 [30]			SOP [42]		
Methods ↓	Venue	Arch/Dim.	R@1	R@2	NMI	R@1	R@2	NMI	R@1	R@10	NMI
Margin [70]	*ICCV '17*	R50/128	63.6	74.4	69.0	79.6	86.5	69.1	72.7	86.2	**90.7**
Div&Conq [55]	*CVPR '19*	R50/128	65.9	76.6	69.6	**84.6**	**90.7**	**70.3**	75.9	88.4	90.2
MIC [49]	*ICCV '19*	R50/128	66.1	76.8	69.7	82.6	89.1	68.4	77.2	89.4	90.0
PADS [50]	*CVPR '20*	R50/128	**67.3**	**78.0**	69.9	83.5	89.7	68.8	76.5	89.0	89.9
RankMI [23]	*CVPR '20*	R50/128	66.7	77.2	**71.3**	83.3	89.8	69.4	74.3	87.9	90.5
PANC + **EL-niVMF**	-	R50/128	67.0	77.6	70.0	84.0	90.0	69.5	**78.6**	**90.5**	90.1
NormSoft [73]	*BMVC '19*	R50/512	61.3	73.9	–	84.2	90.4	–	78.2	90.6	–
EPSHN [72]	*WACV '20*	R50/512	64.9	75.3	–	82.7	89.3	–	78.3	90.7	–
Circle [61]	*CVPR '20*	R50/512	66.7	77.2	–	83.4	89.7	-	78.3	90.5	–
DiVA [37]	*ECCV '20*	R50/512	69.2	79.3	71.4	**87.6**	**92.9**	72.2	79.6	**91.2**	90.6
DCML-MDW [76]	*CVPR '21*	R50/512	68.4	77.9	71.8	85.2	91.8	**73.9**	**79.8**	90.8	**90.8**
PANC + **EL-niVMF**	–	R50/512	**69.3**	**79.3**	**72.1**	86.2	91.9	70.3	79.4	90.7	90.6
Group [12]	*ECCV '20*	IBN/512	65.5	77.0	69.0	85.6	91.2	**72.7**	75.1	87.5	**90.8**
DR-MS [11]	*TAI '20*	IBN/512	66.1	77.0	–	85.0	90.5	–	–	–	–
ProxyGML [78]	*NeurIPS '20*	IBN/512	66.6	77.6	69.8	85.5	91.8	72.4	78.0	90.6	90.2
DRML [77]	*ICCV '21*	IBN/512	68.7	78.6	69.3	**86.9**	**92.1**	72.1	71.5	85.2	88.1
PANC + MemVir [29]	*ICCV '21*	IBN/512	69.0	79.2	-	86.7	92.0	–	**79.7**	**91.0**	-
PANC + **EL-niVMF**	-	IBN/512	**69.5**	**80.0**	**71.0**	86.4	92.0	71.3	79.2	90.4	90.2

for that, we find competitive performance on all benchmarks (c.f. Table 2), even when compared against other, much more complex state-of-the-art methods relying on multitask learning (DiVA [37], MIC [49]) or reinforcement learning (PADS [50]). This makes our probabilistic take on proxy-based DML a generally attractive approach to DML, with further potential improvements down the line by implementing the probabilistic perspective into these orthogonal extensions.

Computational Overhead. We do note that training with **EL-nivMF** requires the differentiable drawing of samples from vMF-distributions (see Eq. 5 and [8]). This can increase the overall training time, but we found 2–5 samples to already be suitable, limiting the impact on overall walltime to $< 25\%$ against pure ProxyNCA. This is in line with other extensions of ProxyNCA (s.a. [21,37,49,50,55]). The retrieval walltime remains unaffected as cosine-similarity is deployed. As an alternative for rapid training, we provide further probabilistic distribution-to-distribution distances ($d_{\text{EL-vMF}}$, $d_{\text{B-vMF}}$, $d_{\text{KL-vMF}}$) along with analytical solutions (Supp. ??), so that no sampling is required and computational overhead is negligible. We study them in the next section.

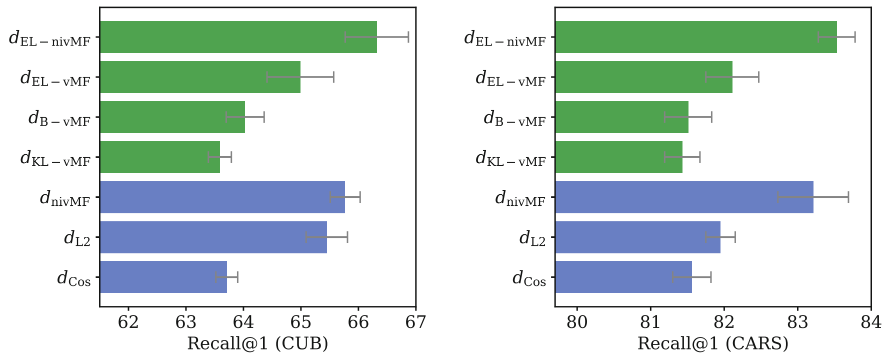


Fig. 5. Distance-to-point (blue) vs. distance-to-distance (green) metrics on CUB and CARS. Bars show average R@1 with standard deviation. (Color figure online)

4.3 Quantitative Comparison of Metrics

Sects. 3.3 and 3.2 provided numerous modeling choices for distributions and distance metrics that can be plugged into the probabilistic DML framework in Eq. 2. This section investigates these possibilities, ultimately motivating the particular choice of $d_{\text{EL-nivMF}}$, and also compares to more traditional distribution-to-point metrics. To ensure fair comparisons, we return to the standardized benchmark protocol of [52] using a 512-dimensional ResNet-50. All hyperparameters are fixed, except for the initial proxy norm and temperature, which are tuned via grid search on a validation set.

Figure 5 shows the R@1 of all three distribution-to-point and four distribution-to-distribution metrics on CUB and CARS. Comparing the distribution-to-point metrics, d_{L2} outperforms d_{Cos} on both datasets, but is dominated by d_{nivMF}. The non-isotropic approach also performs best within the distribution-to-distribution metrics. Within the three isotropic distribution-to-distribution metrics, $d_{\text{KL-vMF}}$ shows the worst performance, with a small gap to the Bhattacharyya and a larger gap to the expected likelihood PPKs. This stands in line with preliminary findings of [7]. The latter performs within one standard deviation of d_{L2}. Altogether, we find that adding non-isotropy to the standard d_{Cos} (i.e., using d_{nivMF}) increases the R@1 by 2.1pp on CUB and 1.7pp on CARS. Further considering the image norm (i.e., $d_{\text{EL-nivMF}}$) adds another 0.6pp on CUB and 0.3pp on CARS.

The enhancement by non-isotropic modeling can be seen as inductive bias towards better resolution of intra-class variances and substructures (see Supp. ??), which drives generalization performance [32,37,52,72,77]. The strong performance of d_{L2} is surprising as many current approaches use a d_{Cos}-based loss [9,26,63]. The crux is that d_{L2} in our setting still uses the cosine distance at retrieval-time, similar to, e.g., [2]. Using d_{L2} also as the retrieval metric would reduce the R@1 by up to $-5.34pp$ across all metrics and datasets, with the highest reduction appearing on the d_{L2}-trained model itself (see Supp. ??). This supports the usage of the norm only during training, discussed in Sect. 3.5, where d_{L2} shares the uncertainty-awareness of distribution-to-distribution metrics, explaining the small gap between d_{L2} and $d_{\text{EL-vMF}}$. Thus, ultimately, we conjecture

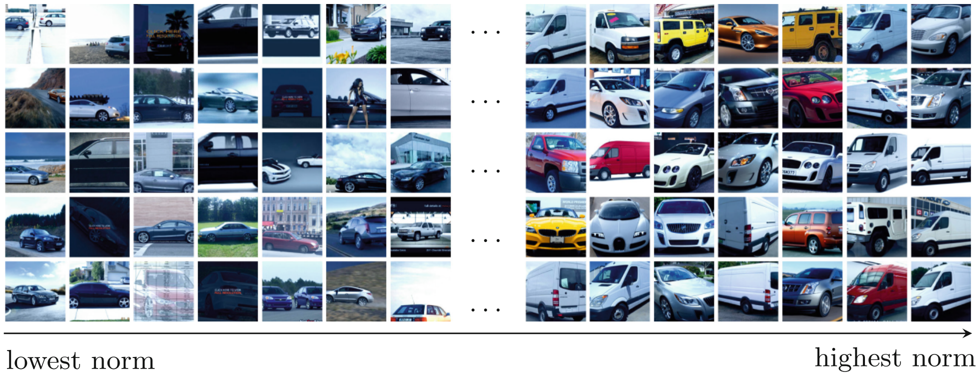


Fig. 6. CARS train images with lowest (left) to highest (right) embedding norms.

that it doesn't matter whether an approach is motivated from a distribution-to-distribution or distribution-to-point perspective, as long as it considers the ambiguity of images (and proxies) during training.

4.4 Embedding Norms Encode Uncertainty

In the previous section, we found that considering the norms of embeddings during training leads to a higher performance. In this section, we qualitatively support that the learned norms actually correspond to a sample-wise ambiguity.

For this, we study the **EL-nivMF** model on CARS. Figure 6 shows the images with the lowest and highest embedding norm in the training set. In many samples with low norm, characteristic parts of the cars are cropped out by the data augmentation (this also happens in the test set, hindering perfect accuracy). Others are overlaid or portray multiple distracting objects. In high-norm images, illumination and camera angle facilitate the detection of class-discriminative features. A competing hypothesis could be that high-norm images comprise mostly car classes with more distinctive designs. However, the differences between low and high norm images also hold within classes, see Supp. ?? and ??. These findings are in line with [31,48,57] and support the hypothesis that the image norm indicates image certainties, motivated by being the sum of visible class-discriminative parts [57]. This justifies the $\kappa_z = \|z\|$ duality underlying the vMF assumption and is consistent with our analysis of uncertainty-aware training in Sect. 3.5.

5 Conclusion

This work proposes non-isotropic probabilistic proxy-based deep metric learning (DML) through uncertainty-aware training and non-isotropic proxy-distributions. Uncertainty-aware training is achieved by treating sample embeddings not as deterministic points but as directional distributions parametrized by embedding directions and, beyond popular DML approaches, norms. This allows semantic ambiguities to be decoupled from the directional semantic context, which mathematically manifests itself in sample-wise temperature scaling and certainty-weighed gradients. Additionally, our non-isotropic von Mises-Fisher distribution for proxies better models intra-class uncertainty, which introduces a low-parameter inductive prior for better generalizing embedding spaces. We support our approach through various ablation studies, which showcase that our proposed framework can operate both as a standalone objective and a probabilistic regularizer on top of existing proxy-based objectives. In both cases, we further found strong performances on the standard DML benchmarks, in parts matching or beating existing state-of-the-art methods. Our findings strongly indicate that a probabilistic treatment of proxy-based DML offers simple, orthogonal enhancements to existing DML methods and enables better generalization.

Limitations. We find that for applications with only few samples per class, the ability to estimate the non-isotropic proxy densities is limited (c.f. performance on SOP). For future work in such sparse settings, returning to the proposed isotropic distribution-to-distribution metrics or introducing across-class priors for the covariance matrices might serve as alternatives.

Acknowledgements. This work has been partially funded by the ERC (853489 - DEXIM) and DFG (2064/1 - Project number 390727645) under Germany's Excellence Strategy. Michael Kirchhof and Karsten Roth thank the International Max Planck Research School for Intelligent Systems (IMPRS-IS) for support. Karsten Roth further acknowledges his membership in the European Laboratory for Learning and Intelligent Systems (ELLIS) PhD program.

References

1. Bouchacourt, D., Tomioka, R., Nowozin, S.: Multi-level variational autoencoder: Learning disentangled representations from grouped observations. In: Thirty-Second AAAI Conference on Artificial Intelligence (AAAI) (2018)
2. Boudiaf, M., et al.: A unifying mutual information view of metric learning: cross-entropy vs. pairwise losses. In: Vedaldi, A., Bischof, H., Brox, T., Frahm, J.-M. (eds.) ECCV 2020. LNCS, vol. 12351, pp. 548–564. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-58539-6_33
3. Brattoli, B., Tighe, J., Zhdanov, F., Perona, P., Chalupka, K.: Rethinking zero-shot video classification: End-to-end training for realistic applications. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2020)
4. Chen, S., Luo, L., Yang, J., Gong, C., Li, J., Huang, H.: Curvilinear distance metric learning. In: Advances in Neural Information Processing Systems 32, pp. 4223–4232. Curran Associates, Inc. (2019). https://papers.nips.cc/paper/8675-curvilinear-distance-metric-learning.pdf
5. Chen, T., Kornblith, S., Norouzi, M., Hinton, G.: A simple framework for contrastive learning of visual representations. In: Proceedings of the 37th International Conference on Machine Learning (ICML) (2020)

6. Chen, W., Chen, X., Zhang, J., Huang, K.: Beyond triplet loss: a deep quadruplet network for person re-identification. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2017)
7. Chun, S., Oh, S.J., De Rezende, R.S., Kalantidis, Y., Larlus, D.: Probabilistic embeddings for cross-modal retrieval. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2021)
8. Davidson, T.R., Falorsi, L., De Cao, N., Kipf, T., Tomczak, J.M.: Hyperspherical variational auto-encoders. In: 34th Conference on Uncertainty in Artificial Intelligence (UAI) (2018)
9. Deng, J., Guo, J., Xue, N., Zafeiriou, S.: ArcFace: Additive angular margin loss for deep face recognition. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2019)
10. Duan, Y., Zheng, W., Lin, X., Lu, J., Zhou, J.: Deep adversarial metric learning. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2018)
11. Dutta, U.K., Harandi, M., Sekhar, C.C.: Unsupervised deep metric learning via orthogonality based probabilistic loss. IEEE Trans.actions Artif. Intell. **1**(1), 74–84 (2020)
12. Elezi, I., Vascon, S., Torcinovich, A., Pelillo, M., Leal-Taixé, L.: The group loss for deep metric learning. In: Vedaldi, A., Bischof, H., Brox, T., Frahm, J.-M. (eds.) ECCV 2020. LNCS, vol. 12352, pp. 277–294. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-58571-6_17
13. Fisher, R.A.: Dispersion on a sphere. Proc. Royal Society London. Series A. Math. Phys. Sci. **217** 295–305 (1953)
14. Goldberger, J., Hinton, G.E., Roweis, S., Salakhutdinov, R.R.: Neighbourhood components analysis. In: Advances in Neural Information Processing Systems (NeurIPS) (2004)
15. Hadsell, R., Chopra, S., LeCun, Y.: Dimensionality reduction by learning an invariant mapping. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (2006)
16. Harwood, B., Kumar, B., Carneiro, G., Reid, I., Drummond, T., et al.: Smart mining for deep metric learning. In: Proceedings of the IEEE International Conference on Computer Vision (ICCV) (2017)
17. Hasnat, M.A., Bohné, J., Milgram, J., Gentric, S., Chen, L.: von Mises-Fisher mixture model-based deep learning: Application to face verification. arXiv preprint arXiv:1706.04264 (2017)
18. He, K., Fan, H., Wu, Y., Xie, S., Girshick, R.: Momentum contrast for unsupervised visual representation learning. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2020)
19. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2016)
20. Hu, J., Lu, J., Tan, Y.: Discriminative deep metric learning for face verification in the wild. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2014)
21. Jacob, P., Picard, D., Histace, A., Klein, E.: Metric learning with horde: High-order regularizer for deep embeddings. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2019)
22. Jebara, T., Kondor, R.: Bhattacharyya and expected likelihood kernels. In: Learning Theory and Kernel Machines (2003)

23. Kemertas, M., Pishdad, L., Derpanis, K.G., Fazly, A.: RankMI: A mutual information maximizing ranking loss. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2020)
24. Kent, J.T.: The Fisher-Bingham distribution on the sphere. J. Royal Stat. Society: Series B (Methodological) **44**(1) 71–80 (1982)
25. Khosla, P., et al.: Supervised contrastive learning. Advances in Neural Information Processing Systems (NeurIPS) (2020)
26. Kim, S., Kim, D., Cho, M., Kwak, S.: Proxy anchor loss for deep metric learning. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2020)
27. Kim, S., Kim, D., Cho, M., Kwak, S.: Embedding transfer with label relaxation for improved metric learning. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2021)
28. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. In: Bengio, Y., LeCun, Y. (eds.) 3rd International Conference on Learning Representations (ICLR) (2015)
29. Ko, B., Gu, G., Kim, H.G.: Learning with memory-based virtual classes for deep metric learning. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV) (2021)
30. Krause, J., Stark, M., Deng, J., Fei-Fei, L.: 3d object representations for fine-grained categorization. In: Proceedings of the IEEE International Conference on Computer Vision Workshops (CVPR) (2013)
31. Li, S., Xu, J., Xu, X., Shen, P., Li, S., Hooi, B.: Spherical confidence learning for face recognition. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2021)
32. Lin, X., Duan, Y., Dong, Q., Lu, J., Zhou, J.: Deep variational metric learning. In: Proceedings of the European Conference on Computer Vision (ECCV) (2018)
33. Liu, W., Wen, Y., Yu, Z., Li, M., Raj, B., Song, L.: Sphereface: Deep hypersphere embedding for face recognition. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2017)
34. Marcel, S., Rodriguez, Y.: Torchvision the machine-vision package of torch. MM '10, Association for Computing Machinery (2010)
35. Mardia, K.V., Jupp, P.E.: Directional statistics (2009)
36. Mardia, K.V.: Statistics of directional data. J. Royal Stat. Society: Series B (Methodological) **37**(3), 349–393 (1975)
37. Milbich, T., et al.: DiVA: diverse visual feature aggregation for deep metric learning. In: Vedaldi, A., Bischof, H., Brox, T., Frahm, J.-M. (eds.) ECCV 2020. LNCS, vol. 12353, pp. 590–607. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-58598-3_35
38. Milbich, T., Roth, K., Brattoli, B., Ommer, B.: Sharing matters for generalization in deep metric learning. IEEE Trans. Pattern Anal. Mach. Intell. **44**(1), 416–427 (2022). https://doi.org/10.1109/TPAMI.2020.3009620
39. Milbich, T., Roth, K., Sinha, S., Schmidt, L., Ghassemi, M., Ommer, B.: Characterizing generalization under out-of-distribution shifts in deep metric learning. In: Ranzato, M., Beygelzimer, A., Dauphin, Y., Liang, P., Vaughan, J.W. (eds.) Advances in Neural Information Processing Systems. vol. 34, pp. 25006–25018. Curran Associates, Inc. (2021), https://proceedings.neurips.cc/paper/2021/file/d1f255a373a3cef72e03aa9d980c7eca-Paper.pdf
40. Movshovitz-Attias, Y., Toshev, A., Leung, T.K., Ioffe, S., Singh, S.: No fuss distance metric learning using proxies. In: Proceedings of the IEEE International Conference on Computer Vision (ICCV) (2017)

41. Musgrave, K., Belongie, S., Lim, S.-N.: A metric learning reality check. In: Vedaldi, A., Bischof, H., Brox, T., Frahm, J.-M. (eds.) ECCV 2020. LNCS, vol. 12370, pp. 681–699. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-58595-2_41
42. Oh Song, H., Xiang, Y., Jegelka, S., Savarese, S.: Deep metric learning via lifted structured feature embedding. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2016)
43. Opitz, M., Waltner, G., Possegger, H., Bischof, H.: Bier-boosting independent embeddings robustly. In: Proceedings of the IEEE International Conference on Computer Vision (ICCV) (2017)
44. Opitz, M., Waltner, G., Possegger, H., Bischof, H.: Deep metric learning with BIER: Boosting independent embeddings robustly. IEEE Trans. Pattern Analysis Mach. Intell. **42**(2), 276–290 (2018)
45. Park, J., Yi, S., Choi, Y., Cho, D.Y., Kim, J.: Discriminative few-shot learning based on directional statistics. arXiv preprint arXiv:1906.01819 (2019)
46. Paszke, A., et al.: Automatic differentiation in pytorch. In: NIPS Workshop on Automatic Differentiation (2017)
47. Qian, Q., Shang, L., Sun, B., Hu, J., Li, H., Jin, R.: Softtriple loss: Deep metric learning without triplet sampling. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV) (2019)
48. Ranjan, R., Castillo, C.D., Chellappa, R.: L2-constrained softmax loss for discriminative face verification. arXiv preprint arXiv:1703.09507 (2017)
49. Roth, K., Brattoli, B., Ommer, B.: Mic: Mining interclass characteristics for improved metric learning. In: Proceedings of the IEEE International Conference on Computer Vision (ICCV) (2019)
50. Roth, K., Milbich, T., Ommer, B.: PADS: Policy-adapted sampling for visual similarity learning. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2020)
51. Roth, K., Milbich, T., Ommer, B., Cohen, J.P., Ghassemi, M.: Simultaneous similarity-based self-distillation for deep metric learning. In: Proceedings of the 38th International Conference on Machine Learning (ICML) (2021)
52. Roth, K., Milbich, T., Sinha, S., Gupta, P., Ommer, B., Cohen, J.P.: Revisiting training strategies and generalization performance in deep metric learning. In: Proceedings of the 37th International Conference on Machine Learning (ICML) (2020)
53. Roth, K., Vinyals, O., Akata, Z.: Integrating language guidance into vision-based deep metric learning. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 16177–16189 (June 2022)
54. Roth, K., Vinyals, O., Akata, Z.: Non-isotropy regularization for proxy-based deep metric learning. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 7420–7430 (2022)
55. Sanakoyeu, A., Tschernezki, V., Buchler, U., Ommer, B.: Divide and conquer the embedding space for metric learning. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2019)
56. Schroff, F., Kalenichenko, D., Philbin, J.: Facenet: A unified embedding for face recognition and clustering. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2015)
57. Scott, T.R., Gallagher, A.C., Mozer, M.C.: von Mises-Fisher loss: An exploration of embedding geometries for supervised learning. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV) (2021)
58. Shi, Y., Jain, A.K.: Probabilistic face embeddings. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV) (2019)

59. Sinha, S., et al.: Uniform priors for data-efficient learning. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops, pp. 4017–4028 (2022)
60. Sohn, K.: Improved deep metric learning with multi-class n-pair loss objective. In: Advances in Neural Information Processing Systems (NeurIPS) (2016)
61. l. Sun, Y., et al.: Circle loss: A unified perspective of pair similarity optimization. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2020)
62. Szegedy, C., et al.: Going deeper with convolutions. In: Computer Vision and Pattern Recognition (CVPR) (2015)
63. Teh, E.W., DeVries, T., Taylor, G.W.: ProxyNCA++: Revisiting and revitalizing proxy neighborhood component analysis. In: Proceedings of the European Conference on Computer Vision (ECCV) (2020)
64. Wah, C., Branson, S., Welinder, P., Perona, P., Belongie, S.: The Caltech-UCSD birds-200-2011 dataset. Tech. Rep. CNS-TR-2011-001, California Institute of Technology (2011)
65. Wang, J., Zhou, F., Wen, S., Liu, X., Lin, Y.: Deep metric learning with angular loss. In: Proceedings of the IEEE International Conference on Computer Vision (ICCV) (2017)
66. Wang, T., Isola, P.: Understanding contrastive representation learning through alignment and uniformity on the hypersphere. In: Proceedings of the 37th International Conference on Machine Learning (ICML) (2020)
67. Wang, X., Han, X., Huang, W., Dong, D., Scott, M.R.: Multi-similarity loss with general pair weighting for deep metric learning. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2019)
68. Weisstein, E.W.: Hypersphere (2002)
69. Wightman, R.: Pytorch image models. https://github.com/rwightman/pytorch-image-models (2019)
70. Wu, C.Y., Manmatha, R., Smola, A.J., Krahenbuhl, P.: Sampling matters in deep embedding learning. In: Proceedings of the IEEE International Conference on Computer Vision (ICCV) (2017)
71. Xuan, H., Stylianou, A., Pless, R.: Improved embeddings with easy positive triplet mining. In: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV) (March 2020)
72. Xuan, H., Stylianou, A., Pless, R.: Improved embeddings with easy positive triplet mining. In: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV) (March 2020)
73. Zhai, A., Wu, H.: Making classification competitive for deep metric learning. arXiv Preprint arXiv:1811.12649 (2018)
74. Zhe, X., Chen, S., Yan, H.: Directional statistics-based deep metric learning for image classification and retrieval. Pattern Recognition 93 (2018)
75. Zheng, W., Chen, Z., Lu, J., Zhou, J.: Hardness-aware deep metric learning. The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2019)
76. Zheng, W., Wang, C., Lu, J., Zhou, J.: Deep compositional metric learning. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2021)
77. Zheng, W., Zhang, B., Lu, J., Zhou, J.: Deep relational metric learning. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV) (2021)

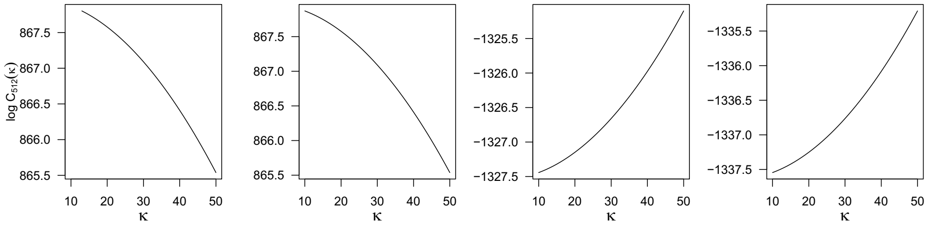
78. Zhu, Y., Yang, M., Deng, C., Liu, W.: Fewer is more: A deep graph metric learning perspective using fewer proxies. In: Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M.F., Lin, H. (eds.) Advances in Neural Information Processing Systems (NeurIPS) (2020)
79. Zimmermann, R.S., Sharma, Y., Schneider, S., Bethge, M., Brendel, W.: Contrastive learning inverts the data generating process. In: Proceedings of the 38th International Conference on Machine Learning (ICML) (2021)

Supplementary Material:
A Non-isotropic Probabilistic Take on Proxy-based Deep Metric Learning

Michael Kirchhof[1,*], Karsten Roth[1,*], Zeynep Akata[1], and Enkelejda Kasneci[1]

[1]University of Tübingen, Germany. (*) equal contribution

A Approximation of the von Mises-Fisher Distribution's Normalizing Constant



(a) Exact values (b) Our approx. (c) Approx. of [3] (d) Approx. of [9]

Fig. 8: Comparison of approximations and exact values of the logarithmized normalization constant of the vMF distribution $\log C_M(\kappa)$ for $M = 512$ dimensions.

As we aim to resolve sample-specific ambiguities captured by κ_z, we need to calculate the logarithmic normalizing constant of the vMF distribution:

$$\log C_M(\kappa) = \log \frac{\kappa^{M/2-1}}{(2\pi)^{M/2} I_{M/2-1}(\kappa)}, \tag{11}$$

where I_d is the modified Bessel function of first kind at order d and M is the dimensionality of the embedding space. However, I_d is expensive to compute and impossible to backpropagate through in high dimensions since it has no closed form. Hence, it is commonly approximated in the literature. [3] and [9] for example utilize approximations from lower and upper bounds which are shown in Figure 8c and 8d for $M = 512$. However, if we calculate $\log C_M$ from the exact Bessel functions implemented in `R 4.1.1`'s base package [7], we see in Figure 8a that $\log C_M$ is monotonically decreasing, because I_d is monotonically increasing with κ [5, Section 10.37].

To account for this issue, we thus choose to derive an approximation by directly fitting a quadratic model to the exact Bessel function for $M \in \{128, 512\}$ with $\kappa \in$

$\{10, \ldots, 50\}$. The resulting approximations are

$$\log C_{128}(\kappa) \approx 127 - 0.01909 \cdot \kappa - 0.003355 \cdot \kappa^2 \text{ and} \tag{12}$$

$$\log C_{512}(\kappa) \approx 868 - 0.0002662 \cdot \kappa - 0.0009685 \cdot \kappa^2. \tag{13}$$

The mean squared error of these approximations to the ground truth values is smaller than 0.1%, which is visually confirmed in Figure 8b. During experimentation, we found that the model is insensitive to perturbations in the precise coefficients. Also, we found that a linear model is too simple and an exponential model imposed very high gradients and inverts the behaviour of the metrics when κ is high. Hence, we decided for the quadratic approximation as the simplest yet well extrapolating function. As a reference for future work, we note that [2] recently gave an additional approximation implemented in PyTorch.

B Derivation of the Non-isotropic von Mises-Fisher Distribution

The nivMF can be motivated by a transformed vMF distribution, which we assume to be parametrized by $\mu \in \mathcal{S}^{M-1}$ and $K = \text{diag}(\kappa) \in \mathbb{R}_{>0}^{(M \times M)}$, $\kappa \in \mathbb{R}_{>0}^{M}$. Transforming our parameters into $\tilde{\mu} = \frac{K\mu}{\|K\mu\|}$ and $\tilde{\kappa} = \|K\mu\|$, we can define an ordinary vMF distribution $\tilde{X} \sim \text{vMF}(\tilde{\mu}, \tilde{\kappa})$ with density

$$f_{\tilde{X}}(\tilde{x}) = C_M(\tilde{\kappa}) \exp\left(\tilde{\kappa}\tilde{x}^\top \tilde{\mu}\right) \ . \tag{14}$$

For ease of notation, we do not include the subscript p to denote specific proxies. Now, we substitute $\tilde{x} := g(x) = \frac{Kx}{\|Kx\|}$. Note that g is bijective as a function $g : \mathcal{S}^{M-1} \rightarrow \mathcal{S}^{M-1}$, but non-bijective when seen as a function $g : \mathbb{R}^M \rightarrow \mathbb{R}^M$, since it would lose a degree of freedom due to normalization. We will still treat it as the latter and ignore the non-bijectivity, such that the following should be seen as motivation and not proof, and comment on the implications further below. We now seek the density of $X = g^{-1}(\tilde{X})$. The change-of-variable theorem gives

$$f_X(x) = f_{\tilde{X}}(\tilde{x}) |\det \frac{\partial g(x)}{\partial x}| \, . \tag{15}$$

By Equation 130 given in [6] and the chain rule, we obtain

$$\frac{\partial g(x)}{\partial x} = \left(\frac{1}{\|Kx\|} I_m - \frac{K^\top x x^\top K}{\|Kx\|^3} \right) K^\top \tag{16}$$

$$= \left(\frac{1}{\tilde{\kappa}} I_M - \frac{(\tilde{\kappa}\tilde{\mu})(\tilde{\kappa}\tilde{\mu})^\top}{\tilde{\kappa}^3} \right) K^\top \tag{17}$$

$$= \frac{1}{\tilde{\kappa}} \left(I_M - \tilde{\mu}\tilde{\mu}^\top \right) K^\top \ . \tag{18}$$

Since the first part of this matrix is a projection on the orthogonal complement of $\tilde{\mu}$, the matrix has rank $M - 1$ and the determinant becomes zero. This is a consequence of the broken bijectivity assumption from above. However, we can see that Equation 18 essentially projects K on the tangential plane of $\tilde{\mu}$. By taking its determinant, we measure the volume of the remaining $(M-1)$-dimensional concentration sphere. Performing a singular value decomposition on Equation 18 reveals that μ is the eigenvector with eigenvalue 0. So, if we substract the contribution of μ to the volume of K, which is $\|K\mu\| = \tilde{\kappa}$, we obtain

$$D(K) = \frac{\prod_{m=1}^{M} \kappa_m}{\tilde{\kappa}} \,. \tag{19}$$

When we plug this heuristic into Equation 15, we arrive at the nivMF density:

$$\begin{aligned} f_X(x) &= C_M(\tilde{\kappa}) \exp\left(\tilde{\kappa}\tilde{x}^\top \tilde{\mu}\right) D(K) & (20)\\ &= C_M(\|K\mu\|)\, D(K) \exp\left(\|K\mu\| \left(\frac{Kx}{\|Kx\|}\right)^\top \frac{K\mu}{\|K\mu\|}\right) & (21)\\ &= C_M(\|K\mu\|)\, D(K) \exp\left(\|K\mu\|\, s(Kx, K\mu)\right) \,. & (22) \end{aligned}$$

We stress that $D(K)$ is a heuristic choice, such that the proposed nivMF density strictly speaking yields only a measure and not necessarily a probability measure. An analytical solution is promising material for future work. It may also enable the density of the nivMF to become a true expansion of the vMF density, i.e., $D(K)$ may vanish when $K = \kappa I_M$ for $\kappa > 0$, which is currently not the case. In empirical tests, dropping $D(K)$ lead to a considerably severed performance.

C Further distribution-to-distribution Metrics

We can define further distribution-to-distribution metrics beyond $d_{\text{EL-nivMF}}$. One starting point are probability product kernels (PPK) [1]. They are a family of metrics to compare two distributions ρ and ζ by the product of their densities:

$$\text{PPK}_\gamma(\rho, \zeta) = \int_{\mathcal{E}} \rho(a)^\gamma \zeta(a)^\gamma da, \text{ with } \gamma > 0. \tag{23}$$

Since the loss in Equation 2 takes the exponential of the distance metrics, we take their logarithms here to retain the PPK as actual score in nominator and denominator. In particular, if we assume a vMF distribution for both ρ and ζ

$$d_{\text{B-vMF}}(\rho, \zeta) := -\log(\text{PPK}_{0.5}(\rho, \zeta)) \tag{24}$$

gives the Bhattacharyya distance and

$$d_{\text{EL-vMF}}(\rho, \zeta) := -\log(\text{PPK}_{1}(\rho, \zeta)) \tag{25}$$

gives the expected likelihood distance, also known as mutual likelihood score [10]. Their analytical solutions are provided in Supp. D.

The previous metrics are symmetric in ρ and ζ. To capture the inherent asymmetry between samples and proxies, we also study the Kullback-Leibler divergence $d_{\text{KL-vMF}}(\rho, \zeta) := \text{KL}(\zeta || \rho)$. Its analytical solution if both ρ and ζ are vMF densities is given in Supp. E.

D Analytical Solutions of Bhattacharyya and Expected Likelihood Distance

Let ζ and ρ be densities of two vMF-distributed random variables with parameters $\nu_z = \kappa_z \mu_z$ and $\nu_p = \kappa_p \mu_p$, respectively.

Bhattacharyya distance. Since the vMF is a member of the exponential family, [1] gives us that

$$\text{PPK}_{0.5}(\rho, \zeta) = \exp(K(\nu_z/2 + \nu_p/2) - K(\nu_z)/2 - K(\nu_p)/2), \text{ with} \tag{26}$$

$$K(\nu) = -\log C_M(\|\nu\|)\,. \tag{27}$$

Thus,

$$d_{\text{B-vMF}}(\rho, \zeta) = -\text{log}(\text{PPK}_{0.5}(\rho, \zeta)) \tag{28}$$

$$= \log C_M(\|\nu_z + \nu_p\|/2) - \log C_M(\nu_z)/2 - \log C_M(\nu_p)/2\,. \tag{29}$$

Expected likelihood distance. We can extend

$$\text{PPK}_1(\rho, \zeta) = \int_{\mathcal{E}} \zeta(\tilde{z})\rho(\tilde{z}) d\tilde{z} \tag{30}$$

$$= C_M(\kappa_z) \cdot C_M(\kappa_p) \int_{\mathcal{E}} \exp((\kappa_z \mu_z + \kappa_p \mu_p)^\top \tilde{z}) d\tilde{z} \tag{31}$$

$$= \frac{C_M(\kappa_z) \cdot C_M(\kappa_p)}{C_M(\|\nu_0\|)} \int_{\mathcal{E}} C_M(\|\nu_0\|) \exp(\nu_0^\top \tilde{z}) d\tilde{z}, \text{ with} \tag{32}$$

$$\nu_0 := \kappa_z \mu_z + \kappa_p \mu_p, \tag{33}$$

such that the latter is again the density of a vMF distributed random variable, whose integral over the embedding space is 1. Then,

$$d_{\text{EL-vMF}}(\rho, \zeta) = -\text{log}(\text{PPK}_1(\rho, \zeta)) \tag{34}$$

$$= \log C_M(\|\nu_z + \nu_p\|) - \log C_M(\nu_z) - \log C_M(\nu_p)\,. \tag{35}$$

Note that both $d_{\text{EL-vMF}}$ and $d_{\text{B-vMF}}$ depend on $\|\nu_z + \nu_p\|$ which implicitely respects the cosine similarity between μ_z and μ_p, but also processes κ_z and κ_p.

E Analytical Solution of KL-Divergence

Let ζ and ρ be densities of two vMF-distributed random variables with parameters μ_z, κ_z and μ_p, κ_p, respectively. Then

$$KL(\zeta || \rho) = \int_{\mathcal{E}} \zeta(\tilde{z}) \log \frac{\zeta(\tilde{z})}{\rho(\tilde{z})} d\tilde{z} \tag{36}$$

$$= \int_{\mathcal{E}} \log C_M(\kappa_z) - \log C_M(\kappa_p) + (\kappa_z \mu_z^\top - \kappa_p \mu_p^\top) \tilde{z} d\zeta(\tilde{z}) \tag{37}$$

$$= \log C_M(\kappa_z) - \log C_M(\kappa_p) + (\kappa_z \mu_z^\top - \kappa_p \mu_p^\top) \int_{\mathcal{E}} \tilde{z} d\zeta(\tilde{z}) \tag{38}$$

$$= \log C_M(\kappa_z) - \log C_M(\kappa_p) + (\kappa_z \mu_z^\top - \kappa_p \mu_p^\top) \mu_z \tag{39}$$

F Gradients of d_{L2} and d_{Cos}

We are interested in differentiating the loss $\mathcal{L}_{\text{NCA++}}$ from Equation 1 in §3.2 by the cosine similarity between the image z and a proxy of interest p. Let p^* denote the ground-truth proxy of z and $\frac{\delta}{\delta s} := \frac{\delta}{\delta s(\mu_p, \mu_z)}$. Then,

$$\frac{\delta}{\delta s}\mathcal{L}_{\text{NCA++}} = \begin{cases} \frac{\delta}{\delta s} d(\rho^*, \zeta)/t + \frac{\delta}{\delta s}\log(\sum_{c=1}^{C}\exp(-d(\rho_c, \zeta)/t)) & \text{, if } p = p^* \\ \frac{\delta}{\delta s}\log(\sum_{c=1}^{C}\exp(-d(\rho_c, \zeta)/t)) & \text{, else} \end{cases} \tag{40}$$

and by the chain rule we get

$$\frac{\delta}{\delta s}\log\left(\sum_{c=1}^{C}\exp(-d(\rho_c, \zeta)/t)\right) = -\frac{\exp(-d(\rho, \zeta)/t)}{\sum_{c=1}^{C}\exp(-d(\rho_c, \zeta)/t)}\frac{\delta}{\delta s}d(\rho_c, \zeta)/t\ . \tag{41}$$

Let's consider the $\mathcal{L}_{\text{NCA++}}^{\text{Cos}}$ loss, i.e., $d(\rho, \zeta) = -s(\mu_p, \mu_z)$. We can plug $\frac{\delta}{\delta s}d(\rho, \zeta) = -1$ into Equations 40 and 41 and obtain:

$$\frac{\delta}{\delta s}\mathcal{L}_{\text{NCA++}}^{\text{Cos}} = \begin{cases} \frac{1}{t}\left(-1 + \frac{\exp(-d(\rho,\zeta)/t)}{\sum_{c=1}^{C}\exp(-d(\rho_c,\zeta)/t)}\right) & \text{, if } p = p^* \\ \frac{1}{t}\frac{\exp(-d(\rho,\zeta)/t)}{\sum_{c=1}^{C}\exp(-d(\rho_c,\zeta)/t)} & \text{, else} \end{cases} \tag{42}$$

$$= \begin{cases} \frac{1}{t}\left(-1 + \frac{\exp(s(\mu_p,\mu_z)/t)}{\sum_{c=1}^{C}\exp(s(\mu_{p_c},\mu_z)/t)}\right) & \text{, if } p = p^* \\ \frac{1}{t}\frac{\exp(s(\mu_p,\mu_z)/t)}{\sum_{c=1}^{C}\exp(s(\mu_{p_c},\mu_z)/t)} & \text{, else} \end{cases}\ . \tag{43}$$

Now, consider $\mathcal{L}_{\text{NCA++}}^{\text{L2}}$, i.e., $d(\rho, \zeta) = \|\nu_p - \nu_z\|^2 = \kappa_p^2 + \kappa_z^2 - 2\kappa_p\kappa_z s(\mu_p, \mu_z)$, following from the law of cosines. Here, $\frac{\delta}{\delta s}d(\nu_p, \nu_z) = -2\kappa_p\kappa_z$, which we can again plug into Equations 40 and 41 and obtain:

$$\frac{\delta}{\delta s}\mathcal{L}_{\text{NCA++}}^{\text{L2}} = \begin{cases} -\frac{2\kappa_p\kappa_z}{t} + \frac{2\kappa_p\kappa_z}{t}\frac{\exp(-d(\rho,\zeta)/t)}{\sum_{c=1}^{C}\exp(-d(\rho_c,\zeta)/t)} & \text{, if } p = p^* \\ \frac{2\kappa_p\kappa_z}{t}\frac{\exp(-d(\rho,\zeta)/t)}{\sum_{c=1}^{C}\exp(-d(\rho_c,\zeta)/t)} & \text{, else} \end{cases} \tag{44}$$

$$= \begin{cases} -\frac{2\kappa_p\kappa_z}{t} + \frac{2\kappa_p\kappa_z}{t}\frac{\exp((\kappa_p^2+2\kappa_p\kappa_z s(\mu_p,\mu_z))/t)}{\sum_{c=1}^{C}\exp((\kappa_{p_c}^2+2\kappa_p\kappa_z s(\mu_{p_c},\mu_z))/t)} & \text{, if } p = p^* \\ \frac{2\kappa_p\kappa_z}{t}\frac{\exp((\kappa_p^2+2\kappa_p\kappa_z s(\mu_p,\mu_z))/t)}{\sum_{c=1}^{C}\exp((\kappa_{p_c}^2+2\kappa_{p_c}\kappa_z s(\mu_{p_c},\mu_z))/t)} & \text{, else} \end{cases}\ . \tag{45}$$

G Summary of Loss Calculation

Algorithm 1 sketches how **EL-nivMF** is implemented practically. As discussed, the parameters of the proxies are learnable parameters, whereas the vMF distributions of points are predicted by an encoder. Thus, the module in Algorithm 1 can be plugged on-top of an encoder and trained jointly. Since test-time retrieval only requires access to the image-embeddings, the module can be discarded after training.

Algorithm 1: Module to compute **EL-nivMF** loss

```
Function initialize($C$: num proxies, $M$: dimensions, $N$: num samples):
    $\mu_\rho \leftarrow$ learnable tensor $\in [C, M]$
    $\kappa_\rho \leftarrow$ learnable tensor $\in [C, M]$
    $t \leftarrow$ learnable parameter $\in [1]$
    Save $C, M, N$
Function loss($z$: image embedding $\in [1, M]$, $c^*$: ground-truth proxy index):
    samples $\leftarrow$ empty matrix $\in [N, D]$
    for $n = 1, \ldots, N$ do
        samples$[n, :] \sim$ vMF$\left(\mu = \frac{z}{\|z\|}, \kappa = \|z\|\right)$
    end
    sim_to_proxy $\leftarrow$ empty vector $\in [C]$
    for $c = 1, \ldots, C$ do
        logls $\leftarrow$ empty vector $\in [N]$
        for $n = 1, \ldots, N$ do
            logls$[n]$
              $=$ log(nivmf_likelihood$(z, \mu = \mu_\rho[c, :], K =$ diag$(\kappa_\rho[c, :])$))
        end
        sim_to_proxy$[c] \leftarrow$ logsumexp(logls$/t$)
    end
    logloss $\leftarrow -$sim_to_proxy$[c^*]$ + logsumexp(sim_to_proxy)
    return logloss
```

H Experimental Details

As already noted in §3.3, we generally utilize $N \approx 10$ for our Monte-Carlo estimation of the PPK kernel (Eq. 5), but switch to $N = 5$ for hyperparameter searches and $N = 20$ for our ablation experiments, as within this range, we found performance to be similar.

I Experimental Details Ablation Study

To reduce any influences of covariates, we seek to keep experimental settings in the ablation study in §4.3 constant across all benchmarked metrics. Hence, we fixed all hyperparameters as in the previous experiment, and tuned the following hyperparameters for each approach on validation data:

$$t \in \{1, 1/32, 1/256\} \tag{46}$$

$$\kappa_p \in \{10, 50, 200\} \text{ (for \textbf{ni-vMF}, this is for each dimension)} . \tag{47}$$

Across all metrics, we used the dimensionality $M = 512$, a batchsize of 106, and 150 epochs on CARS and 50 on CUB. To reduce the initialization noise, we initiated each hyperparameter-tuning experiment 3 times with random seeds, then calculated the median of the maximum $R@1$ performance on the validation set, and ran the best hyperparameter settings with 5 seeds.

J L_2 Distance as Retrieval Metric

Table 4: R@1 of the same trained models from Figure 5, but using the euclidean instead of the cosine distance for retrieval.

Method	CUB	CARS
d_{L2}	61.89 ± 0.36	76.61 ± 0.17
d_{Cos}	62.01 ± 0.35	76.94 ± 0.49
d_{nivMF}	63.74 ± 0.18	78.62 ± 0.41
$d_{\text{B-vMF}}$	62.29 ± 0.34	79.69 ± 0.15
$d_{\text{EL-vMF}}$	62.49 ± 0.56	80.17 ± 0.24
$d_{\text{KL-vMF}}$	61.68 ± 0.36	76.65 ± 0.20
$d_{\text{EL-nivMF}}$	63.69 ± 0.56	76.37 ± 5.32

K Qualitative Impact on Image Norms

To understand in more detail the difference in learned and assigned image norms produced when training with $d_{\text{EL-nivMF}}$, we compare the distribution of image norms between those belonging to originally correctly and incorrectly classified samples (initial separation done using a standard baseline DML model operating on d_{cos}) for CUB & CARS, respectively. Results are shown in Fig. 9, which reveal that correct classifications on average have higher norms while miss-classifications are more often attributed to lower norms. This aligns well with the underlying motivation assigning low norms to ambiguous images (compare to e.g. Sec.4.4).

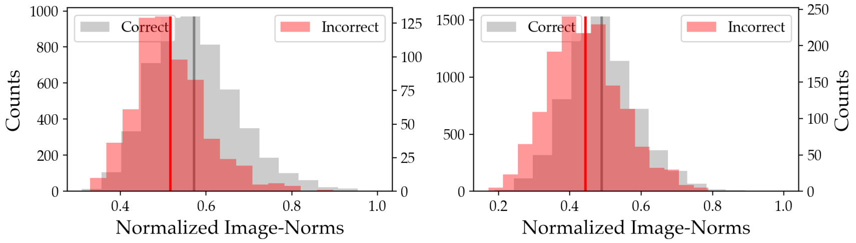


Fig. 9: Norms of prev. correct/incorrect pred. on CUB/CARS.

L Non-isotropic Proxies Encourage Diverse Representations

Finally, we qualitatively investigate the metric representation spanned by metric learners trained using $d_{\text{EL-nivMF}}$. To do so, we follow both [8] and look at the feature diversity, as well as evaluating the cluster diversity to see whether encouraging unique class-proxy distributions helps in learning a more diverse class-specific encoding. For the former, we follow [8] and evaluate the uniformity of the sorted spectral value distribution of all training image embeddings to measure the number of significant directions of variances in feature space. The latter is simply computed as the variance (i.e. diversity) of intraclass distances for each class-cluster. For both cases, we specifically care about relative changes compared to models trained without probabilistic treatment (i.e. using d_{cos}) as well as changes going from an isotropic ($d_{\text{EL-vmf}}$) to a non-isotropic setup ($d_{\text{EL-nivMF}}$). Results are summarized in Tab. 5, showcasing a consistent improvement in both feature and cluster diversity when incorporating both a probabilistic treatment and a non-isotropic encoding of proxy distributions. This provides further heuristic evidence linking the usage of $d_{\text{EL-nivMF}}$ to a better capture of the semantic class variability as well as an improved incorporation of a more diverse feature set, shown to facilitate generalisation [8,4].

Dataset	Metric	$d_{\text{cos}} \rightarrow d_{\text{EL-vMF}}$	$d_{\text{cos}} \rightarrow d_{\text{EL-nivMF}}$
CARS	Cluster-Div.↑	$+24\%$	$+31\%$
	Feat.-Div. ↑	$+13\%$	$+14\%$
CUB	Cluster-Div. ↑	$+11\%$	$+25\%$
	Feat.-Div. ↑	$+6\%$	$+8\%$

Table 5: Metrics on how **EL-nivMF** structures the embeddings.

M Further Qualitative Embedding Norm Studies

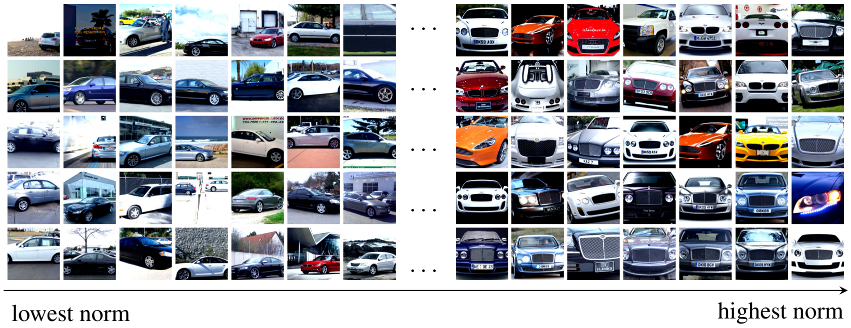


Fig. 10: CARS train images with lowest (left) to highest (right) embedding norms on a $M = 512$ dimensional ResNet-50 backend.

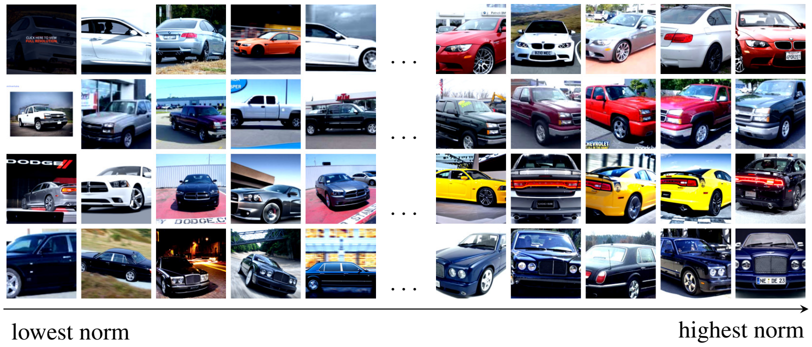


Fig. 11: Images for four randomly chosen classes (rows) of the CARS training set, ordered by their norm from lowest (left) to highest (right). Obtained from the $d_{\text{EL-vMF}}$ model on a ResNet-50, where the norms of image embeddings range from 70.58 to 140.09 whereas the proxy norms are between 45.95 to 79.98.

References

1. Jebara, T., Kondor, R.: Bhattacharyya and expected likelihood kernels. In: Learning Theory and Kernel Machines (2003) 3, 4
2. Kim, M.: On PyTorch implementation of density estimators for von Mises-Fisher and its mixture. arXiv preprint arXiv:2102.05340 (2021) 2
3. Kumar, S., Tsvetkov, Y.: Von Mises-Fisher loss for training sequence to sequence models with continuous outputs. In: International Conference on Learning Representations (ICLR) (2019) 1
4. Milbich, T., Roth, K., Bharadhwaj, H., Sinha, S., Bengio, Y., Ommer, B., Cohen, J.P.: Diva: Diverse visual feature aggregation for deep metric learning. In: Vedaldi, A., Bischof, H., Brox, T., Frahm, J.M. (eds.) Proceedings of the European Conference on Computer Vision (ECCV) (2020) 8
5. Olver, F.W.J., Daalhuis, A.B.O., Lozier, D.W., Schneider, B.I., Boisvert, R.F., Clark, C.W., Miller, B.R., Saunders, B.V., Cohl, H.S., M. A. McClain, e.: NIST Digital library of mathematical functions, `https://dlmf.nist.gov/10.37` 1
6. Petersen, K.B., Pedersen, M.S., et al.: The matrix cookbook. Technical University of Denmark **7**(15) (2008) 2
7. R Core Team: R: A language and environment for statistical computing. R Foundation for Statistical Computing, Vienna, Austria (2021) 1
8. Roth, K., Milbich, T., Sinha, S., Gupta, P., Ommer, B., Cohen, J.P.: Revisiting training strategies and generalization performance in deep metric learning. In: Proceedings of the 37th International Conference on Machine Learning (ICML) (2020) 8
9. Scott, T.R., Gallagher, A.C., Mozer, M.C.: von Mises-Fisher loss: An exploration of embedding geometries for supervised learning. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV) (2021) 1
10. Shi, Y., Jain, A.K.: Probabilistic face embeddings. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV) (2019) 3

B

Probabilistic contrastive learning recovers the correct aleatoric uncertainty of ambiguous inputs

This appendix contains the full paper and appendix discussed in Chapter 3, reproduced with permission.

Michael Kirchhof, Enkelejda Kasneci, and Seong Joon Oh. Probabilistic contrastive learning recovers the correct aleatoric uncertainty of ambiguous inputs, 2023. First published in: *Proceedings of the 40th International Conference on Machine Learning*, in *Proceedings of Machine Learning Research* 202:17085-17104. Available from `https://proceedings.mlr.press/v202/kirchhof23a.html`. Reproduced under a Creative Commons Attribution 4.0 International License (`http://creativecommons.org/licenses/by/4.0/legalcode`).

Probabilistic Contrastive Learning Recovers the Correct Aleatoric Uncertainty of Ambiguous Inputs

Michael Kirchhof[1] **Enkelejda Kasneci**[2] **Seong Joon Oh**[3]

Abstract

Contrastively trained encoders have recently been proven to invert the data-generating process: they encode each input, e.g., an image, into the true latent vector that generated the image (Zimmermann et al., 2021). However, real-world observations often have inherent ambiguities. For instance, images may be blurred or only show a 2D view of a 3D object, so multiple latents could have generated them. This makes the true posterior for the latent vector probabilistic with heteroscedastic uncertainty. In this setup, we extend the common InfoNCE objective and encoders to predict latent distributions instead of points. We prove that these distributions recover the correct posteriors of the data-generating process, including its level of aleatoric uncertainty, up to a rotation of the latent space. In addition to providing calibrated uncertainty estimates, these posteriors allow the computation of credible intervals in image retrieval. They comprise images with the same latent as a given query, subject to its uncertainty. Code is at `https://github.com/mkirchhof/Probabilistic_Contrastive_Learning`.

1. Introduction

Contrastive learning (Chen et al., 2020) trains encoders to output embeddings that are close to one another for semantically similar inputs and far apart for unsimilar inputs. This general notion of similarity allows transferring pretrained encoders to downstream tasks (Wang et al., 2022; Ardeshir & Azizan, 2022; Islam et al., 2021; Khosla et al., 2020).

Recently, Zimmermann et al. (2021) corroborated this intuition by a theoretical result: under weak assumptions, the embeddings learned under an InfoNCE (Oord et al., 2018) loss are exactly equal to the true latent vectors, up to a rotation of the spherical latent space. This comes from a nonlinear Independent Component Analysis (ICA) perspective (Comon & Jutten, 2010). It assumes an unknown nonlinear generative process that transforms true latents into our observations. Contrastively trained encoders *invert* this nonlinear function and recover the original latent space.

This holds for the class of generative processes that are deterministic and injective, so that each image could have been generated by only one latent vector. This is often violated in practice. In Figure 1, the lower image of an animal is in low-resolution, so it is impossible to tell which exact species, i.e., which latent variables, underlie the image. In fact, most scenarios in the wild involve some form of such aleatoric uncertainty, including 3D-to-2D projections (Chen et al., 2021), partially covered objects (Kraus & Dietmayer, 2019), or images with a low resolution or bad crop (Li et al., 2021). It also manifests itself outside the image domain, such as in the inherent ambiguity of natural language (Chun et al., 2022) or measurement noise in general (Meech & Stanley-Marbell, 2021). Quantifying such uncertainties is a key goal of the recent reliable machine learning efforts (Tran et al., 2022; Galil et al., 2023). This has use cases in safety-critical downstream applications like medical imaging (Barbano et al., 2022). If an image is too ambiguous, a model can reject it or defer the prediction to a human. Another application is active learning, where we want to choose samples with high uncertainty (Lewis & Catlett, 1994).

This work generalizes the previous theoretical result to this more challenging setting. We do not assume that generative process is an injective and deterministic function, but allow it to be a conditional distribution. We propose Monte-Carlo InfoNCE (MCInfoNCE), a probabilistic analog of InfoNCE. It trains encoders to predict distributions over the possible latents, called probabilistic embeddings (Oh et al., 2019; Shi & Jain, 2019). We prove that MCInfoNCE attains its global minimum when the encoder recovers *the true posteriors* of the generative process, up to a rotation of the latent space; both in terms of both the mean (which latent is most likely to have generated the image) and the variance (the level of

[1]University of Tübingen, Germany [2]TUM University, Munich, Germany [3]University of Tübingen, Tübingen AI Center, Germany. Correspondence to: Michael Kirchhof <michael dot kirchhof at uni dash tuebingen dot de>.

Proceedings of the 40^{th} International Conference on Machine Learning, Honolulu, Hawaii, USA. PMLR 202, 2023. Copyright 2023 by the author(s).

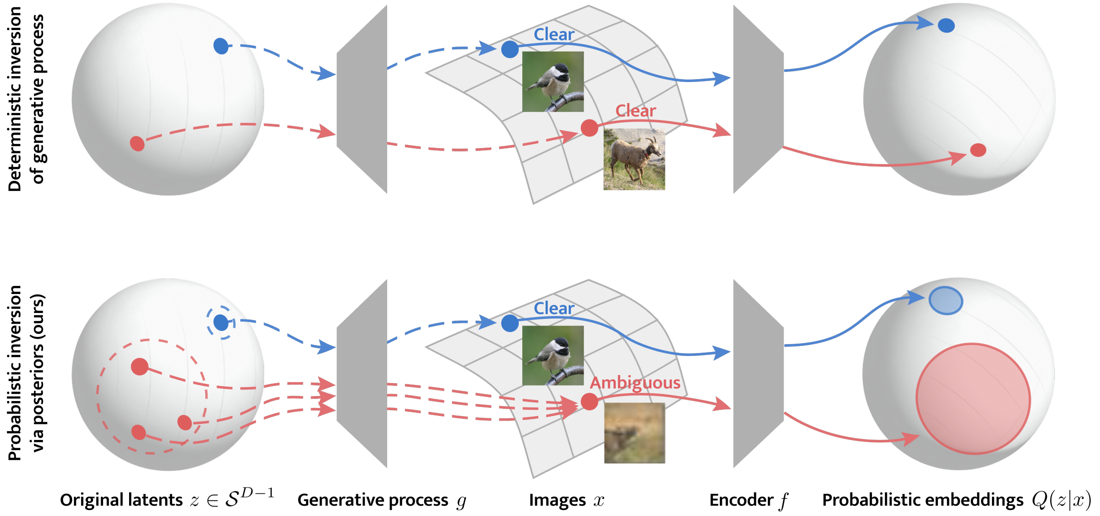

Deterministic inversion of generative process
Clear
Clear
Probabilistic inversion via posteriors (ours)
Clear
Ambiguous
Original latents $z \in \mathcal{S}^{D-1}$
Generative process g
Images x
Encoder f
Probabilistic embeddings $Q(z|x)$

Figure 1. Deteministic encoders embed images to points in the latent space. This recovers the latent vectors that generated them (dashed), up to a rotation (top). However, if an image is ambiguous there are multiple possible latents that could have generated it (bottom). An encoder trained with MCInfoNCE correctly recovers this posterior of the generative process, up to a rotation, from contrastive supervision.

aleatoric uncertainty of the individual image). Our work thus generalizes the previous theoretical result in nonlinear ICA to a broader class of generative processes, and provides a theoretical foundation for probabilistic embeddings.

We show empirically that an encoder trained with MCInfoNCE learns the correct posteriors in a controlled experiment with known posteriors. We find that it even provides sensible embeddings when the distribution family or the encoder dimensionality is misspecified and when the generative process may be injective, making it robust in practice. We then show that these predicted uncertainties are consistent with human annotator disagreements reported in the recent CIFAR-10H dataset (Peterson et al., 2019), providing a way to handle uncertainty for high-dimensional inputs. We also demonstrate that knowing the true posteriors enables new applications, such as computing credible intervals for image retrieval tasks. They visualize how uncertain we are about a query image by showing other images that represent the region of latents the query is in with a given probability.

In summary, **(1)** We extend nonlinear ICA to non-injective non-deterministic generative processes to model realistic input ambiguities. **(2)** We propose MCInfoNCE for training encoders that predict probabilistic embeddings. **(3)** We show theoretically and empirically that the predicted posteriors are correct and reflect the true amount of aleatoric uncertainty.

2. Related Works

Our work serves as a bridge between the theoretical understanding of contrastive learning via nonlinear ICA, probabilistic embeddings, and recent discussions on the aleatoric uncertainty inherent in vision problems. Below, we discuss how our work extends and connects recent work in these three fields. Extended literature reviews can be found in Kendall & Gal (2017) and Karpukhin et al. (2022).

Nonlinear ICA. From a nonlinear Independent Component Analysis (ICA) perspective (Hyvärinen & Oja, 2000; Comon & Jutten, 2010), images x are generated from ground-truth latent components z via an unknown nonlinear generative process. The goal is to invert it to recover the original latents z, which are useful for downstream tasks. This formalization allows for theoretical proofs of which (contrastive) losses achieve this. Building on Wang & Isola (2020), Zimmermann et al. (2021) recently proved that optimizing a contrastive InfoNCE loss (Oord et al., 2018) recovers z up to a rotation of the latent space, as visualized in Figure 1. This requires certain assumptions about the generative process. A recent strain of literature seeks to reduce these assumptions (Leemann et al., 2022) to allow modeling broader classes of generative processes, bringing the theoretical results closer to practice. Our work broadens this class by no longer requiring the injectivity assumption of Zimmermann et al. (2021) and at the same time allowing stochasticity. This is made possible by modeling the generative process as a conditional distribution $P(x|z)$ instead of a function, which generalizes the class of generative processes. In the vein of Zimmermann et al. (2021), we prove that our contrastive MCInfoNCE loss recovers the correct posterior distribution $P(z|x)$ of the original latents, up to a rotation of the latent space.

Aleatoric Uncertainty. The above generalization allows us to model scenarios in which we encounter aleatoric uncertainty, i.e., the input has reduced information such that z is only recoverable only up to some uncertainty. A prominent practical example is face recognition, where images may be blurred or in low-resolution (Shi & Jain, 2019; Schlett et al., 2022). Other problems with ambiguous inputs include 3D reconstruction from 2D data (Chen et al., 2021), partially occluded traffic participants (Kraus & Dietmayer, 2019), or noisy physical sensors (Meech & Stanley-Marbell, 2021). Such problems with aleatoric uncertainty can be detected by label noise: CIFAR-10H (Peterson et al., 2019) comprises multiple labels for each image in the CIFAR-10 test-set, and shows that the more ambiguous an image is, the more annotator labels disagree. This finding occurs in several other recent classification datasets (Schmarje et al., 2022; Mehrtens et al., 2023; Tran et al., 2022), but also in more complex tasks such as multimodal visual question answering (VQA). Chun et al. (2022) show that there are many possible textual answers to the same visual prompt because language is inherently more ambiguous than vision; i.e., language has more aleatoric uncertainty. Our MCInfoNCE loss explicitly accounts for these uncertainties and learns the correct level of aleatoric uncertainty, which we demonstrate on high-dimensional image inputs.

Probabilistic Embeddings. An emerging approach to modeling this uncertainty is to have encoders predict distributions over the latent space instead of point estimates. There are three main lines of work to learn these probabilistic embeddings. The first idea is to compute a match probability between point estimates, but to integrate it over the predicted distributions. This idea was pioneered via Hedged Instance Embeddings (HIB) (Oh et al., 2019) and has since been successfully extended, e.g., to the above multimodal VQA problem (Chun et al., 2021; Neculai et al., 2022). A second line of works turns existing losses into probabilistic ones by integrating the whole loss over the predicted probabilistic embeddings (Scott et al., 2021; Roads & Love, 2021). Our MCInfoNCE extension of InfoNCE demonstrates that this blueprint strategy can inherit the properties of the original losses, like Zimmermann et al. (2021)'s identifiability theorem. The third line of works provides distribution-to-distribution distances to replace point-to-point distances in losses. The most popular approach is the expected likelihood kernel (ELK) (Jebara & Kondor, 2003; Shi & Jain, 2019). It has recently shown success even in high dimensional embedding spaces (Kirchhof et al., 2022; Karpukhin et al., 2022). Yet, there is no answer to whether and in what sense the predicted probabilistic embeddings, and in particular their variances, are *correct*. Our work answers this question through its proof and a controlled experiment where the true posteriors are recovered. The experiments on CIFAR-10H further ground this theoretical correctness in the human perception of uncertainty. We also show novel practical applications of probabilistic embeddings, such as retrieving credible intervals on which latents the image might show.

3. Probabilistic Generative Processes

In this section, we extend the generative processes commonly used in nonlinear ICA to non-injective, randomized ones. This allows modeling real-world image distributions better and serves as a framework for the upcoming proof.

Let us first understand the class of generative processes for which Zimmermann et al. (2021) prove identifiability. They take the nonlinear ICA perspective that there is a natural generative process g that transforms latent components $z \in \mathcal{Z}$ into the images $x = g(z)$ we observe, as shown in Figure 1. Following the popular cosine-based similarity comparisons (Deng et al., 2019; Teh et al., 2020), $\mathcal{Z}$ is assumed to be a D-dimensional hypersphere $\mathcal{Z} = \mathcal{S}^{D-1}$. We are interested in recovering the latents z that underlie the images x, because they are low-dimensional descriptions useful for downstream tasks. To formalize this problem, they assume that $g : \mathcal{Z} \to \mathcal{X}$ is an injective (and deterministic) function. Thus, only one latent z can correspond to each image x, and g is invertible. They prove that an encoder f trained with a contrastive InfoNCE loss achieves this inversion and recovers the correct latent z, i.e., $f(x) = f(g(z)) = \hat{z} = Rz$, up to an orthogonal rotation R of the learned embedding space.

However, let us move on to setups where an image x may be motion blurred, low-resolution, or partially obscured. For instance, a 2D projection x of a 3D object z does not show the back part of z, and there are several possible z that could have generated x. In other words, the generative process g is non-injective and the best our encoder can do is to recover the set of possible latents $\{\hat{z} | g(\hat{z}) = x\}$. Further, g may be stochastic. E.g., a random patch of pixels may be occluded, or the image may be zoomed in and show only a random crop of z. The best the encoder can do is to predict a posterior over the possible latents, see Figure 1.

The common denominator of these setups is that g loses information about z and x becomes ambiguous. To subsume them, we can model g as a likelihood $P(x|z)$. This general formulation allows for a large class of operations within g. However, this generality comes at the cost that $P(x|z)$ can be very complicated and difficult to parameterize. We therefore apply a *posterior trick*: instead of explicitly characterizing g by $P(x|z)$ we implicitly characterize it by its posteriors $P(z|x)$. We parameterize $P(z|x)$ by simple von Mises-Fisher distributions vMF$(z; \mu(x), \kappa(x))$:

$$P(z|x) = C(\kappa(x)) e^{\kappa(x)\mu(x)^\top z} \; . \tag{1}$$

This distribution on $\mathcal{S}^{D-1}$ is unimodal around the location parameter $\mu(x) \in \mathcal{Z}$ with a certain concentration (i.e., an *inverse* variance) $\kappa(x) \in \mathbb{R}_{>0}$, and a normalizing constant $C(\cdot)$. The functions $\mu : \mathcal{X} \to \mathcal{S}^{D-1}$ and $\kappa : \mathcal{X} \to \mathbb{R}_{>0}$ fully parameterize the posterior of each image x. In particular, $\kappa(\cdot)$ represents the aleatoric uncertainty due to information loss, which can be heterogeneous across the images.

The intuition behind modeling the posterior of the generative process as a vMF is that latents of degraded images can usually be located down to sets of semantically similar rather than very dissimilar latents. This is reflected in the unimodality of the vMF and its use of the dot product, which commonly represents how semantically similar two latents are. There may still be images where it is impossible to tell which highly dissimilar latents they show. In these cases, $\kappa(x)$ is low and the posterior spreads broadly across the latent space. At the other end of the spectrum, as $\kappa(x) \to \infty$, $P(z|x)$ converges to a Dirac distribution. This allows modelling deterministic and injective generative processes as in Zimmermann et al. (2021). This makes the vMF a reasonable and flexible choice for the posterior of generative processes.

4. Probabilistic Contrastive Learning

This section presents our main theoretical result: a probabilistic encoder trained under an MCInfoNCE loss recovers the true posteriors of probabilistic generative processes, up to a rotation, from simple contrastive supervision.

4.1. MCInfoNCE for Probabilistic Contrastive Learning

Let us first formalize the contrastive learning setup. Each training triplet comprises a reference sample x along with a positive (similar) sample x^+ and negative (dissimilar) samples $x_1^-, \dots, x_M^-$ against which it is to be contrasted. As introduced in the previous section, we assume that these samples are generated from corresponding latents $z, z^+, z_1^-, \dots, z_M^-$. Following Zimmermann et al. (2021), the reference z is drawn from the marginal distribution in the latent space, a uniform distribution. The positive sample z^+ is drawn from a close region around z, while negatives $z_1^-, \dots, z_M^-$ are random i.i.d. draws from the marginal:

$$z \sim P(z) = \text{Unif}(z; \mathcal{S}^{D-1}), \tag{2}$$

$$z^+ \sim P(z^+|z) = \text{vMF}(z^+; z, \kappa_{\text{pos}}), \tag{3}$$

$$z_m^- \sim P(z^-|z) =: P(z^-) = \text{Unif}(z^-; \mathcal{S}^{D-1}). \tag{4}$$

The fixed constant $\kappa_{\text{pos}} > 0$ controls how close latents must be to be considered positive to each other[1]. This formalization of contrastive learning ensures that positive samples are semantically similar and negatives are dissimilar. Zimmermann et al. (2021) showed this is the generative process InfoNCE implicitly assumes. The probabilistic generative process comes into play when the latents $z, z^+, z_1^-, \dots, z_M^-$ are transformed into observations $x, x^+, x_1^-, \dots, x_M^-$ via $P(x|z)$. This defines $P(x)$, $P(x^+|x)$, and $P(x^-)$, and thus our contrastive training data $(x, x^+, x_1^-, \dots, x_M^-)$.

Our Monte-Carlo InfoNCE (MCInfoNCE) loss is

$$L_f := -\log \mathop{\mathbb{E}}_{\substack{z\sim Q(z|x) \\ z^+\sim Q(z^+|x^+) \\ z_m^-\sim Q(z_m^-|x_m^-), m=1,\dots,M}} \left(\frac{e^{\kappa_{\text{pos}} z^\top z^+}}{\frac{1}{M} e^{\kappa_{\text{pos}} z^\top z^+} + \frac{1}{M} \sum_{m=1}^{M} e^{\kappa_{\text{pos}} z^\top z_m^-}} \right) \tag{5}$$

and is evaluated over the contrastive training dataset via

$$\mathcal{L} := \mathop{\mathbb{E}}_{\substack{x\sim P(x) \\ x^+\sim P(x^+|x) \\ x_m^-\sim P(x^-), m=1,\dots,M}} \left(L_f \left(x, x^+, \{x_m^-\}_{m=1,\dots,M} \right) \right). \tag{6}$$

This probabilistically generalizes the widely used InfoNCE family (Oord et al., 2018), and, in the limit of $M \to \infty$, SimCLR (Chen et al., 2020). Instead of outputting a point embedding, the encoder f we train outputs probabilistic embeddings $Q(z|x) := \text{vMF}(z; \hat{\mu}(x), \hat{\kappa}(x))$ by predicting $f(x) = (\hat{\mu}(x), \hat{\kappa}(x))$. The InfoNCE fraction within L_f is evaluated over these posteriors. In practice, we backpropagate through $K = 512$ MC samples via a reparametrization trick for vMFs (Davidson et al., 2018; Ulrich, 1984):

$$L_f \approx -\log \left(\frac{1}{K} \sum_{k=1}^{K} \frac{e^{\kappa_{\text{pos}} z_k^\top z_k^+}}{\frac{1}{M} e^{\kappa_{\text{pos}} z_k^\top z_k^+} + \frac{1}{M} \sum_{m=1}^{M} e^{\kappa_{\text{pos}} z_k^\top z_{m,k}^-}} \right). \tag{7}$$

The only training data for MCInfoNCE are contrastive examples, without any additional supervision on the true aleatoric uncertainty $\kappa(x)$ or the generative latents z.

4.2. Provably Learning the Correct Posteriors

We prove below that the optimizer of this loss learns the *correct* latent posteriors. More precisely, it predicts the correct location $\hat{\mu}(x) = R \cdot \mu(x)$, up to a constant orthogonal rotation R of the latent space, and the correct level of ambiguity $\hat{\kappa}(x) = \kappa(x)$ for each observation x. To prove this, we first show that MCInfoNCE is a cross-entropy between the generative process and the learned contrastive encoder (Proposition 4.1). This means that the loss matches the expected positivity of a pair (x, x^+) computed using the true $P(z|x)$ to that computed using $Q(z|x)$. We then show that this expected positivity can be written as a function and depends only on $(\mu(\cdot)^\top \mu(\cdot), \kappa(\cdot))$, resp. $(\hat{\mu}(\cdot)^\top \hat{\mu}(\cdot), \hat{\kappa}(\cdot))$ (Proposition 4.2). Due to monotonicity, the predicted function value can only match that of the generative process if their arguments $(\mu(\cdot)^\top \mu(\cdot), \kappa(\cdot))$ and $(\hat{\mu}(\cdot)^\top \hat{\mu}(\cdot), \hat{\kappa}(\cdot))$ are equal (Proposition 4.3). In summary, the posteriors must be equal, up to a rotation of the latent space (Theorem 4.4).

[1] κ_{pos} should not to be confused with $\kappa(x)$, which controls the heteroscedastic uncertainty of the generative process.

First, we generalize Zimmermann et al. (2021) and Wang & Isola (2020) to probabilistic generative processes.

Proposition 4.1 ($\mathcal{L}$ is minimized iff expected positivity matches)**.** *Let the latent marginal $P(z) = \int P(z|x)dP(x)$ and $\int Q(z|x)dP(x)$ be uniform. $\lim_{M\to\infty} \mathcal{L}$ attains its minimum when $\forall x, x^+ \in \{x \in \mathcal{X} | P(x) > 0\}$*

$$\iint Q(z|x)Q(z^+|x^+)P(z^+|z)dz^+dz =$$
$$\iint P(z|x)P(z^+|x^+)P(z^+|z)dz^+dz \ .$$

The intuition is that MCInfoNCE corresponds to a cross-entropy between the true latents and our model predictions. This characterizes the solution set: An encoder Q minimizes MCInfoNCE if and only if the chance of (x, x^+) being a positive pair *computed using* Q is equal to the *true chance* of being a positive pair *computed using the GT distribution* P for all data pairs (x, x^+). We refer to this chance, the upper integral, as expected positivity. Next, we prove that the equality of the expected positivities implies that the predicted posteriors Q must be equal to the GT P, up to the mentioned rotations. To this end, we first find that the expected positivity marginalizes out all random variables and can be written as a *function* of $\mu(x)$ and $\kappa(x)$.

Proposition 4.2 (Expected positivity is a function)**.** *Let $P(z|x)$ and $P(z^+|z)$ be vMF distributions as defined in Section 4.1. Given $x, x^+ \in \mathcal{X}$, we can rewrite*

$$\iint P(z|x)P(z^+|x^+)P(z^+|z)dz^+dz \tag{8}$$
$$=: h_{\kappa_{pos}}(\mu(x)^\top \mu(x^+), \kappa(x), \kappa(x^+)), \tag{9}$$

i.e., as a function $h_{\kappa_{pos}}$ that depends only on $\mu(x)^\top\mu(x^+), \kappa(x)$, and $\kappa(x^+)$. The same function can be used for $\hat{\mu}(x)^\top\hat{\mu}(x^+), \hat{\kappa}(x), \hat{\kappa}(x^+)$:

$$\iint Q(z|x)Q(z^+|x^+)P(z^+|z)dz^+dz \tag{10}$$
$$= h_{\kappa_{pos}}(\hat{\mu}(x)^\top \hat{\mu}(x^+), \hat{\kappa}(x), \hat{\kappa}(x^+)). \tag{11}$$

The key is that the expected positivities calculated using Q and P have the *same* functional form h_{pos}; they differ only in their arguments, where they use either the true $\kappa(x), \mu(x)$ or the predicted $\hat{\kappa}(x), \hat{\mu}(x)$. What remains to show is that the expected positivities can only be equal if the arguments match, i.e., $\hat{\kappa}(x) = \kappa(x)$ and $\hat{\mu}(x)^\top\hat{\mu}(x^+) = \mu(x)^\top\mu(x^+)$. Proposition 4.3 proves this via some monotonicities of h_{pos}.

Proposition 4.3 (Arguments of h_{pos} must be equal)**.** *Define h_{pos} as in Proposition 4.2. Let $\mathcal{X}' \subseteq \mathcal{X}$, $\mu, \hat{\mu} : \mathcal{X}' \to \mathcal{Z}$, $\kappa, \hat{\kappa} : \mathcal{X}' \to \mathbb{R}_{>0}$, $\kappa_{pos} > 0$. If $h_{\kappa_{pos}}(\hat{\mu}(x)^\top\hat{\mu}(x^+), \hat{\kappa}(x), \hat{\kappa}(x^+)) = h_{\kappa_{pos}}(\mu(x)^\top\mu(x^+), \kappa(x), \kappa(x^+))\ \forall x, x^+ \in \mathcal{X}'$, then*

$$\hat{\mu}(x)^\top\hat{\mu}(x^+) = \mu(x)^\top\mu(x^+) \text{ and} \tag{12}$$
$$\hat{\kappa}(x) = \kappa(x)\ \ \forall x, x^+ \in \mathcal{X}'. \tag{13}$$

In the above Equation (12), the pairwise cosine similarities in the true and the predicted latent space can only be equal if the two spaces are the same up to a rotation, i.e., $\hat{\mu}(x) = R\mu(x)$. This is ensured by the Extended Mazur-Ulam Theorem (Zimmermann et al., 2021). We can now combine these ingredients to derive our main result: If an encoder minimizes the MCInfoNCE loss, then it must have identified the correct posteriors, up to a constant orthogonal rotation of the latent space.

Theorem 4.4 ($\mathcal{L}$ identifies the correct posteriors)**.** *Let $\mathcal{Z} = \mathcal{S}^{D-1}$ and $P(z) = \int P(z|x)dP(x)$ and $\int Q(z|x)dP(x)$ be the Unif$(z; \mathcal{Z})$. Let g be a probabilistic generative process defined in Formulas 2, 3, and 4 with known[2] κ_{pos}. Let g have vMF posteriors $P(z|x) =$ vMF$(z; \mu(x), \kappa(x))$ with $\mu : \mathcal{X} \to \mathcal{S}^{D-1}$ and $\kappa : \mathcal{X} \to \mathbb{R}_{>0}$. Let an encoder $f(x)$ parametrize vMF distributions vMF$(z; \hat{\mu}(x), \hat{\kappa}(x))$. Then $f^* = \arg\min_f \lim_{M\to\infty} \mathcal{L}$ has the correct posteriors up to a rotation, i.e., $\hat{\mu}(x) = R\mu(x)$ and $\hat{\kappa}(x) = \kappa(x)$, where R is an orthogonal matrix, $\forall x \in \{x \in \mathcal{X} | P(x) > 0\}$.*

This generalizes the recent results of Zimmermann et al. (2021) to the broader family of probabilistic generative processes. MCInfoNCE recovers not only the correct (mean) embeddings $\mu(x)$ under a noisy and non-injectivity generator, but also the heterogeneous aleatoric uncertainty $\kappa(x)$.

5. Experiments

5.1. MCInfoNCE Learns the Correct Posteriors

In this section, we experimentally confirm the theoretical result that *probabilistic embeddings learned under a MCInfoNCE loss recover the correct posteriors up to a rotation*. We also test its robustness to violated assumptions.

Setup. To test whether MCInfoNCE recovers the correct posteriors, we need a controlled experiment where the true posteriors of the generative process are known. Previous nonlinear ICA experiments randomly initialize a multilayer perceptron (MLP) as the nonlinear data-generating process and train a second one to invert it (Hyvarinen & Morioka, 2017; Zimmermann et al., 2021). In our probabilistic setup we randomly initialize two MLPs to parameterize $\mu(x)$ and $\kappa(x)$ of the vMF posteriors of the generative process. The MLP for $\mu(x)$ outputs normalized vectors of dimension $D = 10$ and the MLP for $\kappa(x)$ outputs a scalar $\tilde{\kappa}(x)$ wrapped in an exponential Softplus function $\kappa(x) = 1 + \exp(\tilde{\kappa}(x))$ to ensure the strict positivity of $\kappa(x)$

[2] In practice, κ_{pos} is a tuneable temperature hyperparameter.

Table 1. MCInfoNCE recovers the generative processes' true posteriors for various degrees of ambiguity and even in the limit of an injective generative process. Mean ± std. err. for five seeds.

	True vs Pred. Location $\hat{\mu}(x)$		True vs Pred. Certainty $\hat{\kappa}(x)$	
Generative Process Ambiguity	RMSE ↓	Rank Corr. ↑	RMSE ↓	Rank Corr. ↑
Ambiguous ($\kappa(x) \in [16, 32]$)	0.04 ± 0.00	0.99 ± 0.00	6.15 ± 0.61	0.82 ± 0.04
Clear ($\kappa(x) \in [64, 128]$)	0.05 ± 0.00	0.98 ± 0.00	125.02 ± 10.64	0.64 ± 0.04
Injective ($\kappa(x) = \infty$)	0.05 ± 0.01	0.98 ± 0.00	$\hat{\kappa}(x) \to \infty$	

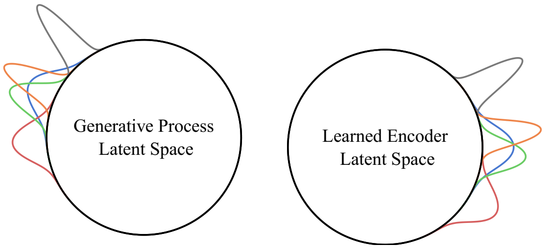


Figure 2. Five posteriors of the generative process and the encoder trained in a run with a 2D latent space. The encoder correctly predicts the posteriors of the generative process, up to a rotation: Rank corr. between $\hat{\mu}(x)$ and the true $\mu(x)$ is 1.00 ± 0.00 (RMSE 0.05 ± 0.00) and that of $\hat{\kappa}(x)$ is 0.82 ± 0.05 (RMSE 2.89 ± 0.56).

Table 2. MCInfoNCE predicts sensible vMF posteriors if the true generative posteriors are non-vMF. Mean ± std. err. for five seeds.

	True vs Pred. Location $\hat{\mu}(x)$		True vs Pred. Spread	
Posterior	RMSE ↓	Rank Corr. ↑	RMSE ↓	Rank Corr. ↑
vMF	0.04 ± 0.00	0.99 ± 0.00	0.05 ± 0.00	0.75 ± 0.04
Gaussian	0.04 ± 0.00	0.99 ± 0.00	0.04 ± 0.00	0.70 ± 0.05
Laplace	0.05 ± 0.01	0.98 ± 0.00	0.02 ± 0.00	0.66 ± 0.06

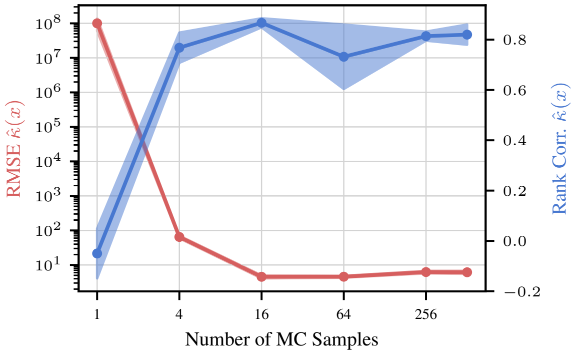


Figure 3. The marginal likelihood approximation bias diminishes with sufficient MC samples. Mean ± std. err. for five seeds.

(Li et al., 2021; Shi & Jain, 2019). We sample contrastive training data $(x, x^+, (x_m^-)_{m=1,\ldots,M})$ from the generative process parameterized by $\mu(x)$ and $\kappa(x)$ via rejection sampling, as explained in the supplementary. On this data, we train two MLPs to predict $\hat{\mu}(x)$ and $\hat{\kappa}(x)$. All hyperparameters of the generative process and MLP architectures follow the deterministic counterpart of this experiment in Zimmermann et al. (2021) and are reported in the supplementary.

Metrics. To quantify if the predicted posteriors are correct up to a rotation, i.e., $\hat{\kappa}(x) = \kappa(x)$ and $\hat{\mu}(x) = R\mu(x)$ with an orthogonal matrix R, we compare $\hat{\kappa}(x)$ to $\kappa(x)$ on 10^4 samples of x and compare $\hat{\mu}(x_1)^\top \hat{\mu}(x_2)$ to $\mu(x_1)^\top \mu(x_2)$ on all pairs (x_1, x_2) of the 10^4 samples. We use the root mean square error (RMSE) to test for exact correctness and Spearman's rank correlation (Rank Corr.) to test for correct ordering. The latter is sufficient in practical scenarios that are invariant to scale, such as retrieval based on embedding distances $\hat{\mu}(x_1)^\top \hat{\mu}(x_2)$ or abstention from prediction based on a threshold of the predicted certainty $\hat{\kappa}(x)$.

Results. Table 1 shows that MCInfoNCE recovers the correct posteriors of ambiguous inputs up to a high rank correlation of 0.99 for $\hat{\mu}(x)$ and 0.82 for $\hat{\kappa}(x)$. Figure 2 visualizes this in a simplified 2D case. The learned latent space equals the true latent space up to a rotation. However, we can see in Table 1 that $\hat{\kappa}(x)$ tends to be overconfident (RMSE = 125.02) especially for high values of $\kappa(x) \in [64, 128]$ (yet, the ranking is still largely preserved, Rank Corr. = 0.64). This is because Formula 7 is a biased MC estimator of the loss in Formula 5. This is also known as marginal likelihood estimation problem (Perrakis et al., 2014; Burda et al., 2015). The bias decreases with the number of MC samples, as shown in Figure 3. In the standard setup with $\kappa(x) \in [16, 32]$, it is largely mitigated with 16 samples (RMSE = 4.55), or already with 4 samples if only the relative ordering of the samples matters in practice (Rank Corr. = 0.77). This coincides with the range of number of MC samples used by other probabilistic embedding losses: Oh et al. (2019) use 10 and Kirchhof et al. (2022) use 5. In summary, MCInfoNCE behaves as theoretically expected and fulfills our main theoretical hypothesis.

Violated Assumptions. We test MCInfoNCE in setups where its assumptions are violated. First, we change the posterior of the generative process to Gaussian and Laplace distributions on $\mathcal{S}^{D-1}$ while the encoder still predicts vMFs. Since these distributions have incomparable variance parameters, we measure their spread by the avg. absolute cosine distance from the mode. Table 2 shows that the vMFs model Gaussians almost as well as vMFs (Rank Corr. 0.70 vs 0.75), since Gaussians with normalized outputs are similar to vMFs (Mardia et al., 2000). For Laplace, the encoder predicts vMFs with high concentrations ($\hat{\kappa}(x) \approx 2000$), because the Laplace distribution is more concentrated around its mode than the vMF the encoder uses. Second, we over- and underparameterize the latent dimension of the encoder compared to that of the generative process ($D = 10$). Figure 4 shows that encoder dimensions between 8 and 32 still all yield $\hat{\kappa}$ predictions with a Rank Corr. ≥ 0.6. Third, we test the behaviour of MCInfoNCE when the generative pro-

Table 3. Besides MCInfoNCE, ELK also gives correct probabilistic embeddings. Mean ± std. err. for five seeds.

	True vs Pred. Location $\hat{\mu}(x)$		True vs Pred. Certainty $\hat{\kappa}(x)$	
Loss	RMSE ↓	Rank Corr. ↑	RMSE ↓	Rank Corr. ↑
HIB	0.18 ± 0.02	0.82 ± 0.03	$10^{14} \pm 10^{14}$	-0.02 ± 0.09
ELK	0.02 ± 0.00	1.00 ± 0.00	21.70 ± 0.31	0.92 ± 0.00
MCInfoNCE	0.04 ± 0.00	0.99 ± 0.00	6.15 ± 0.61	0.82 ± 0.04

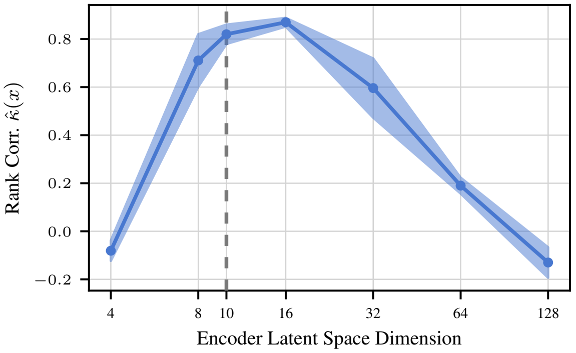


Figure 4. MCInfoNCE learns good $\hat{\kappa}(x)$ even when the encoder latent space dimension mismatches the true generative dimensionality ($D = 10$). Mean ± std. err. for five seeds.

Table 4. Predicted certainties $\hat{\kappa}(x)$ of MCInfoNCE correlate with human annotator disagreement and information reduction via cropping images smaller. Rank correlation on unseen test data.

Loss	Annotator Entropy ↑	Crop Size ↑
HIB	0.28 ± 0.00	0.69 ± 0.02
ELK	0.14 ± 0.05	0.51 ± 0.03
MCInfoNCE	0.29 ± 0.01	0.68 ± 0.01

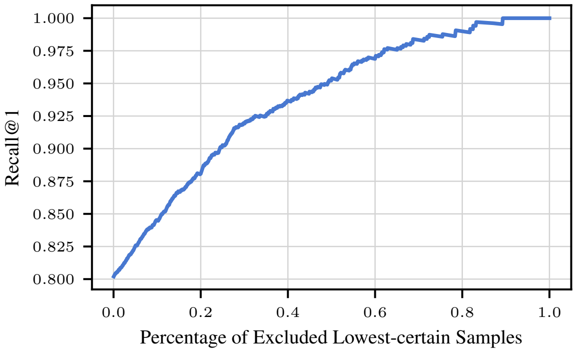


Figure 5. Rejecting images with low certainty values $\hat{\kappa}(x)$ improves the performance on the remaining data monotonically with the threshold. This shows that $\hat{\kappa}(x)$ is predictive of performance.

cess is injective and deterministic, i.e., when all posteriors are Diracs. This is a limiting case of the vMFs the encoder uses. Table 1 shows that the predicted vMFs converge to infinite concentrations $\hat{\kappa}(x)$, recovering the Diracs. Last, the uniformity assumption was violated in all experiments as we only ensured $\mu(x)$ to be not collapsed, but not necessarily fully spread around $\mathcal{S}^{D-1}$. In summary, these results indicate that MCInfoNCE is a robust approach even when characteristics of the generative process such as its (non-) injectivity, posterior family, or dimension are unknown.

Further losses. Recent literature has proposed other losses to predict probabilistic embeddings. We investigate their empirical successes further under our experimental setup to find whether they *exactly* match the true posteriors. We reimplement Hedged Instance Embeddings (HIB) (Oh et al., 2019) and Expected Likelihood Kernels (ELK) (Kirchhof et al., 2022) and modify them to our contrastive setup, as detailed in the supplementary. All losses are hyperparameter tuned via grid search. Table 3 shows that all losses recover $\mu(x)$ with a Rank Corr. ≥ 0.82 despite the high noise in our experimental setup. We find that, besides MCInfoNCE, ELK also recovers $\kappa(x)$ well (Rank Corr. $= 0.92$). This is the first confirmation that ELK predicts correct posteriors in a controlled setup and opens space for future theoretical investigations.

5.2. Posteriors Reflect Aleatoric Uncertainty in Practice

After confirming that the predicted posteriors are correct, this section shows that they resemble the aleatoric uncertainty in image data. We also show that this enables novel applications such as credible intervals for image retrieval.

Measuring Aleatoric Uncertainty. In the upcoming experiment, we do not have access to any ground-truth $\kappa(x)$ against which to compare $\hat{\kappa}(x)$. Instead, we need to compare it to various indicators of aleatoric uncertainty. We use three different indicators that capture human uncertainty, information loss, and performance decrease with respect to the amount of aleatoric uncertainty. First, if an image is ambiguous, human annotators disagree about the latent that it shows. We therefore conduct our experiment on CIFAR-10H (Peterson et al., 2019). It comprises fifty class annotations for each image. This gives a soft-label distribution whose entropy reflects the ambiguity of the image. We compute the Rank Corr. between $1/\hat{\kappa}(x)$ and this annotator entropy to measure how well $\hat{\kappa}(x)$ reflects human-perceived input ambiguity. Second, we induce controlled information loss by deteriorating the image. (Wu & Goodman, 2020) identified cropping to increase aleatoric uncertainty most clearly. Thus, we crop test images to percentages $\texttt{crop\_size} \sim \text{Unif}([0.25, 1])$ of their original size. The aleatoric uncertainty increases the more the image is cropped. We thus report the Rank Corr. between $1/\hat{\kappa}(x)$ and the crop size as a second met-

Table 5. $\hat{\kappa}(x)$ can be learned by MCInfoNCE from both soft and hard labels. Rank correlation on unseen test data.

Labels	Annotator Entropy ↑	Crop Size ↑
CIFAR-10H Soft Labels	0.29 ± 0.01	0.68 ± 0.01
CIFAR-10H Hard Labels	0.24 ± 0.01	0.64 ± 0.02
CIFAR-10 Hard Labels	0.28 ± 0.01	0.69 ± 0.02

ric. Third, ambiguous images inevitably lead to decreased performance. To investigate whether $\hat{\kappa}(x)$ is indicative of performance, we calculate the Recall@1 (Jegou et al., 2010) on the $p\%$ images with the highest $\hat{\kappa}(x)$. If $\hat{\kappa}(x)$ correctly reflects aleatoric uncertainty, removing ambiguous images should improve performance, so the Recall@1 should increase monotonically with p. This metric also illustrates the popular use case of abstaining from uncertain predictions.

Architecture and Training. We translate the CIFAR-10H classification task into a contrastive task by considering images to be positive if they are in the same class and negative otherwise. We create training examples $(x, x^+, x_1^-, \dots, x_M^-)$ by drawing class labels for each image from its soft class distribution, selecting a random image x, an image x^+ with the same class label, and M images x_m^- with different class labels. On this data, we train a ResNet-18 (He et al., 2016) pre-trained on CIFAR-10 (Phan, 2021) that outputs embeddings $e(x)$. We define $\hat{\mu}(x) := e(x)/\|e(x)\|_2$ and, following common practices for probabilistic embeddings (Kirchhof et al., 2022; Scott et al., 2021; Li et al., 2021), $\hat{\kappa}(x)$ as $\|e(x)\|_2$. We run a 5-fold cross validation where we train for 175 epochs and select the best epoch via the Rank Corr. with the crop size on validation data. We choose this metric over the others because it can be computed on any dataset without additional supervision. All details on generating the contrastive data and the hyperparameter search are in the supplementary.

Results. Table 4 shows that $\hat{\kappa}(x)$ learned via MCInfoNCE has a high Rank Corr. of 0.68 with the information lost due to cropping, i.e., images with less information return more uncertain posteriors. The correlation with the human annotator entropy is lower (0.29), but positive. HIB achieves a similar performance, while ELK shows lower correlations with both ground-truths (0.51 and 0.14, resp.). Figure 5 shows the performance decrease metric. Up to noise, the Recall@1 increases monotonically as images with the lowest $\hat{\kappa}(x)$ are rejected. This means that $\hat{\kappa}(x)$ is a good predictor of performance. As an additional qualitative metric the supplementary shows images with the lowest and highest $\hat{\kappa}(x)$ of each class. MCInfoNCE learns from labeling noise in this experiment, since the image class was drawn anew from its soft label distribution each time the image was used. In practice, we may have only one annotation per image, so that labeling noise occurs across examples rather than on each individual image. To this end, we further train on hard labels. These are either the most likely class of each soft label distribution on CIFAR-10H or the classical class labels on the CIFAR-10. Table 5 shows that MCInfoNCE can learn under both of these circumstances with a performance roughly equal to that when soft labels are available.

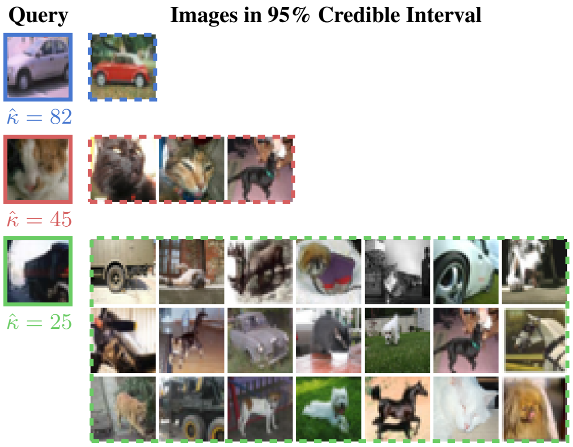


Figure 6. We use an image's posterior to define the credible interval that its latents lie in with a given probability. Clear query images (top) have small credible intervals containing images of the same class as the query. More ambiguous queries (bottom) return larger credible intervals with images from multiple possible classes.

Credible Intervals for Image Retrieval. Since we estimate posteriors $Q(z|x)$, we can also introduce Bayesian credible intervals (Lee, 1989) to our image representation task. Such intervals $\mathrm{CI}_p(x) \subset \mathcal{Z}$ contain the true generative latent z of x with a user-defined probability $p \in [0, 1]$, i.e., $P(z \in \mathrm{CI}_p(x)) = p$ for $x \sim P(x|z)$. Credible intervals help understand the degree to which our model can identify the latent that x shows. We can visualize these latents by searching for images whose $\hat{\mu}(x)$ fall within CI_p. Figure 6 shows such intervals on our MCInfoNCE model for CIFAR-10H. A clear image (top) has a sharp posterior and thus a small CI containing only one image from the same class. The CI of a more ambiguous query image, like the second, tells us that the model places the query in the region of cats, but that it could also be a dog. Highly ambiguous queries, like the last one, lead to wide CIs that span multiple possible classes. They examples show how credible intervals can augment retrieval with uncertainty-awareness: They determine the number of images to retrieve subject to the query's ambiguity and allow users to judge the uncertainty better than a simple scalar uncertainty value.

6. Discussion

Relations to Broader Variational Inference. Our work advances the recent theoretical discussions about contrastive learning and variational inference. Oord et al. (2018) and Poole et al. (2019) initially showed that the minimizer of InfoNCE is the likelihood ratio of positive and negative densities of the generative process. Zimmermann et al. (2021) used this to show that the minimizer recovers the latents, modulo rotations. Our work shows that we can even learn the correct posterior of a *probabilistic* generative process, modulo rotations, i.e., the internal probabilistic latent representations of our specific encoder are indeed *correct*. This may have implications to other works on variational approaches and contrastive learning, like Aitchison (2021).

Multi-modal Posteriors. The vMF posteriors should be able to capture most augmentations in self-supervised contrastive learning that deteriorate the image whole image, i.e., all latent factors equally. However, it is also interesting to think about deteriorations that lead to multi-modal posteriors. In this case, Proposition 4.1 does not make any parametric assumption on the posteriors and thus still holds. Proposition 4.2 and Proposition 4.3 need to be extended regarding the identifiability of the mixture component, but could then utilize our propositions for each component. We see this as an exciting direction for future works.

7. Conclusion

This work presented MCInfoNCE, a probabilistic contrastive loss that predicts posteriors instead of points. We proved that it learns the generative processes' true posteriors. This provides a theoretical grounding for the recent probabilistic embeddings literature and connects it to a probabilistic extension of nonlinear ICA. In practice, the posteriors allow predicting the level of aleatoric uncertainty in ambiguous inputs as well as estimating credible intervals with flexible sizes depending on a query's ambiguity in image retrieval. These are only two usages that correct posteriors enable and further usages are a promising area for future research. Aleatoric uncertainty is not only faced in computer vision and retrieval. We hope that the blueprint way of enhancing InfoNCE into MCInfoNCE inspires applications in further tasks with intrinsic ambiguities in their inputs.

Acknowledgements

Kay Choi has helped designing Figure 1. This work was funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) under Germany's Excellence Strategy – EXC number 2064/1 – Project number 390727645. The authors thank the International Max Planck Research School for Intelligent Systems (IMPRS-IS) for supporting Michael Kirchhof.

References

Aitchison, L. InfoNCE is a variational autoencoder. *arXiv preprint arXiv:2107.02495*, 2021.

Ardeshir, S. and Azizan, N. Uncertainty in contrastive learning: On the predictability of downstream performance. *arXiv preprint arXiv:2207.09336*, 2022.

Barbano, R., Arridge, S., Jin, B., and Tanno, R. Uncertainty quantification in medical image synthesis. In *Biomedical Image Synthesis and Simulation*, pp. 601–641. Elsevier, 2022.

Burda, Y., Grosse, R., and Salakhutdinov, R. Importance weighted autoencoders. *arXiv preprint arXiv:1509.00519*, 2015.

Chen, H., Huang, Y., Tian, W., Gao, Z., and Xiong, L. Monorun: Monocular 3d object detection by reconstruction and uncertainty propagation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 10379–10388, 2021.

Chen, T., Kornblith, S., Norouzi, M., and Hinton, G. A simple framework for contrastive learning of visual representations. In *Proceedings of the 37th International Conference on Machine Learning (ICML)*, 2020.

Chun, S., Oh, S. J., De Rezende, R. S., Kalantidis, Y., and Larlus, D. Probabilistic embeddings for cross-modal retrieval. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021.

Chun, S., Kim, W., Park, S., Chang, M. C., and Oh, S. J. ECCV caption: Correcting false negatives by collecting machine-and-human-verified image-caption associations for ms-coco. In *European Conference on Computer Vision (ECCV)*, 2022.

Comon, P. and Jutten, C. *Handbook of Blind Source Separation: Independent component analysis and applications*. 2010.

Davidson, T. R., Falorsi, L., De Cao, N., Kipf, T., and Tomczak, J. M. Hyperspherical variational auto-encoders. *34th Conference on Uncertainty in Artificial Intelligence (UAI)*, 2018.

Deng, J., Guo, J., Xue, N., and Zafeiriou, S. ArcFace: Additive angular margin loss for deep face recognition. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (CVPR)*, pp. 4690–4699, 2019.

Galil, I., Dabbah, M., and El-Yaniv, R. What can we learn from the selective prediction and uncertainty estimation performance of 523 imagenet classifiers? In *International Conference on Learning Representations (ICLR)*, 2023.

He, K., Zhang, X., Ren, S., and Sun, J. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR)*, pp. 770–778, 2016.

Hyvarinen, A. and Morioka, H. Nonlinear ICA of temporally dependent stationary sources. In *Artificial Intelligence and Statistics (AISTATS)*, 2017.

Hyvärinen, A. and Oja, E. Independent component analysis: algorithms and applications. *Neural Networks*, 13(4-5): 411–430, 2000.

Islam, A., Chen, C.-F. R., Panda, R., Karlinsky, L., Radke, R., and Feris, R. A broad study on the transferability of visual representations with contrastive learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 8845–8855, 2021.

Jebara, T. and Kondor, R. Bhattacharyya and expected likelihood kernels. In *Learning Theory and Kernel Machines*. 2003.

Jegou, H., Douze, M., and Schmid, C. Product quantization for nearest neighbor search. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(1):117–128, 2010.

Karpukhin, I., Dereka, S., and Kolesnikov, S. Probabilistic embeddings revisited. *arXiv preprint arXiv:2202.06768*, 2022.

Kendall, A. and Gal, Y. What uncertainties do we need in bayesian deep learning for computer vision? *Advances in Neural Information Processing Systems (NeurIPS)*, 30, 2017.

Khosla, P., Teterwak, P., Wang, C., Sarna, A., Tian, Y., Isola, P., Maschinot, A., Liu, C., and Krishnan, D. Supervised contrastive learning. *Advances in Neural Information Processing Systems (NeurIPS)*, 33:18661–18673, 2020.

Kirchhof, M., Roth, K., Akata, Z., and Kasneci, E. A non-isotropic probabilistic take on proxy-based deep metric learning. In *European Conference on Computer Vision (ECCV)*, 2022.

Kraus, F. and Dietmayer, K. Uncertainty estimation in one-stage object detection. In *2019 IEEE Intelligent Transportation Systems Conference (ITSC)*, pp. 53–60, 2019.

Lee, P. M. *Bayesian statistics*. Oxford University Press London, 1989.

Leemann, T., Kirchhof, M., Rong, Y., Kasneci, E., and Kasneci, G. Disentangling embedding spaces with minimal distributional assumptions. *arXiv preprint arXiv:2206.13872*, 2022.

Lewis, D. D. and Catlett, J. Heterogeneous uncertainty sampling for supervised learning. In *Machine learning proceedings 1994*, pp. 148–156. Elsevier, 1994.

Li, S., Xu, J., Xu, X., Shen, P., Li, S., and Hooi, B. Spherical confidence learning for face recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021.

Mardia, K. V., Jupp, P. E., and Mardia, K. *Directional statistics*, volume 2. Wiley Online Library, 2000.

Meech, J. T. and Stanley-Marbell, P. An algorithm for sensor data uncertainty quantification. *IEEE Sensors Letters*, 6 (1):1–4, 2021.

Mehrtens, H. A., Kurz, A., Bucher, T.-C., and Brinker, T. J. Benchmarking common uncertainty estimation methods with histopathological images under domain shift and label noise. *arXiv preprint arXiv:2301.01054*, 2023.

Neculai, A., Chen, Y., and Akata, Z. Probabilistic compositional embeddings for multimodal image retrieval. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition MULA Workshop (CVPR MULA)*, pp. 4547–4557, 2022.

Oh, S. J., Gallagher, A. C., Murphy, K. P., Schroff, F., Pan, J., and Roth, J. Modeling uncertainty with hedged instance embeddings. In *International Conference on Learning Representations (ICLR)*, 2019.

Oord, A. v. d., Li, Y., and Vinyals, O. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018.

Perrakis, K., Ntzoufras, I., and Tsionas, E. G. On the use of marginal posteriors in marginal likelihood estimation via importance sampling. *Computational Statistics & Data Analysis*, 77:54–69, 2014.

Peterson, J. C., Battleday, R. M., Griffiths, T. L., and Russakovsky, O. Human uncertainty makes classification more robust. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (CVPR)*, pp. 9617–9626, 2019.

Phan, H. Pytorch CIFAR-10 v3.0.1, 2021. URL `https://doi.org/10.5281/zenodo.4431043`.

Poole, B., Ozair, S., Van Den Oord, A., Alemi, A., and Tucker, G. On variational bounds of mutual information. In *International Conference on Machine Learning (ICML)*, 2019.

Roads, B. D. and Love, B. C. Enriching imagenet with human similarity judgments and psychological embeddings. In *Proceedings of the ieee/cvf conference on computer vision and pattern recognition*, pp. 3547–3557, 2021.

Romanazzi, M. Discriminant analysis with high dimensional von mises-fisher distributions. In *8th Annual International Conference on Statistics*, 2014.

Schlett, T., Rathgeb, C., Henniger, O., Galbally, J., Fierrez, J., and Busch, C. Face image quality assessment: A literature survey. *ACM Computing Surveys (CSUR)*, 54 (10s):1–49, 2022.

Schmarje, L., Grossmann, V., Zelenka, C., Dippel, S., Kiko, R., Oszust, M., Pastell, M., Stracke, J., Valros, A., Volkmann, N., et al. Is one annotation enough? A data-centric image classification benchmark for noisy and ambiguous label estimation. *arXiv preprint arXiv:2207.06214*, 2022.

Scott, T. R., Gallagher, A. C., and Mozer, M. C. von Mises-Fisher loss: An exploration of embedding geometries for supervised learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021.

Shi, Y. and Jain, A. K. Probabilistic face embeddings. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2019.

Teh, E. W., DeVries, T., and Taylor, G. W. ProxyNCA++: Revisiting and revitalizing proxy neighborhood component analysis. In *European Conference on Computer Vision (ECCV)*, pp. 448–464, 2020.

Tran, D., Liu, J., Dusenberry, M. W., Phan, D., Collier, M., Ren, J., Han, K., Wang, Z., Mariet, Z., Hu, H., et al. Plex: Towards reliability using pretrained large model extensions. *arXiv preprint arXiv:2207.07411*, 2022.

Ulrich, G. Computer generation of distributions on the m-sphere. *Journal of the Royal Statistical Society: Series C (Applied Statistics)*, 33(2):158–163, 1984.

Wang, T. and Isola, P. Understanding contrastive representation learning through alignment and uniformity on the hypersphere. In *Proceedings of the 37th International Conference on Machine Learning (ICML)*, 2020.

Wang, Y., Tang, S., Zhu, F., Bai, L., Zhao, R., Qi, D., and Ouyang, W. Revisiting the transferability of supervised pretraining: an mlp perspective. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 9183–9193, 2022.

Wu, M. and Goodman, N. A simple framework for uncertainty in contrastive learning. *arXiv preprint arXiv:2010.02038*, 2020.

Zimmermann, R. S., Sharma, Y., Schneider, S., Bethge, M., and Brendel, W. Contrastive learning inverts the data generating process. In *Proceedings of the 38th International Conference on Machine Learning (ICML)*, 2021.

A. Proofs

A.1. Proof of Proposition 4.1

Proposition 4.1 ($\mathcal{L}$ is minimized iff marginals match) Let the latent marginal distributions $P(z) = \int P(z|x)dP(x)$ and $\int Q(z|x)dP(x)$ be uniform. $\lim_{M\to\infty}\mathcal{L}$ attains its minimum when $\forall x, x^+ \in \{x \in \mathcal{X} | P(x) > 0\}$

$$\iint Q(z|x)Q(z^+|x^+)P(z^+|z)dz^+dz = \\ \iint P(z|x)P(z^+|x^+)P(z^+|z)dz^+dz \; .$$

Proof. All of the above densities are integrable, so we can write the loss function $\mathcal{L}$ in the form of Riemann integrals.

$$\lim_{M\to\infty}\mathcal{L} = -\lim_{M\to\infty}\int P(x)P(x^+|x)\int\prod_{m=1}^{M} P(x_m^-)\log\int Q(z|x)Q(z^+|x^+) \tag{14}$$

$$\prod_{m=1}^{M} Q(z_m^-|x_m^-)\frac{e^{\kappa_{\text{pos}} z^\top z^+}}{\frac{1}{M}e^{\kappa_{\text{pos}} z^\top z^+}+\frac{1}{M}\sum\limits_{m=1}^{M} e^{\kappa_{\text{pos}} z^\top z_m^-}}dz_1^-\dots z_M^- dz^+dzdx_1^-\dots dx_M^-dx^+dx \tag{15}$$

We know that $\kappa_{\text{pos}} < \infty$, $\kappa(x) < \infty \, \forall x \in \mathcal{X}$, the normalization constants $C(\kappa) < \infty \, \forall \kappa < \infty$, and the dot products are bounded. This implies that all densities inside these integrals as well as the exponentials in the fraction are bounded. Thus, the whole term inside the outmost integral is bounded. Due to the dominated convergence theorem we can pull the limit into the integral.

$$= -\int P(x)P(x^+|x)\lim_{M\to\infty}\int\prod_{m=1}^{M} P(x_m^-)\log\int Q(z|x)Q(z^+|x^+) \tag{16}$$

$$\prod_{m=1}^{M} Q(z_m^-|x_m^-)\frac{e^{\kappa_{\text{pos}} z^\top z^+}}{\frac{1}{M}e^{\kappa_{\text{pos}} z^\top z^+}+\frac{1}{M}\sum\limits_{m=1}^{M} e^{\kappa_{\text{pos}} z^\top z_m^-}}dz_1^-\dots z_M^- dz^+dzdx_1^-\dots dx_M^-dx^+dx \tag{17}$$

The strong law of large numbers and the fact that $\int Q(z^-|x^-)P(x^-)dx^- = P(z)$ imply

$$= -\int P(x)P(x^+|x)\lim_{M\to\infty}\log\int Q(z|x)Q(z^+|x^+)\frac{e^{\kappa_{\text{pos}} z^\top z^+}}{\frac{1}{M}e^{\kappa_{\text{pos}} z^\top z^+}+\underset{z^-\sim P(z)}{\mathbb{E}}\left(e^{\kappa_{\text{pos}} z^\top z^-}\right)}dz^+dzdx^+dx \; . \tag{18}$$

Both densities and the fraction inside the inner integral are positive and bounded, so the integral is, too. In this range, i.e., $(0, \infty)$, the logarithm is continuous, so the continuous mapping theorem gives

$$= -\int P(x)P(x^+|x)\log\lim_{M\to\infty}\int Q(z|x)Q(z^+|x^+)\frac{e^{\kappa_{\text{pos}} z^\top z^+}}{\frac{1}{M}e^{\kappa_{\text{pos}} z^\top z^+}+\underset{z^-\sim P(z)}{\mathbb{E}}\left(e^{\kappa_{\text{pos}} z^\top z^-}\right)}dz^+dzdx^+dx \; . \tag{19}$$

With the arguments from above, the inside of the inner integral is bounded, so we can again apply the dominated convergence theorem.

$$= -\int P(x)P(x^+|x)\log\int Q(z|x)Q(z^+|x^+)\lim_{M\to\infty}\frac{e^{\kappa_{\text{pos}} z^\top z^+}}{\frac{1}{M}e^{\kappa_{\text{pos}} z^\top z^+}+\underset{z^-\sim P(z)}{\mathbb{E}}\left(e^{\kappa_{\text{pos}} z^\top z^-}\right)}dz^+dzdx^+dx \tag{20}$$

$$= -\int P(x)P(x^+|x)\log\int Q(z|x)Q(z^+|x^+)\frac{e^{\kappa_{\text{pos}} z^\top z^+}}{\underset{z^-\sim P(z)}{\mathbb{E}}\left(e^{\kappa_{\text{pos}} z^\top z^-}\right)}dz^+dzdx^+dx \tag{21}$$

Since $P(z) = \text{Unif}(\mathcal{S}^{D-1}) = \frac{1}{\|\mathcal{S}^{D-1}\|}$, which we define as $\frac{1}{S}$ in shorthand, we get

$$= -\int P(x)P(x^+|x) \log S \int Q(z|x)Q(z^+|x^+) \frac{e^{\kappa_{\text{pos}} z^\top z^+}}{\int\limits_{\mathcal{S}^{D-1}} e^{\kappa_{\text{pos}} z^\top z^-} dz^-} dz^+ dz dx^+ dx \tag{22}$$

$$= -\int P(x)P(x^+|x) \log S \int Q(z|x)Q(z^+|x^+)P(z^+|z) dz^+ dz dx^+ dx \ . \tag{23}$$

Let us turn our attention to $P(x^+|x)$. By marginalization, factorization, and the conditional independencies of the data-generating process, we get

$$P(x^+|x) \tag{24}$$

$$= \int P(x^+, z^+, z|x) dz^+ dz \tag{25}$$

$$= \int P(x^+|z^+, z, x)P(z^+|z, x)P(z|x) dz^+ dz \tag{26}$$

$$= \int P(x^+|z^+)P(z^+|z)P(z|x) dz^+ dz \ . \tag{27}$$

After a multiplication with 1, Bayes Theorem, and using $P(z) = \frac{1}{S}$, we get

$$= \int \frac{P(x^+|z^+)P(z^+)P(x^+)}{P(z^+)P(x^+)} P(z^+|z)P(z|x) dz^+ dz \tag{28}$$

$$= \int P(z|x)P(z^+|x^+)P(z^+|z) \frac{P(x^+)}{P(z^+)} dz^+ dz \tag{29}$$

$$= P(x^+) \, S \int P(z|x)P(z^+|x^+)P(z^+|z) dz^+ dz \ . \tag{30}$$

We can insert this into Formula 23.

$$-\int P(x)P(x^+) \, S \int P(z|x)P(z^+|x^+)P(z^+|z) dz^+ dz \tag{31}$$

$$\log S \int Q(z|x)Q(z^+|x^+)P(z^+|z) dz^+ dz dx^+ dx \tag{32}$$

$$= \mathop{\mathbb{E}}_{\substack{x \sim P(x) \\ x^+ \sim P(x^+)}} \left(S \int P(z|x)P(z^+|x^+)P(z^+|z) dz^+ dz \log S \int Q(z|x)Q(z^+|x^+)P(z^+|z) dz^+ dz \right) . \tag{33}$$

Note that both terms are conditional on x, x^+ and the expected value is taken over both of these. I.e., $\mathcal{L}$ in the limit is a (non-normalized) cross-entropy between $\int P(z|x)P(z^+|x^+)P(z^+|z)dz^+dz$ and $\int Q(z|x)Q(z^+|x^+)P(z^+|z)dz^+dz$. The loss is minimized iff the two terms match for all values in the outmost expected value, i.e., $\forall x, x^+ \in \{x \in \mathcal{X} | P(x) > 0\}$. □

A.2. Proof of Proposition 4.2

Proposition 4.2 (The marginal is a function) Let $P(z|x)$ and $P(z^+|z)$ be vMF distributions as defined in Section 4.1. Given $x, x^+ \in \mathcal{X}$, we can rewrite

$$\iint P(z|x)P(z^+|x^+)P(z^+|z) dz^+ dz \tag{34}$$

$$=: h_{\kappa_{\text{pos}}}(\mu(x)^\top \mu(x^+), \kappa(x), \kappa(x^+)), \tag{35}$$

i.e., as a function $h_{\kappa_{\text{pos}}}$ that depends only on $\mu(x)^\top \mu(x^+), \kappa(x)$, and $\kappa(x^+)$. The same function can be used for $\hat{\mu}(x)^\top \hat{\mu}(x^+), \hat{\kappa}(x), \hat{\kappa}(x^+)$:

$$\iint Q(z|x)Q(z^+|x^+)P(z^+|z) dz^+ dz \tag{36}$$

$$= h_{\kappa_{\text{pos}}}(\hat{\mu}(x)^\top \hat{\mu}(x^+), \hat{\kappa}(x), \hat{\kappa}(x^+)). \tag{37}$$

Proof. Let us first insert the vMF densities.

$$\iint P(z|x)P(z^+|x^+)P(z^+|z)dz^+dz \tag{38}$$

$$=C(\kappa(x^+))C(\kappa_{\text{pos}}) \iint C(\kappa(x)) \exp[\kappa(x)\mu(x)^\top z + \kappa(x^+)\mu(x^+)^\top z^+ + \kappa_{\text{pos}} z^\top z^+] dz^+dz \tag{39}$$

$$=C(\kappa(x^+))C(\kappa_{\text{pos}}) \int C(\kappa(x)) \exp(\kappa(x)\mu(x)^\top z) \int \exp[(\kappa(x^+)\mu(x^+) + \kappa_{\text{pos}} z)^\top z^+] dz^+dz \tag{40}$$

The term inside the inner integral can be rewritten into an unnormalized vMF density if we specify $\mu^* := \frac{\kappa(x^+)\mu(x^+)+\kappa_{\text{pos}} z}{\|\kappa(x^+)\mu(x^+)+\kappa_{\text{pos}} z\|}$ and $\kappa^* := \|\kappa(x^+)\mu(x^+) + \kappa_{\text{pos}} z\|$. The integral over this density is 1.

$$=C(\kappa(x^+))C(\kappa_{\text{pos}}) \int C(\kappa(x)) \exp(\kappa(x)\mu(x)^\top z) \frac{1}{C(\kappa^*)} \int C(\kappa^*) \exp[\kappa^* {\mu^*}^\top z^+] dz^+dz \tag{41}$$

$$=C(\kappa(x^+))C(\kappa_{\text{pos}}) \int C(\kappa(x)) \exp(\kappa(x)\mu(x)^\top z) \frac{1}{C(\kappa^*)} dz \tag{42}$$

$$=C(\kappa_{\text{pos}}) \mathop{\mathbb{E}}_{z\sim\text{vMF}(\mu(x),\kappa(x))} \left(\frac{C(\kappa(x^+))}{C\left(\sqrt{\kappa(x^+)^2 + \kappa_{\text{pos}}^2 + 2\kappa(x^+)\kappa_{\text{pos}}\mu(x^+)^\top z}\right)} \right) \tag{43}$$

$$=: h_{\kappa_{\text{pos}}}(\mu(x)^\top \mu(x^+), \kappa(x), \kappa(x^+)) \tag{44}$$

In the last step, the expected value is over $\mu(x^+)^\top z, z \sim \text{vMF}(\mu(x), \kappa(x))$. This depends only on the distance $\mu(x)^\top \mu(x^+)$ instead of the full location parameters $\mu(x)$ and $\mu(x^+)$ because the vMF is rotationally symmetric and we can perform a suitable Householder rotation, see also Romanazzi (2014). □

A.3. Proof of Proposition 4.3

Proposition 4.3 (Arguments of h_{pos} must be equal) Define h_{pos} as in Proposition 4.2. Let $\mathcal{X}' \subseteq \mathcal{X}$, $\mu, \hat{\mu} : \mathcal{X}' \to \mathcal{Z}$, $\kappa, \hat{\kappa} : \mathcal{X}' \to \mathbb{R}_{>0}$, $\kappa_{\text{pos}} > 0$. If $h_{\text{pos}}(\hat{\mu}(x)^\top \hat{\mu}(x^+), \hat{\kappa}(x), \hat{\kappa}(x^+)) = h_{\text{pos}}(\mu(x)^\top \mu(x^+), \kappa(x), \kappa(x^+))\ \forall x, x^+ \in \mathcal{X}'$, then

$$\hat{\mu}(x)^\top \hat{\mu}(x^+) = \mu(x)^\top \mu(x^+) \text{ and} \tag{45}$$

$$\hat{\kappa}(x) = \kappa(x)\ \ \forall x, x^+ \in \mathcal{X}'. \tag{46}$$

Proof. (a) The normalization constant of the vMF $C(\kappa) = \frac{\kappa^{D/2-1}}{(2\pi)^{D/2} I_{D/2-1}(\kappa)}$, where I_o is the modified Bessel function of the first kind and order o, is strictly monotonically decreasing and convex (Kirchhof et al., 2022).

(b) Consider arbitrary $x = x^+$, $x \in \mathcal{X}'$. In this case, $\mu(x)^\top \mu(x^+) = \hat{\mu}(x)^\top \hat{\mu}(x^+) = 1$, and both sides of the equality simplify

$$\iint Q(z|x)Q(z^+|x^+)P(z^+|z)dz^+dz = \iint P(z|x)P(z^+|x^+)P(z^+|z)dz^+dz \tag{47}$$

$$\iff h_{\kappa_{\text{pos}}}(1, \kappa(x), \kappa(x)) = h_{\kappa_{\text{pos}}}(1, \hat{\kappa}(x), \hat{\kappa}(x)) \tag{48}$$

$$\iff \tilde{h}_{\kappa_{\text{pos}}}(\kappa(x)) = \tilde{h}_{\kappa_{\text{pos}}}(\hat{\kappa}(x)) \tag{49}$$

with $\tilde{h}_{\kappa_{\text{pos}}}(\kappa) := h_{\kappa_{\text{pos}}}(1, \kappa, \kappa)$. Due to (a), the denominator in Formula 43 grows strictly faster than the numerator. So $\tilde{h}$ is strictly monotonically increasing. Thus, $\tilde{h}_{\kappa_{\text{pos}}}(\kappa(x)) = \tilde{h}_{\kappa_{\text{pos}}}(\hat{\kappa}(x))$ only if $\kappa(x) = \hat{\kappa}(x)$.

(c) Let $x, x^+ \in \mathcal{X}'$ be arbitrary. From (b) we know $\hat{\kappa}(x) = \kappa(x)$, so we can simplify

$$h_{\kappa_{\text{pos}}}(\mu(x)^\top \mu(x^+), \kappa(x), \kappa(x^+)) = h_{\kappa_{\text{pos}}}(\hat{\mu}(x)^\top \hat{\mu}(x^+), \hat{\kappa}(x), \hat{\kappa}(x^+)) \tag{50}$$

$$\iff h^*_{\kappa_{\text{pos}},\kappa(x),\kappa(x^+)}(\mu(x)^\top \mu(x^+)) = h^*_{\kappa_{\text{pos}},\kappa(x),\kappa(x^+)}(\hat{\mu}(x)^\top \hat{\mu}(x^+)) \tag{51}$$

with $h^*_{\kappa_{\text{pos}},\kappa(x),\kappa(x^+)}(\cdot) := h_{\kappa_{\text{pos}}}(\cdot, \kappa(x), \kappa(x^+))$. In other words, both sides of the equality are the same function $h^*_{\kappa_{\text{pos}},\kappa(x),\kappa(x^+)}$ with only one free variable. Due to (a), the denominator in Formula 43 strictly decreases with increasing $\mu(x)^\top \mu(x^+)$ if $\kappa(x^+) > 0$ and $\kappa_{\text{pos}} > 0$. So, $h^*_{\kappa_{\text{pos}},\kappa(x),\kappa(x^+)}$ is strictly monotonically increasing and $h^*_{\kappa_{\text{pos}},\kappa(x),\kappa(x^+)}(\mu(x)^\top \mu(x^+)) = h^*_{\kappa_{\text{pos}},\kappa(x),\kappa(x^+)}(\hat{\mu}(x)^\top \hat{\mu}(x^+))$ implies $\mu(x)^\top \mu(x^+) = \hat{\mu}(x)^\top \hat{\mu}(x^+)$. □

A.4. Proof of Theorem 4.4

Theorem 4.4 ($\mathcal{L}$ identifies the correct posteriors) Let $\mathcal{Z} = \mathcal{S}^{D-1}$ and $P(z) = \int P(z|x)dP(x)$ and $\int Q(z|x)dP(x)$ be the uniform distribution over $\mathcal{Z}$. Let g be a probabilistic generative process defined in Formulas 2, 3, and 4 with known κ_{pos}. Let g have vMF posteriors $P(z|x) = \text{vMF}(z; \mu(x), \kappa(x))$ with $\mu : \mathcal{X} \to \mathcal{S}^{D-1}$ and $\kappa : \mathcal{X} \to \mathbb{R}_{>0}$. Let an encoder $f(x)$ parametrize vMF distributions $\text{vMF}(z; \hat{\mu}(x), \hat{\kappa}(x))$. Then $f^* = \arg\min_f \lim_{M\to\infty} \mathcal{L}$ has the correct posteriors up to a rotation of $\mathcal{Z}$, i.e., $\hat{\mu}(x) = R\mu(x)$ and $\hat{\kappa}(x) = \kappa(x)$, where R is an orthogonal rotation matrix, $\forall x \in \{x \in \mathcal{X} | P(x) > 0\}$.

Proof. If f^* optimizes $\mathcal{L}$, then by Proposition 4.1 $\forall x, x^+ \in \{x \in \mathcal{X}|P(x) > 0\}$ we have

$$\iint Q(z|x)Q(z^+|x^+)P(z^+|z)dz^+dz = \iint P(z|x)P(z^+|x^+)P(z^+|z)dz^+dz \ . \tag{52}$$

Then by Proposition 4.3 with $\mathcal{X}' := \{x \in \mathcal{X}|P(x) > 0\}$ we get $\hat{\kappa}(x) = \kappa(x)$ and $\mu(x)^\top\mu(x^+) = \hat{\mu}(x)^\top\hat{\mu}(x^+)$. With the extended Mazur-Ulam Theorem (Zimmermann et al., 2021), the latter implies $\hat{\mu}(x) = R\mu(x)$ with an orthogonal rotation matrix $R \in \mathbb{R}^{D\times D}$. □

B. Controlled Experiment

B.1. Network Architectures

We use MLPs to parametrize the generative processes' posteriors $\mu(x)$ and $\kappa(x)$ as well as the encoder $\hat{\mu}(x)$ and $\hat{\kappa}(x)$.

For $\mu(x)$ and $\hat{\mu}(x)$ we follow Zimmermann et al. (2021). The MLP for $\mu(x)$ has three linear layers with 10 dimensions and leaky ReLU activations. To prevent collapsed initializations we take 1000 exemplary samples for $\mu(x)$ and re-initiate it if the smallest cosine similarity $x_1^\top x_2$ between any pair x_1, x_2 of them is bigger than 0.5. $\hat{\mu}(x)$ has six hidden linear layers with leaky ReLU activations plus an input and and output layer with the input and output dimensions $[D \to 10{\cdot}D, 10{\cdot}D \to 50{\cdot}D, 50{\cdot}D \to 50{\cdot}D, 50{\cdot}D \to 50{\cdot}D, 50{\cdot}D \to 50{\cdot}D, 50{\cdot}D \to 50{\cdot}D, 50{\cdot}D \to 10{\cdot}D, 10{\cdot}D \to D]$. The outputs of both networks are normalized to an L_2 norm of 1 to ensure they are on the unit sphere.

The MLPs for $\kappa(x)$ and $\hat{\kappa}(x)$ have the same architecture as $\mu(x)$ and $\hat{\mu}(x)$, but $\kappa(x)$ has one less hidden layer than $\mu(x)$. The last layer of both networks outputs only a scalar instead of a D-dimensional vector. It is postprocessed by $\tilde{\kappa}(x) = 1 + \exp(\kappa(x))$ to ensure their strict positivity. Before training, $\hat{\kappa}(x)$ is normalized to output the same range of values as $\kappa(x)$ to improve training stability.

B.2. Generating Contrastive Training Data

The generative process in Section 4.1 first draws latents z and then generates observations x to create contrastive training data. However, we want to control our generative processes' posteriors. Thus, we need to first sample x and then $z \sim P(z|x)$. A method to sample backwards like this while still obtaining samples as if they were from the forward generative process is rejection sampling. We first draw random candidates (x, x^+) from $\mathcal{X} = [0, 1]^D$, then draw (z, z^+) from their corresponding posteriors. To ensure that they form a valid positive example as per the distributions in Formulas 2 and 3, we accept or reject them with a probability proportional to

$$\frac{C(\kappa_{\text{pos}})e^{\kappa_{\text{pos}} z^\top z^+}}{C(\kappa_{\text{pos}})e^{\kappa_{\text{pos}} z^\top z^+} + C(0)} \ . \tag{53}$$

This is the probability that z and z^+ are positive to one another. The proposal distribution's density for rejection sampling is dropped here due to the uniform priors. Negative examples $(x_m^-)_{m=1,\dots,M}$ are drawn randomly from $\mathcal{X}$ due to Formula 4.

B.3. Experiment Parameters

Following Zimmermann et al. (2021), all experiments used $\kappa_{\text{pos}} = 20$ and the above network architectures. The learning rate was 0.0001 and was decreased after each 25% of training progress by a factor of 0.1. Performance was measured at the end of the training without early stopping on 10000 sampled x points. All experiments were implemented in Python 3.8.11, PyTorch 1.9.0 on NVIDIA-RTX 2080TI GPUs with 12GB VRAM. Table 6 below summarizes the remaining parameters used by all ablations of the controlled experiment.

Experiment	Gen. D	Enc. D'	Posterior	$\min(\kappa(x))$	$\max(\kappa(x))$	Batchsize	Number of Batches	Number MC Samples	Comment
Ambiguous ($\kappa(x) \in [16, 32]$)	10	10	vMF	16	32	512	100000	512	Also used for HIB, ELK, InfoNCE
Clear ($\kappa(x) \in [64, 128]$)	10	10	vMF	64	128	512	100000	512	
Injective ($\kappa(x) = \infty$)	10	10	vMF/Dirac	∞	∞	512	100000	512	
$D = 2$	2	2	vMF	16	32	512	8192	512	
Gaussian	10	10	Gaussian	16	32	512	100000	512	$\sigma^2 = 1/\kappa(x)$
Laplace	10	10	Laplace	16	32	512	100000	512	$b = 1/\kappa(x)$
MC Samples	10	10	vMF	16	32	512	100000	x	$x \in \{1, 4, 16, 64, 256, 512\}$
Encoder Dim	10	x	vMF	16	32	512	100000	512	for $x \in \{4, 8, 10, 16, 32\}$
— ” —						512		256	for $x = 64$
— ” —						256		256	for $x = 128$
High Dim	x	x	vMF	16	32	512	100000	512	$x \in \{10, 16\}$
— ” —						256		256	for $x \in \{32, 40, 48, 56, 64\}$

Table 6. Parameters of the generative process and loss in the controlled experiments. x denotes variable parameters. Batchsize and number of MC samples were reduced in high dimensions to not exceed the available VRAM.

B.4. Contrastive Hedged Instance Embeddings

HIB (Oh et al., 2019) is formulated similarly to MCInfoNCE in that it also draws samples of a posterior and computes a probability score with them. HIB originally uses Gaussians and compares L_2 distances between samples. We adapt this to vMFs and cosine distances to align it with the spherical formulation of the latent space. The reformulated HIB loss is

$$\mathcal{L}_{\text{HIB}} := \underset{\substack{x \sim P(x) \\ x^+ \sim P(x^+|x) \\ x_m^- \sim P(x^-), m=1,\dots,M}}{\mathbb{E}} \left(-\log \underset{\substack{z \sim Q(z|x) \\ z^+ \sim Q(z^+|x^+)}}{\mathbb{E}} \big(s(a \cdot z^\top z^+ + b) \big) - \frac{1}{M} \sum_{m=1}^{M} \log \underset{\substack{z \sim Q(z|x) \\ z^+ \sim Q(z^-|x_m^-)}}{\mathbb{E}} \big(1 - s(a \cdot z^\top z_m^- + b) \big) \right), \tag{54}$$

where $s(\cdot)$ is the Sigmoid function and a and b are tuneable hyperparameters. We excluded the KL regularizer originally proposed by Oh et al. since none of the other losses receive prior information on $\kappa(x)$.

B.5. Contrastive Expected Likelihood Kernel

The ELK is commonly used inside a classification cross-entropy loss (Kirchhof et al., 2022). Its key characteristic is that it replaces the point-to-point distance, e.g., cosine distance, by the expected likelihood distance. An analytical solution to compare two vMFs is provided in the supplementary of Kirchhof et al.. We can plug this distance $d_{\text{EL-vMF}}(\hat{\mu}(x_1), \hat{\kappa}(x_1), \hat{\mu}(x_2), \hat{\kappa}(x_2))$ into InfoNCE and transform it into a similarity by multiplying it with -1 to obtain our contrastive ELK loss:

$$\mathcal{L}_{\text{ELK}} := \underset{\substack{x \sim P(x) \\ x^+ \sim P(x^+|x) \\ x_m^- \sim P(x^-), m=1,\dots,M}}{\mathbb{E}} \left(-\log \frac{e^{-\kappa_{\text{pos}} d_{\text{EL-vMF}}(\hat{\mu}(x), \hat{\kappa}(x), \hat{\mu}(x^+), \hat{\kappa}(x^+))}}{\frac{1}{M} e^{-\kappa_{\text{pos}} d_{\text{EL-vMF}}(\hat{\mu}(x), \hat{\kappa}(x), \hat{\mu}(x^+), \hat{\kappa}(x^+))} + \frac{1}{M} \sum_{m=1}^{M} e^{-\kappa_{\text{pos}} d_{\text{EL-vMF}}(\hat{\mu}(x), \hat{\kappa}(x), \hat{\mu}(x_m^-), \hat{\kappa}(x_m^-))}} \right). \tag{55}$$

B.6. Hyperparameter Tuning

All losses were tuned on the ”Standard” experiment setup via grid search. The seed for the generative process was exclusive and not used in the five seeds of the final results. Table 7 below gives the hyperparameters along with the chosen best setup according to the rank correlation between $\kappa(x)$ and $\hat{\kappa}(x)$.

There are two interesting results in this tuning. First, the true generative κ_{pos} was indeed the best choice. All methods performed worse when they learned it themselves (starting from the true value) or when given a different value (not shown here). Second, MCInfoNCE performs best with a high number of negative samples. This corroborates the theoretical study of its limiting behaviour as $M \to \infty$.

Phasewise training is the empirical strategy of first learning $\hat{\mu}(x)$ during the first half of epochs, then fixing it and learning $\hat{\kappa}(x)$ (Shi & Jain, 2019; Li et al., 2021). MCInfoNCE showed an improved performance with this strategy. This is likely because the training signal of $\kappa(x)$ is far lower in the loss than that of $\mu(x)$. During the training phase of $\hat{\mu}(x)$, it turned out beneficial to use negatives from the same batch, i.e., $M = 0$.

	HIB	ELK	MCInfoNCE
Number of negatives M	$\{\mathbf{0}, 1, 32\}$	$\{0, \mathbf{1}, 32\}$	$\{0, 1, \mathbf{32}\}$
κ_{pos} learnable	{yes, **no**}	{yes, **no**}	{yes, **no**}
Phasewise training	{yes, **no**}	{yes, **no**}	{**yes**, no}
a	$\{0.5, \mathbf{1}, 2, 4\}$		
b	$\{-8, -4, -2, -1, \mathbf{0}, 1, 2, 4, 8\}$		

Table 7. Possible hyperparameters and best-performing hyperparameters (**bold**). $M = 0$ corresponds to not sampling negatives, but using one sample from the same batch as a negative. HIB's additional hyperparameters were tuned after the first three parameters to reduce the number of grid-search evaluations.

B.7. Ablation with High Latent Space Dimension

We use the latent space dimension $D = 10$ for most experiments following Zimmermann et al. (2021). Below in Figure 7, we increase the latent space dimension of the generative process and encoder up to 64. We notice considerable performance drops for $D \geq 40$. Other losses than MCInfoNCE also suffer this. Hence, it is likely because of our experimental setup: We use uniformly distributed negatives instead of sophisticated negative mining and the rejection sampling has lower success probabilities in high dimensions, making it harder to generate valid contrastive examples.

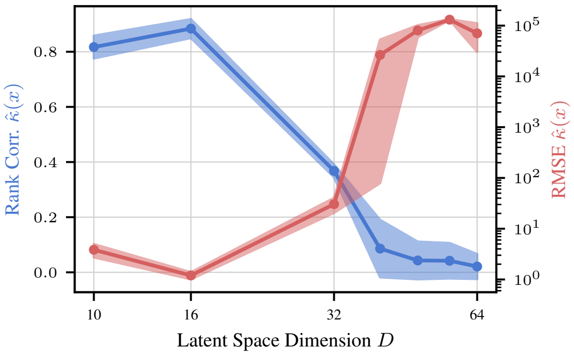


Figure 7. The metrics worsen if the generative process has a latent space of dimension $D \geq 40$. This is likely not due to MCInfoNCE, but a limitation of the contrastive setup of our controlled experiment. Mean ± std. err. for five seeds.

B.8. Ablation with Joint Architecture

In the upper experiments, the networks for $\kappa(x)$ and $\mu(x)$ (and $\hat{\kappa}(x)$ and $\hat{\mu}(x)$) were independent, i.e., did not share parameters. This was to make clear that $\kappa(x)$ characterizes the uncertainty of the input x, rather than the latent of a shared backbone. However, a shared backbone with two heads for $\mu(x)$ and $\kappa(x)$ is a common architecture as, e.g., in VAEs. We've thus run an ablation where $\mu(x)$ is the output of the embedder (a 6-layer MLP) and $\kappa(x)$ is a 3-layer MLP attached after it. This keeps the total number of parameters the same as in the independent case. We rerun the "Ambiguous" setting with $\kappa(x) \in [16, 32]$. Table 8 shows that MCInfoNCE achieves similar performance in both cases.

Table 8. MCInfoNCE also discovers correct posteriors if $\hat{\mu}(x)$ and $\hat{\kappa}(x)$ have a shared backbone. Mean ± std. err. for five seeds.

	True vs Pred. Location $\hat{\mu}(x)$		True vs Pred. Certainty $\hat{\kappa}(x)$	
Architecture	RMSE ↓	Rank Corr. ↑	RMSE ↓	Rank Corr. ↑
Independent Networks	0.04 ± 0.00	0.99 ± 0.00	6.15 ± 0.61	0.82 ± 0.04
Shared Backbone with Two Heads	0.04 ± 0.00	0.99 ± 0.00	7.31 ± 1.53	0.87 ± 0.02

C. CIFAR-10H Experiment

C.1. Contrastive Learning on CIFAR

To test whether the predicted certainty $\hat{\kappa}(x)$ aligns with human-judged aleatoric uncertainty, we require a dataset that provides a ground-truth. CIFAR-10H (Peterson et al., 2019) provides 50 annotations for each test-set image of CIFAR-10. We use the entropy of the probability distribution over these annotations as a measure of aleatoric uncertainty in each image, and compare its negative to the predicted certainty $\hat{\kappa}(x)$ via rank correlation. Since the annotations were only collected for the 10000 images of the test set of CIFAR-10, we apply a 5-fold cross validation. The 10000 images are randomly split into sets of 2000. For five iterations, three of these sets form the train data, one the validation, and one the test data. To prevent confusions with the CIFAR-10 train and test set, we refer to these as the CIFAR-10H train, validation, and test sets. The image indices that belong to each set are provided in our code repository.

This leaves us with the task of redefining the CIFAR classification task into a contrastive learning problem. To this end, we simply assume that images are positive to one another if they belong to the same class and negative if they do not. CIFAR-10H, however, has soft class distributions for each image instead of a crisp class. Thus, we first draw a class c from the class distribution $P(C|x)$ of a reference image x from the train set. We then draw a positive image x^+ from a multinomial distribution over all train images weighed by their probabilities of that class $P(C = c|x^+)$. Negative images x^- are selected the same way, but weighed by the probability of *not* being class c, i.e., $1 - P(C = c|x^-)$. This provides the contrastive data generator required for training.

Since the human annotation data might be noisy in how well it captures the aleatoric uncertainty, we complement it with a synthetical way to introduce aleatoric uncertainty. In a second test dataset, we copy the CIFAR-10H test images, but perform a random crop and rescale that reduces the image to a proportion $\texttt{crop\_size} \sim \text{Unif}([0.25, 1])$ of its original width and length. This directly reduces the information available in the image and therefore increases its aleatoric uncertainty, without introducing artifacts that might let the image go out-of-distribution. We calculate the rank correlation of the reduction in size $\texttt{crop\_size}$ and the (negative) predicted certainty $-\hat{\kappa}(x)$ as an alternative way to evaluate whether $\hat{\kappa}(x)$ reflects loss in information in the input, and therefore aleatoric uncertainty.

C.2. Hyperparameters

We use a ResNet-18 (He et al., 2016) pretrained on the CIFAR-10 train dataset (Phan, 2021) and replace the classification layer by a linear layer with the input and output dimensions $[512, D]$. We then train the linear layer and the ResNet backbone under each loss for 8192 batches of batchsize 128, which corresponds to roughly 175 epochs on the 6000 CIFAR-10H train images. We use the CIFAR-10H validation set to select the best model, evaluated after each 16 batches. The criterion is the rank correlation between $\hat{\kappa}(x)$ and the crop size in the synthetically deteriorated CIFAR-10H validation set. We chose this metric rather than the human annotator disagreement since it can be generated on arbitrary datasets without new annotations. All losses use 128 MC samples and, according to the results in Appendix B.6, a fixed κ_{pos}. We use the same Adam optimizer with a learning rate of 0.0001, learning rate scheduling, and (optional) phase-wise training as in B.6. The remaining hyperparameters were tuned via grid search. The best choices are highlighted in Table 9.

Loss Train Dataset / Label Type	HIB CIFAR-10H soft	ELK CIFAR-10H soft	MCInfoNCE CIFAR-10H soft	MCInfoNCE CIFAR-10H hard	MCInfoNCE CIFAR-10 hard
Latent Dim D	$\{\mathbf{8}, 16\}$	$\{\mathbf{8}, 16\}$	$\{\mathbf{8}, 16\}$	$\{8, \mathbf{16}\}$	$\{\mathbf{8}, 16\}$
Number of negatives M	$\{\mathbf{0}, 1, 32\}$	$\{0, \mathbf{1}, 32\}$	$\{0, 1, \mathbf{32}\}$	$\{\mathbf{0}, 1, 32\}$	$\{\mathbf{0}, 1, 32\}$
κ_{pos}	$\{16, \mathbf{32}, 64\}$	$\{16, \mathbf{32}, 64\}$	$\{\mathbf{16}, 32, 64\}$	$\{\mathbf{16}, 32, 64\}$	$\{16, 32, \mathbf{64}\}$
Phasewise training	{**yes**, no}	{yes, **no**}	{**yes**, no}	{yes, **no**}	{**yes**, no}
a	$\{0.5, 1, \mathbf{2}, 4\}$				
b	$\{-2, -1, 0, \mathbf{1}, 2\}$				

Table 9. Possible hyperparameters and best-performing hyperparameters (**bold**). $M = 0$ corresponds to not sampling negatives, but using one sample from the same batch as a negative. HIB's additional hyperparameters were tuned after the first four parameters to reduce the number of grid-search evaluations.

C.3. Ablation without Pretraining

All experiments on CIFAR started from weights pretrained on CIFAR-10 to reduce the required computational resources. However, it is also an intriguing question if MCInfoNCE is able to train a network from scratch. Table 10 shows that it achieves a similar performance to when it is used on pretrained weights. The small gap in performance may be explained by the fact that we chose the same hyperparameters for both scenarios for fairness. In particular, the learning rate is tuned for the pretrained scenario but not for the non-pretrained one.

Table 10. MCInfoNCE can also be used to train on CIFAR-10H from scratch, without pretraining. Rank correlation on unseen test data.

Pretraining	Annotator Entropy ↑	Crop Size ↑
With Pretraining	0.33	0.70
From Scratch	0.31	0.62

C.4. Uncertainty Estimation is Not At Stakes With First-Moment Estimation

It is a popular question whether uncertainty estimation worsens the general performance, i.e., the estimation of the first-moment embedding $\hat{\mu}(x)$. To add evidence to this discussion, we've implemented the normal InfoNCE loss which estimates only $\hat{\mu}(x)$ but not $\hat{\kappa}(x)$. In both for the CIFAR and controlled experiment. Table 11 shows that MCInfoNCE is not worse than InfoNCE at predicting $\hat{\mu}(x)$. In terms of the RMSE in the controlled experiment, it even outperforms InfoNCE as InfoNCE puts the embeddings too close to one another (RMSE $= 0.83$). This is although InfoNCE was hyperparameter-tuned.

Table 11. MCInfoNCE is not worse than InfoNCE at predicting the first moment of the embedding despite also providing a variance estimate.

Loss	$\mu(x)$ vs $\hat{\mu}(x)$ RMSE ↓	$\mu(x)$ vs $\hat{\mu}(x)$ Rank Corr. ↑	Recall@1 on CIFAR-10H ↑
MCInfoNCE	0.04 ± 0.00	0.99 ± 0.00	0.863
InfoNCE	0.83 ± 0.00	0.99 ± 0.00	0.858

C.5. Credible Intervals

Since we have a (estimated) posterior distribution $P(z|x)$, we can give a credible interval $\text{CI}_p \subseteq \mathcal{Z}$ that the latent z of x falls into with a probability $p \in [0, 1]$, i.e., $P(z \in \text{CI}_p) = p$. We center this interval around the mode of the posterior vMF, such that it is a highest posterior density interval (HPDI). Due to the rotational symmetry of the vMF, for a given $\kappa(x)$ and credible level p, this interval has the form $\text{CI}_p = \{z \in \mathcal{Z} | z^\top \mu(x) \leq t\}$, i.e., all latents z closer to the mode $\mu(x)$ than a certain threshold $t \in [-1, 1]$ measured by cosine similarity. This threshold is the (approximated) $(1 - p)$ quantile of the vMF.

To visualize this latent interval, we define the credible images interval (CII). This is a pre-image of the corresponding CI and gives all images whose mode is within the CI, i.e., $\text{CII}_p := \{x \in \mathcal{X} | \mu(x) \in \text{CI}_p\}$. This can either be visualized via a GAN conditional on $z \in \text{CI}_p$ or by images from the dataset with $\mu(x) \in \text{CII}_p$. We note that this does not reflect the aleatoric uncertainty of those images. We leave this extension for future work.

C.6. Qualitative Evaluation of Aleatoric Uncertainty

Besides the quantitative metrics reported in the main text, we can also take a qualitative look at whether $\hat{\kappa}(x)$ represents aleatoric uncertainty in the inputs. Figure 8 visualizes the five images with the lowest and highest $\hat{\kappa}(x)$ in each class in the CIFAR-10H test set, i.e., on unseen data. It can be seen that images with a low $\hat{\kappa}(x)$ tend to hide characteristic parts of the object via bad crops, being too far away from the object, or an uncommon perspective. Images with a high $\hat{\kappa}(x)$ show characteristic features clearly, making it less ambiguous to tell what they show. In other words, they indeed have a lower aleatoric uncertainty.

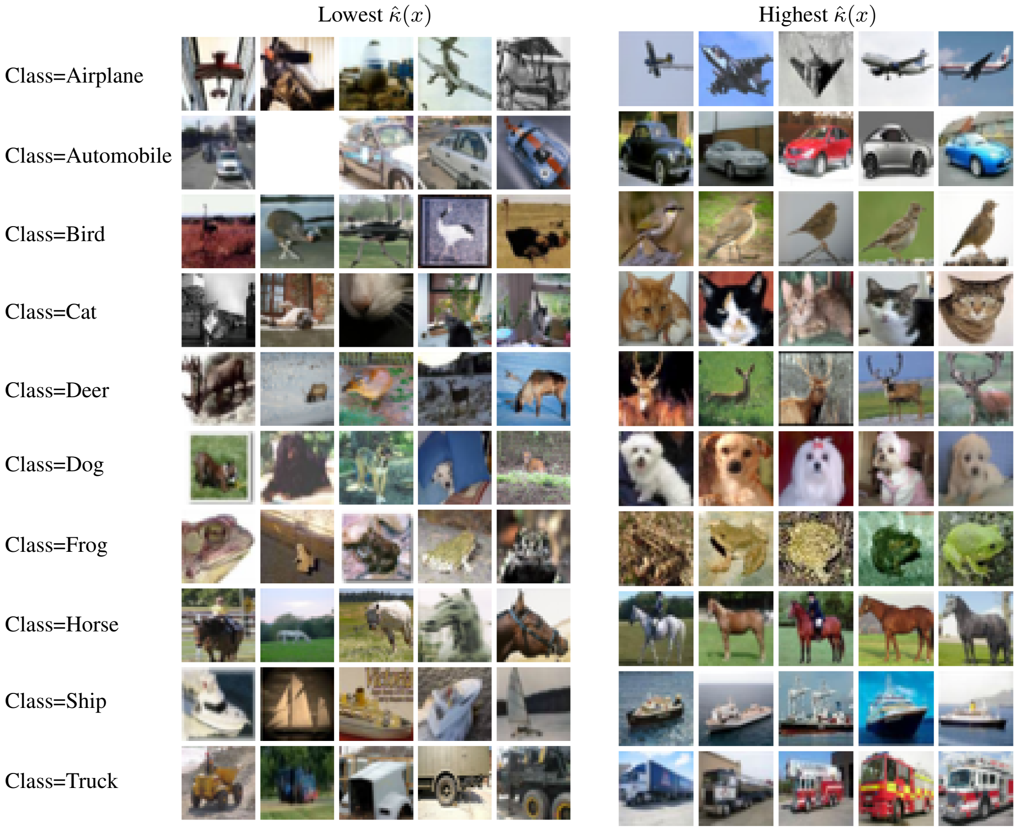


Figure 8. Images for which MCInfoNCE predicts the highest aleatoric uncertainty , i.e., lowest $\hat{\kappa}(x)$, (left) per class qualitatively look more ambiguous than those with the highest predicted $\hat{\kappa}(x)$ (right).

C

URL: A Representation Learning Benchmark for Transferable Uncertainty Estimates

This appendix contains the full paper and appendix discussed in Chapter 4, reproduced with permission.

Michael Kirchhof, Bálint Mucsányi, Seong Joon Oh, and Enkelejda Kasneci. URL: A representation learning benchmark for transferable uncertainty estimates, 2023. First published in: *Neural Information Processing Systems Track on Datasets and Benchmarks (NeurIPS D&B)*, in *Advances in Neural Information Processing Systems 36 pre-proceedings (NeurIPS 2023)*. Available from `https://proceedings.neurips.cc/paper_files/paper/2023/hash/2d421cd0e763f9f01958a30bace955bf-Abstract-Datasets_and_Benchmarks.html`. Reproduced under a Creative Commons Attribution 4.0 International License (`http://creativecommons.org/licenses/by/4.0/legalcode`).

URL: A Representation Learning Benchmark for Transferable Uncertainty Estimates

Michael Kirchhof
University of Tübingen
michael.kirchhof@uni-tuebingen.de

Bálint Mucsányi
University of Tübingen

Seong Joon Oh
University of Tübingen, Tübingen AI Center

Enkelejda Kasneci
TUM University

Abstract

Representation learning has significantly driven the field to develop pretrained models that can act as a valuable starting point when transferring to new datasets. With the rising demand for reliable machine learning and uncertainty quantification, there is a need for pretrained models that not only provide embeddings but also transferable uncertainty estimates. To guide the development of such models, we propose the *Uncertainty-aware Representation Learning* (URL) benchmark. Besides the transferability of the representations, it also meaExamplessures the zero-shot transferability of the uncertainty estimate using a novel metric. We apply URL to evaluate eleven uncertainty quantifiers that are pretrained on ImageNet and transferred to eight downstream datasets. We find that approaches that focus on the uncertainty of the representation itself or estimate the prediction loss directly outperform those that are based on the probabilities of upstream classes. Yet, achieving transferable uncertainty quantification remains an open challenge. Our findings indicate that it is not necessarily in conflict with traditional representation learning goals. Code is available at https://github.com/mkirchhof/url.

1 Introduction

Pretrained models are a vital component of many machine learning applications. The driving force behind their development has been representation learning benchmarks, e.g. Roth et al. (2020); Chen et al. (2020): They task models to output representations $e(x)$ of input data x that generalize across datasets in a zero-shot manner. These pretrained representations provide a valuable starting point for downstream applications, requiring less supervised data to be fine-tuned for specific tasks.

At the same time, uncertainty quantification remains a major challenge in the recent efforts towards reliable machine learning (Collier et al., 2023; Tran et al., 2022). Uncertainty quantification refers to estimating the degree of uncertainty or risk $u(x) \in \mathbb{R}$ in a model's prediction. This is particularly important in high-stakes applications such as medical image classification. Here, the model can refrain from making predictions if the uncertainty, e.g., $u(x) := 1 - \max_y P(Y = y|x)$, is too high (Zou et al., 2023; Bouvier et al., 2022). Beyond classification, uncertainty is an inherent property of vision and language (e.g., low image resolution or ambiguous text inputs) that cannot be learned away even with large amounts of data (Chun et al., 2022; Kendall and Gal, 2017). Consequently, recent literature suggests representing images not as points $e(x)$, but as probabilistic embeddings (Kirchhof et al., 2023; Collier et al., 2023; Chun et al., 2021). Here, $u(x)$ is the variance parameter of a distribution around $e(x)$ in the embedding space, representing the input's inherent ambiguity. This can then be utilized for uncertainty-aware retrieval.

37th Conference on Neural Information Processing Systems (NeurIPS 2023) Track on Datasets and Benchmarks.

A major hurdle on the way to reliable uncertainty estimates is that $u(x)$ needs to be trained from the ground up for each specific task, requiring substantial labeled data. Replicating the successes of representation learning promises to reduce this burden by pretraining a $u(x)$ which can be transferred to downstream tasks in a zero- or finetuned few-shot manner. Yet, this transferability of $u(x)$ to new datasets has not been tested in literature, with previous benchmarks evaluating on the same datasets they trained on (Detommaso et al., 2023; Nado et al., 2021). Thus, we propose a novel *Uncertainty-aware Representation Learning* (URL) benchmark. Models that output both embeddings $e(x)$ and uncertainty estimates $u(x)$ of any form are pretrained on large collections of upstream data and evaluated on unseen downstream datasets. The transferability of their embeddings $e(x)$ is evaluated in terms of the Recall@1 (R@1), as in established representation learning benchmarks (Roth et al., 2020; Chen et al., 2020). The transferability of their uncertainty estimates $u(x)$ is evaluated with a novel metric, the Recall@1 AUROC (R-AUROC). It naturally extends R@1-based benchmarks and can be seamlessly integrated in as little as four lines of code, without requiring any new ground-truth labels. Nonetheless, it is not only an abstract metric but has practical significance: Models with higher R-AUROC are also more aligned with human uncertainties and react better to uncertainty-inducing interventions like image cropping.

On this benchmark, we reimplement and train eleven state-of-the-art uncertainty estimators, from class-entropy baselines over probabilistic embeddings to ensembles, with ResNet (He et al., 2016) and ViT (Dosovitskiy et al., 2021) backbones on ImageNet-1k (Deng et al., 2009). Our main findings are:

1. Transferable uncertainty estimation is an unsolved challenge (Section 4.2),
2. MCInfoNCE and direct loss prediction generalize best (Section 4.3),
3. Uncertainty estimation is not always in conflict with embedding estimation (Section 4.4),
4. Models with good uncertainties upstream are not necessarily good downstream (Section 4.5),
5. URL captures how aligned a model is with human uncertainty (Section 4.6).

These findings demonstrate that pretraining models for downstream uncertainty estimation is an important yet unsolved challenge. We hope that our benchmark will serve as a valuable resource in guiding the field towards pretrained models with reliable and transferable uncertainty estimates.

2 Related work

Our benchmark connects recent uncertainty quantification benchmarks with representation and zero-shot learning for unseen data, which we introduce below. Specific datasets and methods for uncertainty quantification are described in the experiments section when they are benchmarked.

Uncertainty benchmarks. Uncertainty quantification has become an essential consideration for reliable machine learning, and so several libraries have been recently developed to guide its advancement (Detommaso et al., 2023; Nado et al., 2021). These libraries provide various metrics for evaluating and improving uncertainty estimates on in-distribution data. Galil et al. (2023b) and Galil et al. (2023a) benchmarked over 500 large vision models trained on ImageNet from the timm (Wightman, 2019) library and reported that Vision Transformers (ViT) provide the best uncertainty estimates. Further, scaling of these ViTs to up to 22B parameters and pretraining on a large corpus of upstream data results in very accurate uncertainty estimates (Dehghani et al., 2023; Tran et al., 2022). However, when moving away from in-distribution data, the quality of uncertainty estimates deteriorates (Tran et al., 2022) and we can only expect that they will be generally higher and allow for out-of-distribution detection (Ovadia et al., 2019). This motivates our benchmark: We aim to develop pretrained models that can generalize their uncertainty estimates and discriminate certain from uncertain examples even within unseen datasets. While some works applied their uncertainty estimates to unseen datasets (Cui et al., 2023; Collier et al., 2023; Ardeshir and Azizan, 2022; Karpukhin et al., 2022), their downstream evaluations focused on embeddings, leaving the uncertainty estimates untested. Our benchmark intends to bridge this gap and assess the *transferability of uncertainty estimates*, with the goal of enhancing large pretrained models towards zero-shot uncertainty estimation. To design a benchmark for transferability, we connect the upper benchmarking techniques to paradigms from representation and zero-shot learning below.

```python
def url_benchmark(pretrained_model, downstream_loader):
    # Predict embeddings and uncertainties on downstream data
    labels = embeddings = uncertainties = list()
    for (image, label) in downstream_loader:
        embededding, uncertainty = pretrained_model(image)
        (labels, embeddings, uncertainties).append(label, embedding, uncertainty)

    # Calculate R@1 and R-AUROC
    next_neighbor_idx = search_most_similar(embeddings)
    is_same_class = labels == labels[next_neighbor_idx]
    r_at_1 = mean(is_same_class)
    r_auroc = compute_auroc(uncertainties, not is_same_class)

    return r_at_1, r_auroc
```

Algorithm 1: Adding URL to existing representation learning benchmarks takes only the four highlighted lines of code.

Representation and zero-shot learning. Transferability is generally evaluated by testing whether models can make sensible decisions on unseen data. In zero-shot learning (Xian et al., 2017), the model is tasked to give class-predictions on new downstream classes. This requires both learning a transferable representation space on upstream data and creating classifier heads for the new classes from auxiliary information. Representation learning benchmarks (Roth et al., 2020; Khosla et al., 2020; Bengio et al., 2013) focus on the former. To this end, they use a metric similar to class accuracy, the Recall@1 (R@1) (Mikolov et al., 2013). It calculates the model's embeddings of all unseen downstream data and compares whether each embedding's next neighbor is in the same class or not. This tells whether the embeddings are semantically meaningful, such that the pretrained model can be successfully transferred to downstream tasks. We extend representation learning benchmarks to additionally judge the transferability of uncertainty estimates. To this end, we propose a metric that can be implemented on top of the R@1 in four lines of code in the next section.

3 Uncertainty-aware representation learning (URL) benchmark

3.1 Evaluating uncertainty about representations

To quantify its uncertainty, a model $f : \mathcal{X} \to \mathcal{E} \times \mathcal{U}$ is assumed to predict both an embedding $e(x) \in \mathcal{E}$ and a scalar uncertainty value $u(x) \in \mathcal{U} \subset \mathbb{R}$ for each input image $x \in \mathcal{X}$. We do not impose restrictions on how $u(x)$ is calculated, e.g., it could be the negative maximum probability of a softmax classifier, a predicted variance from a dedicated uncertainty module, or the disagreement between ensemble members. The predicted uncertainty $u(x)$ is commonly benchmarked in terms of its expected calibration error (ECE), negative log-likelihood, area under the receiver-operator characteristics curve (AUROC), or abstained prediction curves. All of these measures are w.r.t. the correctness of a classification decision. Hence, $u(x)$ can only be evaluated in-distribution with known classes. Our setup involves unseen datasets and classes, so we need to develop a fitting measure.

To this end, let us take Lahlou et al. (2023)'s decision-theoretic perspective on uncertainty quantification: Uncertainty quantification is loss prediction. The uncertainty expresses the expected loss of a model's decision on a specific datapoint. In Gaussian regression with an L_2 loss, the expected loss is the target's variance, so an uncertainty quantifier $u(x)$ should be proportional to it. In classification with a 0-1 loss, $u(x)$ should be proportional to the probability of returning the correct class.

In representation learning, the model's decision is the embedding $e(x)$ and the loss is the R@1. The uncertainty quantifier's goal is then to report the loss attached to the embedding, i.e., $u(x)$ should be proportional to whether the R@1 will be correct or not. This demonstrates the use case of $u(x)$: Telling whether an embedding $e(x)$ can be trusted or could be misplaced in the embedding space. This is an important property as models of the form $x \to e \to y$ have an information bottleneck in the quality of the embedding $e(x)$ due to the data-processing inequality (Cover, 1999). For every downstream task, a higher uncertainty $u(x)$ about $e(x)$ monotonically increases the loss of $y(e(x))$. In other words, if the embedding is wrong, then the prediction in any downstream task will be wrong.

We measure whether the uncertainty quantifier $u(x)$ is proportional to the correctness of the embedding $e(x)$ via the AUROC with respect to whether the R@1 is correct (one) or not (zero), named R-AUROC. As the R@1 is a 0-1 loss, the R-AUROC can be interpreted as the probability that an incorrect embedding will receive a higher uncertainty score than a correct embedding (Fawcett, 2006). An R-AUROC close to 1 means that $u(x)$ clearly separates correct from incorrect embeddings, while an R-AUROC of 0.5 means that it has no more predictive power than a random guess.

A positive trait of the R-AUROC is that it is indicative of how well-aligned the model is with human uncertainties and how well the model reacts to uncertainty-inducing interventions (see Fig. 5 and Section 4.6). It also does not require uncertainty ground-truths and takes only four lines of code to implement into existing representation learning benchmarks, as shown in Algorithm 1. We choose the AUROC over other calibration measures such as the ECE because it accepts uncertainties $u(x) \in \mathbb{R}$ (instead of $u(x) \in [0, 1]$) and because it avoids some loopholes of the ECE. We discuss these and more design choices behind this metric in Appendix A.

3.2 URL benchmark protocol

The R-AUROC can be evaluated on any downstream dataset that allows calculating the R@1, i.e., has class labels. Yet, in order to keep future results comparable, we propose a benchmark protocol for uncertainty-aware representation learning (URL). Our code is based on `timm` (Wightman, 2019) and available at `https://github.com/mkirchhof/url`.

Datasets. We train each model on ImageNet-1k (Deng et al., 2009) as upstream dataset. We note that future works may use larger-scale upstream datasets (Collier et al., 2023; Tran et al., 2022) or auxiliary information (Han et al., 2023; Ortiz-Jimenez et al., 2023), as long as they stay disjoint to the downstream datasets. As downstream datasets, we follow the standardized representation learning protocol of Roth et al. (2020) and use CUB-200-2011 (Wah et al., 2011), CARS196 (Krause et al., 2013), and Stanford Online Products (Song et al., 2016). We follow the original splits that divide their classes into equally sized train and test sets. Following Roth et al. (2020), we further divide the classes in the train set equally into a train and a validation split. In our zero-shot transfering setup, all models are trained only on the upstream ImageNet dataset and do not use the downstream train splits. All results report the performance on the test sets, averaged across the three datasets and three seeds.

Hyperparameters. We use the downstream validation split to select the best learning rate, early stopping, and further hyperparameters of each model individually, see also Appendix B. Each model is tuned for 10 search iterations via Bayesian Active Learning (Biewald, 2020). The best model is chosen based on the R-AUROC on validation data, where models with an R@1 below 0.1 on the validation splits are filtered out. The best model is replicated on three seeds.

Architectures. Following uncertainty quantification and representation learning benchmarks (Wen et al., 2021; Dusenberry et al., 2020; Roth et al., 2020), we use a ResNet-50 (He et al., 2016) with an embedding space dimension of 2048 as a backbone. We further study ViT-Medium (Dosovitskiy et al., 2021) backbones due to their performance (Galil et al., 2023a,b) and increasing number of large-scale uncertainty quantifiers built on top of them (Collier et al., 2023; Tran et al., 2022). Methods that predict $u(x)$ with explicit modules use a 3-layer MLP head attached to the embeddings.

Training infrastructure. Each model is trained with an aggregated batch size of 2048, as recent studies indicate higher batch sizes might benefit uncertainty quantification (Galil et al., 2023b). We use the Lamb optimizer (You et al., 2020) with cosine annealing learning rate scheduling (Loshchilov and Hutter, 2017) for all models since it performed best in preliminary experiments. The ResNets and ViTs are trained on NVIDIA RTX 2080 Ti and A100 GPUs, respectively, for 32 epochs from a checkpoint pretrained on ImageNet to reduce the computational costs. In total, the experiments took 3.2 GPU years of runtime.

Further metrics. Uncertainty estimates aim to assess the errors made by individual models, so that they are necessarily model- and performance-dependent. To provide a comprehensive view, we not only evaluate the quality of the uncertainty estimate using R-AUROC but also consider the model's representation learning performance using R@1.

4 Experiments

4.1 Uncertainty estimators

We apply URL to benchmark two baselines (**CE**, **InfoNCE**), five probabilistic embeddings approaches (**MCInfoNCE**, **ELK**, **nivMF**, **HIB**, **HET-XL**), two direct variance models (**Losspred**, **SNGP**), and two ensembles (**Ensemble**, **MCDropout**). We introduce each approach below and explain further details on their reimplementations and hyperparameters in Appendix B and runtimes in Appendix C.7.

Cross Entropy (CE) is a supervised baseline which trains under a cross-entropy loss. It uses the entropy of the upstream class probabilities $u(x) := \mathcal{H}(P(Y|x))$ as uncertainty estimate.

InfoNCE (Oord et al., 2018) is an unsupervised baseline. Following SIMCLR (Chen et al., 2020), it takes two random transforms of each image and pulls their embeddings towards each other and repels them from the remaining batch. **InfoNCE** itself does not estimate $u(x)$, so we use the embedding norm $u(x) := \|e(x)\|_2$ as a heuristic (Kirchhof et al., 2022; Scott et al., 2021; Li et al., 2021).

MCInfoNCE (Kirchhof et al., 2023) follows the unsupervised setup of **InfoNCE**, but predicts a certainty $\kappa(x) =: 1/u(x)$ along with each embedding to define a distribution in the embedding space, so called probabilistic embeddings. It draws samples from them and applies **InfoNCE** on each.

Expected Likelihood Kernel (ELK) (Kirchhof et al., 2022; Shi and Jain, 2019) also predicts certainties $\kappa(x) =: 1/u(x)$ to define probabilistic embeddings. The probabilistic embeddings are compared to class distribution via an expected likelihood distribution-to-distribution kernel (Jebara and Kondor, 2003). This makes it a supervised probabilistic embedding-based loss.

Non-isotropic von Mises-Fisher (nivMF) (Kirchhof et al., 2022) is analoguous to **ELK**, but models class distributions as non-isotropic von Mises-Fisher distributions, thereby allowing different variances along each embedding space axis. Image certainties are still scalars $\kappa(x) =: 1/u(x)$.

Hedged Instance Embeddings (HIB) (Oh et al., 2019) predicts variances $\sigma(x) =: u(x)$ of probabilistic embeddings. Samples are drawn to compute match probabilities between two images. It aims to increase the match probabilities of same-class pairs and decrease that of different-class ones.

Heteroscedastic Classifier (HET-XL) (Collier et al., 2023) differs from the above probabilistic embeddings approaches in that it predicts full covariance matrices $\Sigma(x)$ in the embedding space. It draws samples from these probabilistic embeddings to calculate the expected $P(Y|x)$. We test both $u(x) := \det \Sigma(x)$ and the class entropy $u(x) := \mathcal{H}(P(Y|x))$ as possible uncertainty estimates.

Spectral-normalized Neural Gaussian Processes (SNGP) (Liu et al., 2020) model class logits as Gaussian Processes with a predicted mean and a heteroscedastic variance. They are pooled into class probabilities $P(Y|x)$ and trained under a **CE** loss. The entropy of these probabilities serves as uncertainty value $u(x) := \mathcal{H}(P(Y|x))$.

Loss Prediction (Losspred) approaches (Upadhyay et al., 2023; Lahlou et al., 2023; Levi et al., 2022; Laves et al., 2020; Yoo and Kweon, 2019) in regression treat uncertainty quantification as secondary regression task. We apply the same principle to classification, where we task the uncertainty module $u(x)$ to predict the (gradient-detached) **CE** loss at each sample via an L_2 loss added to the train loss.

Deep Ensembles (Lakshminarayanan et al., 2017) train multiple randomly initiated networks under a **CE** loss to obtain several predictions. They are pooled one class distribution $P(Y|x)$. We define the uncertainty either via its entropy $u(x) := \mathcal{H}(P(Y|x))$ or the Jensen–Shannon divergence between the ensemble members' class probability distributions. Following (Lee et al., 2015), we only train multiple output heads and share the backbone to reduce computational complexity.

MCDropout (Gal and Ghahramani, 2016) applies Dropout (Srivastava et al., 2014) at inference time. This gives multiple predictions per input, imitating the upper **Ensemble**. We use both the entropy $u(x) := \mathcal{H}(P(Y|x))$ and the Jensen–Shannon divergence between the ensemble members' class probability distributions as possible uncertainty metrics.

4.2 Transferable uncertainty estimation is an unsolved challenge

Fig. 1 presents the URL benchmark results, i.e., the R-AUROC calculated for all above approaches on ResNet and ViT backbones. The barplot shows the minimum, average, and maximum performance across three seeds of each hyperparameter-tuned approach.

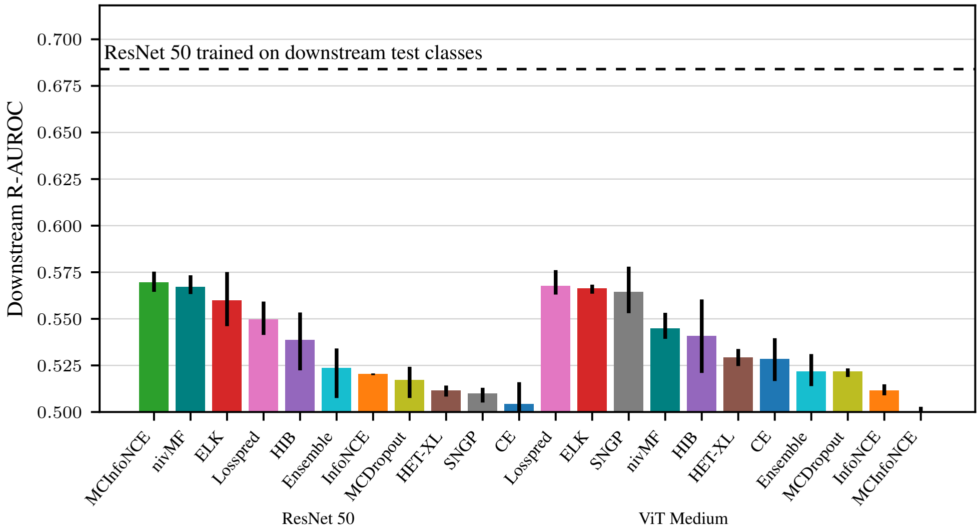


Figure 1: Zero-shot uncertainty estimates of pretrained models (bars) do not reach the performance of many-shot models yet (dashed line). The URL benchmark aims to guide the field to close this gap. Minimum, average, and maximum R-AUROC across three seeds.

Before comparing the models, we first investigate whether the transferability problem URL addresses is already solved by any of the existing methods. To obtain an upper reference, we additionally train a ResNet 50 with standard cross-entropy loss and entropy of the class probabilities as uncertainty prediction on the downstream test classes (split into a train and test split for this experiment only). This many-shot performance of 0.68 is not reached by any of the methods that transfer their uncertainty in a zero-shot way, marking URL as an open challenge. In the standard R@1, this gap has already been closed in representation learning (see Appendix C.1) and we hope that URL guides the field towards the same for transferable uncertainty estimation.

4.3 MCInfoNCE and direct loss prediction generalize best

To compare the approaches in detail, in addition to Fig. 1, Fig. 2 reports both the downstream uncertainty and R@1 performance. The overall best method is **Losspred**, with the second-best average R-AUROC of 0.568, close to the best method, **MCInfoNCE**, with an average R-AUROC of 0.569, while maintaining the second-best average R@1 of 0.53, close to the best R@1 of 0.57 achieved by **nivMF**. **MCInfoNCE** marks the best performance in both metrics within the ResNet models, closely followed by **nivMF**. This is remarkable as it is the only unsupervised method aside from the **InfoNCE** baseline. One final noteworthy mention is **ELK** which provides decent uncertainty estimates on both ResNets and ViTs, whereas most other models vary in their performance depending on the backbone.

When grouping the approaches, those that directly model the variance (**Losspred**, **SNGP**) appear to have an edge on the ViTs, especially **Losspred**, which is the only method that disentangles variance estimation from how the class logits are calculated. Such disentanglement via having two losses could be added to other approaches in future works. Probabilistic embeddings, especially **MCInfoNCE**, **nivMF** and **ELK**, also show promising performance both on the bigger ViTs and the smaller ResNets. Ensembles fail to provide transferable uncertainty estimates. The baselines unsurprisingly fail, indicating that well-calibrated class probabilities on the upstream dataset do not serve as good uncertainties on downstream data. We investigate this further in Section 4.5.

4.4 Uncertainty estimation is not always in conflict with embedding estimation

A commonly raised concern is whether or not uncertainty quantification deteriorates the prediction, or, in the representation learning setup, the embedding quality. In the previous section, we have

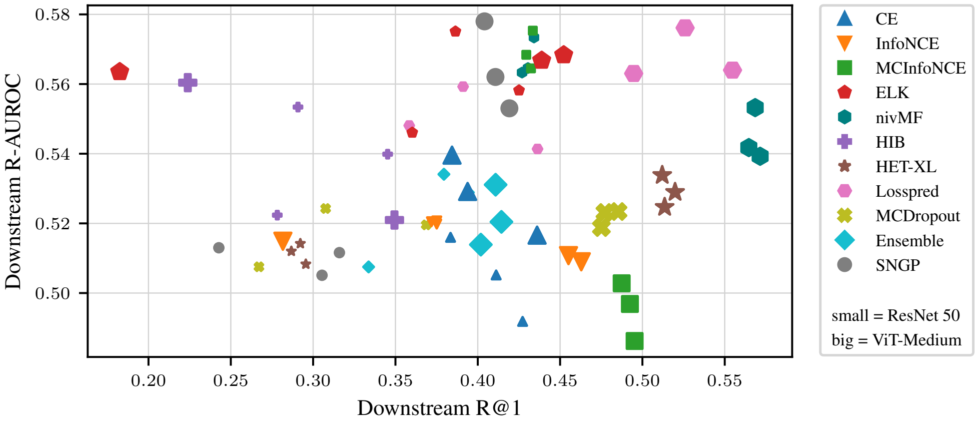


Figure 2: Among ViTs and ResNets, respectively, **Losspred** and **MCInfoNCE** transfer best both in terms of uncertainty estimates (y-axis), measured by our R-AUROC, and embedding quality (x-axis), measured by Recall@1. Three seeds per model and architecture.

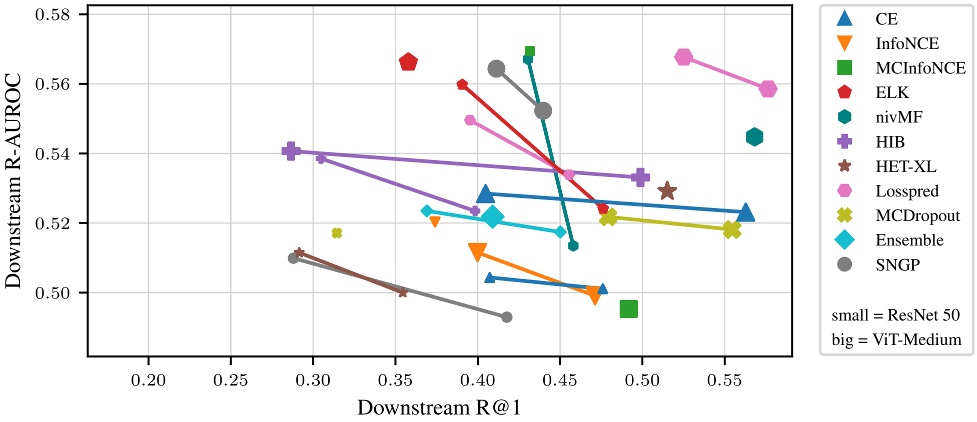


Figure 3: Best hyperparameters chosen for R@1 and for R-AUROC for each model. For some models, there is one best hyperparameter for both, resulting in a point, but most have a large trade-off. Average performance across three seeds.

already seen that **Losspred** can achieve both with only a slight trade-off to the best method in each category. In this section, we further detail this question within each model class.

Fig. 3 shows the performance of the best hyperparameters chosen according to R-AUROC or according to the R@1. If there was no trade-off, the points would lay at the same position or only have a short line connecting them. This is the case for **MCInfoNCE**, **ELK**, **nivMF**, **HET-XL**, and **Ensemble** on ViTs and **MCInfoNCE**, **MCDropout**, and **InfoNCE** on ResNets. The remaining 14 of the 22 approaches show large tradeoffs, e.g., -0.21 R@1 for $+0.01$ R-AUROC for **HIB** on ViTs. Whereas this comparison regards only the two extreme ends of the spectrum, Appendix C.2 measures the rank correlation across all tested hyperparameters. It shows a similar conlusion, with 15 out of 22 approaches trading off uncertainty and prediction. However, from another perspective, **Losspred**, **nivMF**, and **MCInfoNCE** are model classes that provide good performance in both simultaneously. Hence, the question of whether there is a general trade-off between uncertainty estimation and prediction is still up to debate. Studying these models and mitigating the model-internal trade-offs is an interesting future work that we hope URL can enable.

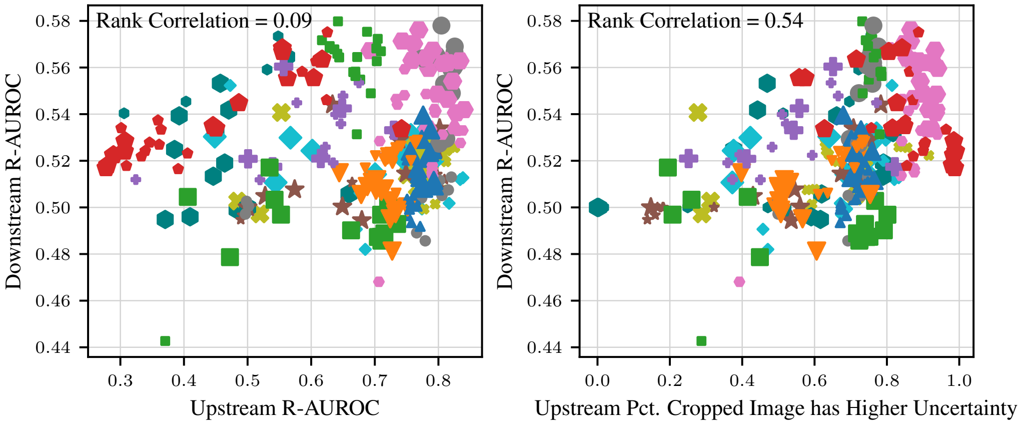


Figure 4: The R-AUROC on upstream data does not indicate the performance on downstream data (left). Yet, the percentage of upstream images where a cropped version receives a higher uncertainty is indicative (right). Plots show all hyperparameters, including non-optimal ones.

4.5 Models with good uncertainties upstream are not necessarily good downstream

We have seen that CE is unable to transfer its well-calibrated upstream uncertainty estimates to downstream datasets. This brings up the question of how much the upstream and downstream uncertainty quantification abilities coincide in general. Fig. 4 (left) shows that the majority of models achieves an R-AUROC above 0.7 on the upstream seen classes. But this does not indicate a good downstream performance (Rank Corr. 0.09), neither across nor within model classes (unlike up- and downstream R@1, which transfers better, see Appendix C.3). This demonstrates that transferable uncertainty quantification will not solve itself by merely becoming better on the upstream data.

This also opens a question about model choice: If the upstream performance cannot tell how well the model's uncertainty predictions will perform downstream, how should we select pretrained models? In this paper, we used downstream validation data. However, if we are limited to upstream data, we may test the uncertainty module in a more general task that also holds downstream. In Fig. 4 (right), we calculate how often the model assigns a higher uncertainty value to a cropped version of an image than to the original image. The rank correlation of 0.54 with the downstream R-AUROC signals that models that perform well on this general uncertainty task also tend to generalize better to the downstream data. This means that general uncertainty tasks might be good heuristics to choose models, reinforcing practices in recent literature (Kirchhof et al., 2023).

4.6 URL captures how well-aligned a model is with human uncertainty

While the R-AUROC is simple and theoretically founded, readers might still wonder why we want to drive the development of models based on this rather technical-seeming metric. In this section, we show that the R-AUROC reflects how well-aligned the model is with human uncertainties.

To verify this, we use five additional downstream datasets from Schmarje et al. (2022): CIFAR-10H (Peterson et al., 2019), Benthic (Langenkämper et al., 2020; Schoening et al., 2020), Pig (Schmarje et al., 2022), Turkey (Volkmann et al., 2022, 2021), and Treeversity#1 (Arnold Arboretum, 2020). They present human annotators with naturally ambiguous images and record their uncertainty by collecting multiple class annotations per image. The entropy of this distribution measures the human uncertainty $h(x) = \mathcal{H}(P_{\text{human}}(Y|x))$. We can then measure the alignment of the model f's uncertainties with human uncertainties via rank correlation $a(f) = \text{Rank Corr.}(\{u(x), h(x)\}_x)$. Fig. 5 (left) shows that this alignment metric $a(f)$ is positively correlated with the R-AUROC (Rank Corr. 0.80). Further, Fig. 5 (right) shows that the same holds for the correlation between R-AUROC and how well a model detects the uncertainty introduced synthetically via cropping (Rank Corr. 0.71), as in the previous section. This means that the R-AUROC is not just a technical metric, but reveals

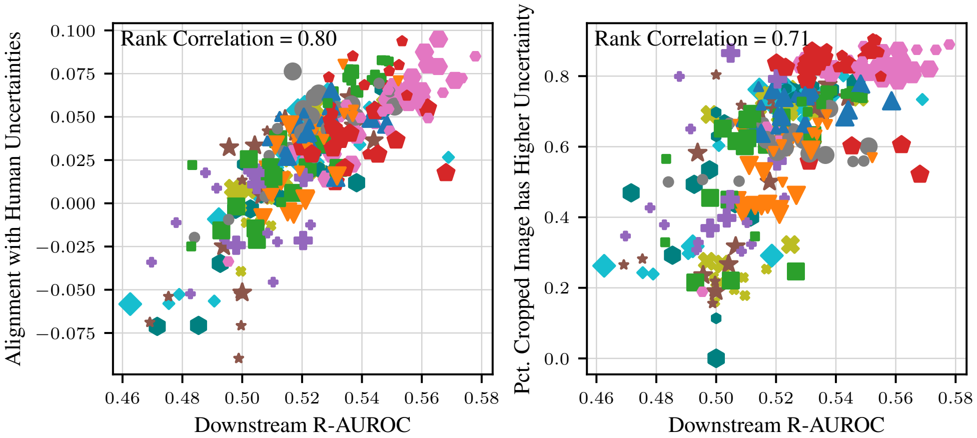


Figure 5: If a model has a high R-AUROC, it is likely also well-aligned with human uncertainties (left). Further, it is likely able to detect uncertainties induced via synthetical cropping (right). All results on five further downstream datasets from Schmarje et al. (2022).

how well a model's uncertainty estimate is aligned with human and synthetical notions of uncertainty, despite not requiring access to human uncertainty ground truths.

4.7 URL is no out-of-distribution detection benchmark

Last, we want to clarify how URL is different from out-of-distribution (OOD) detection benchmarks like (Ovadia et al., 2019). While both test uncertainty estimates on OOD data, the goal is different: In OOD detection benchmarks, the uncertainty estimates are tasked to be generally higher for OOD than for in-distribution (ID) samples. In URL, we look only at the OOD data and see if the uncertainties within this data are correctly sorted. This is because our use-case is not to build OOD or anomaly detectors, but pretrained models whose uncertainty estimates generalize to new datasets. In Appendices C.4 and C.5 we show that methods with good OOD detection abilities are not necessarily good in URL or vice versa. This demonstrates that URL is concerned with predictive uncertainty estimation (and generalization), which is largely driven by aleatoric uncertainty, rather than epistemic uncertainty estimation, which is tested in OOD benchmarks.

5 Conclusion

Summary This paper proposes the uncertainty-aware representation learning (URL) benchmark. On top of the Recall@1, URL adds an easy-to-implement metric that evaluates how well models estimate uncertainties on unseen downstream data. Besides having a theoretical foundation, it also behaves similarly to practical metrics like the alignment with human uncertainties. In benchmarking eleven state-of-the-art approaches on ResNet and ViT backbones, we found that the challenge URL poses is far from being solved. We hope that URL guides the field to overcome this challenge and yield models with reliable pretrained uncertainty estimates.

Outlook We gathered some insights that might guide future developments: Both unsupervised and supervised methods can learn transferable uncertainty estimates. This is not necessarily at stakes with the embedding and prediction quality. However, many methods have internal trade-offs in their hyperparameters. A deeper analysis of the reasons for this trade-off could allow us to control and mitigate it. Loss prediction and probabilistic embedding methods are currently the most promising approaches. They may be combined to enhance each other and define a new state-of-the-art.

Limitations Although URL allows using any upstream benchmark, we have focused on ImageNet-1k to train all current methods on the same ground. We leave the investigation of further scaled datasets to forthcoming research. Further, we hyperparameter-tuned each model individually with the same budget, but the vast number of hyperparameters in some models, like SNGP, means that our active learning search may have missed some fruitful combinations. Finally, our study concentrates on zero-shot uncertainty estimates. It will be an interesting endeavour to see if pretrained models with good zero-shot estimates also accelerate learning uncertainties in few-shot scenarios.

Acknowledgements

This work was funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) under Germany's Excellence Strategy – EXC number 2064/1 – Project number 390727645. The authors thank the International Max Planck Research School for Intelligent Systems (IMPRS-IS) for supporting Michael Kirchhof.

References

Shervin Ardeshir and Navid Azizan. Uncertainty in contrastive learning: On the predictability of downstream performance. *arXiv preprint arXiv:2207.09336*, 2022.

Arnold Arboretum. The TreeVersity dataset, 2020. URL https://arboretum.harvard.edu/research/data-resources/.

Yoshua Bengio, Aaron Courville, and Pascal Vincent. Representation learning: A review and new perspectives. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(8):1798–1828, 2013.

Lukas Biewald. Experiment tracking with weights and biases, 2020. URL https://www.wandb.com/.

Victor Bouvier, Simona Maggio, Alexandre Abraham, and Léo Dreyfus-Schmidt. Towards clear expectations for uncertainty estimation. *arXiv preprint arXiv:2207.13341*, 2022.

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *Proceedings of the 37th International Conference on Machine Learning (ICML)*, 2020.

Sanghyuk Chun, Seong Joon Oh, Rafael Sampaio De Rezende, Yannis Kalantidis, and Diane Larlus. Probabilistic embeddings for cross-modal retrieval. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021.

Sanghyuk Chun, Wonjae Kim, Song Park, Minsuk Chang Chang, and Seong Joon Oh. ECCV caption: Correcting false negatives by collecting machine-and-human-verified image-caption associations for ms-coco. In *European Conference on Computer Vision (ECCV)*, 2022.

Mark Collier, Rodolphe Jenatton, Basil Mustafa, Neil Houlsby, Jesse Berent, and Effrosyni Kokiopoulou. Massively scaling heteroscedastic classifiers. *arXiv preprint arXiv:2301.12860*, 2023.

Thomas M Cover. *Elements of information theory*. John Wiley & Sons, 1999.

Peng Cui, Dan Zhang, Zhijie Deng, Yinpeng Dong, and Jun Zhu. Learning sample difficulty from pre-trained models for reliable prediction. *arXiv preprint arXiv:2304.10127*, 2023.

Tim R. Davidson, Luca Falorsi, Nicola De Cao, Thomas Kipf, and Jakub M. Tomczak. Hyperspherical variational auto-encoders. *34th Conference on Uncertainty in Artificial Intelligence (UAI)*, 2018.

Mostafa Dehghani, Josip Djolonga, Basil Mustafa, Piotr Padlewski, Jonathan Heek, Justin Gilmer, Andreas Steiner, Mathilde Caron, Robert Geirhos, Ibrahim Alabdulmohsin, et al. Scaling vision transformers to 22 billion parameters. *arXiv preprint arXiv:2302.05442*, 2023.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *Computer Vision and Pattern Recognition (CVPR)*, 2009.

Nicki Skafte Detlefsen, Jiri Borovec, Justus Schock, Ananya Harsh Jha, Teddy Koker, Luca Di Liello, Daniel Stancl, Changsheng Quan, Maxim Grechkin, and William Falcon. Torchmetrics - measuring reproducibility in pytorch. *Journal of Open Source Software*, 7(70):4101, 2022.

Gianluca Detommaso, Alberto Gasparin, Michele Donini, Matthias Seeger, Andrew Gordon Wilson, and Cedric Archambeau. Fortuna: A library for uncertainty quantification in deep learning. *arXiv preprint arXiv:2302.04019*, 2023.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations (ICLR)*, 2021.

Michael Dusenberry, Ghassen Jerfel, Yeming Wen, Yian Ma, Jasper Snoek, Katherine Heller, Balaji Lakshminarayanan, and Dustin Tran. Efficient and scalable Bayesian neural nets with rank-1 factors. In *International conference on machine learning (ICML)*, 2020.

Tom Fawcett. An introduction to ROC analysis. *Pattern recognition letters*, 27(8):861–874, 2006.

Yarin Gal and Zoubin Ghahramani. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In *International Conference on Machine Learning (ICML)*, 2016.

Ido Galil, Mohammed Dabbah, and Ran El-Yaniv. A framework for benchmarking class-out-of-distribution detection and its application to imagenet. In *The Eleventh International Conference on Learning Representations (ICLR)*, 2023a.

Ido Galil, Mohammed Dabbah, and Ran El-Yaniv. What can we learn from the selective prediction and uncertainty estimation performance of 523 imagenet classifiers? In *International Conference on Learning Representations (ICLR)*, 2023b.

Dongyoon Han, Junsuk Choe, Seonghyeok Chun, John Joon Young Chung, Minsuk Chang, Sangdoo Yun, Jean Y. Song, and Seong Joon Oh. Neglected free lunch – learning image classifiers using annotation byproducts. *arXiv preprint arXiv:2303.17595*, 2023.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Computer Vision and Pattern Recognition (CVPR)*, 2016.

Tony Jebara and Risi Kondor. Bhattacharyya and expected likelihood kernels. In *Learning Theory and Kernel Machines*. 2003.

Ivan Karpukhin, Stanislav Dereka, and Sergey Kolesnikov. Probabilistic embeddings revisited. *arXiv preprint arXiv:2202.06768*, 2022.

Alex Kendall and Yarin Gal. What uncertainties do we need in bayesian deep learning for computer vision? *Advances in Neural Information Processing Systems (NeurIPS)*, 30, 2017.

Prannay Khosla, Piotr Teterwak, Chen Wang, Aaron Sarna, Yonglong Tian, Phillip Isola, Aaron Maschinot, Ce Liu, and Dilip Krishnan. Supervised contrastive learning. *Advances in Neural Information Processing Systems (NeurIPS)*, 33:18661–18673, 2020.

Michael Kirchhof, Karsten Roth, Zeynep Akata, and Enkelejda Kasneci. A non-isotropic probabilistic take on proxy-based deep metric learning. In *European Conference on Computer Vision (ECCV)*, 2022.

Michael Kirchhof, Enkelejda Kasneci, and Seong Joon Oh. Probabilistic contrastive learning recovers the correct aleatoric uncertainty of ambiguous inputs. *International Conference on Machine Learning (ICML)*, 2023.

Jonathan Krause, Michael Stark, Jia Deng, and Li Fei-Fei. 3D object representations for fine-grained categorization. In *Conference on Computer Vision and Pattern Recognition (CVPR) Workshop*, 2013.

Salem Lahlou, Moksh Jain, Hadi Nekoei, Victor I Butoi, Paul Bertin, Jarrid Rector-Brooks, Maksym Korablyov, and Yoshua Bengio. DEUP: Direct epistemic uncertainty prediction. *Transactions on Machine Learning Research (TMLR)*, 2023. ISSN 2835-8856.

Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. Simple and scalable predictive uncertainty estimation using deep ensembles. *Advances in Neural Information Processing Systems (NeurIPS)*, 2017.

Daniel Langenkämper, Robin Van Kevelaer, Autun Purser, and Tim W Nattkemper. Gear-induced concept drift in marine images and its effect on deep learning classification. *Frontiers in Marine Science*, 7:506, 2020.

Max-Heinrich Laves, Sontje Ihler, Jacob F Fast, Lüder A Kahrs, and Tobias Ortmaier. Well-calibrated regression uncertainty in medical imaging with deep learning. In *Medical Imaging with Deep Learning*, pages 393–412. PMLR, 2020.

Stefan Lee, Senthil Purushwalkam, Michael Cogswell, David Crandall, and Dhruv Batra. Why m heads are better than one: Training a diverse ensemble of deep networks. *arXiv preprint arXiv:1511.06314*, 2015.

Dan Levi, Liran Gispan, Niv Giladi, and Ethan Fetaya. Evaluating and calibrating uncertainty prediction in regression tasks. *Sensors*, 22(15):5540, 2022.

Shen Li, Jianqing Xu, Xiaqing Xu, Pengcheng Shen, Shaoxin Li, and Bryan Hooi. Spherical confidence learning for face recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021.

Jeremiah Liu, Zi Lin, Shreyas Padhy, Dustin Tran, Tania Bedrax Weiss, and Balaji Lakshminarayanan. Simple and principled uncertainty estimation with deterministic deep learning via distance awareness. *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

Ilya Loshchilov and Frank Hutter. SGDR: Stochastic gradient descent with warm restarts. In *International Conference on Learning Representations (ICLR)*, 2017.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. *International Conference on Learning Representations (ICLR)*, 2013.

Zachary Nado, Neil Band, Mark Collier, Josip Djolonga, Michael Dusenberry, Sebastian Farquhar, Angelos Filos, Marton Havasi, Rodolphe Jenatton, Ghassen Jerfel, Jeremiah Liu, Zelda Mariet, Jeremy Nixon, Shreyas Padhy, Jie Ren, Tim Rudner, Yeming Wen, Florian Wenzel, Kevin Murphy, D. Sculley, Balaji Lakshminarayanan, Jasper Snoek, Yarin Gal, and Dustin Tran. Uncertainty Baselines: Benchmarks for uncertainty & robustness in deep learning. *arXiv preprint arXiv:2106.04015*, 2021.

Seong Joon Oh, Andrew C. Gallagher, Kevin P. Murphy, Florian Schroff, Jiyan Pan, and Joseph Roth. Modeling uncertainty with hedged instance embeddings. In *International Conference on Learning Representations (ICLR)*, 2019.

Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018.

Guillermo Ortiz-Jimenez, Mark Collier, Anant Nawalgaria, Alexander D'Amour, Jesse Berent, Rodolphe Jenatton, and Effrosyni Kokiopoulou. When does privileged information explain away label noise? *arXiv preprint arXiv:2303.01806*, 2023.

Yaniv Ovadia, Emily Fertig, Jie Ren, Zachary Nado, David Sculley, Sebastian Nowozin, Joshua Dillon, Balaji Lakshminarayanan, and Jasper Snoek. Can you trust your model's uncertainty? Evaluating predictive uncertainty under dataset shift. *Advances in Neural Information Processing Systems (NeurIPS)*, 32, 2019.

Joshua C Peterson, Ruairidh M Battleday, Thomas L Griffiths, and Olga Russakovsky. Human uncertainty makes classification more robust. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (CVPR)*, pages 9617–9626, 2019.

Karsten Roth, Timo Milbich, Samarth Sinha, Prateek Gupta, Bjorn Ommer, and Joseph Paul Cohen. Revisiting training strategies and generalization performance in deep metric learning. In *International Conference on Machine Learning (ICML)*, 2020.

Lars Schmarje, Vasco Grossmann, Claudius Zelenka, Sabine Dippel, Rainer Kiko, Mariusz Oszust, Matti Pastell, Jenny Stracke, Anna Valros, Nina Volkmann, et al. Is one annotation enough? A data-centric image classification benchmark for noisy and ambiguous label estimation. *arXiv preprint arXiv:2207.06214*, 2022.

Timm Schoening, Autun Purser, Daniel Langenkämper, Inken Suck, James Taylor, Daphne Cuvelier, Lidia Lins, Erik Simon-Lledó, Yann Marcon, Daniel OB Jones, et al. Megafauna community assessment of polymetallic-nodule fields with cameras: platform and methodology comparison. *Biogeosciences*, 17(12):3115–3133, 2020.

Tyler R Scott, Andrew C Gallagher, and Michael C Mozer. von Mises-Fisher loss: An exploration of embedding geometries for supervised learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021.

Yichun Shi and Anil K Jain. Probabilistic face embeddings. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2019.

Hyun Oh Song, Yu Xiang, Stefanie Jegelka, and Silvio Savarese. Deep metric learning via lifted structured feature embedding. In *Computer Vision and Pattern Recognition (CVPR)*, 2016.

Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research (JMLR)*, 15(1):1929–1958, 2014.

Dustin Tran, Jeremiah Liu, Michael W Dusenberry, Du Phan, Mark Collier, Jie Ren, Kehang Han, Zi Wang, Zelda Mariet, Huiyi Hu, et al. Plex: Towards reliability using pretrained large model extensions. *arXiv preprint arXiv:2207.07411*, 2022.

Gary Ulrich. Computer generation of distributions on the m-sphere. *Journal of the Royal Statistical Society: Series C (Applied Statistics)*, 33(2):158–163, 1984.

Uddeshya Upadhyay, Jae Myung Kim, Cordelia Schmidt, Bernhard Schölkopf, and Zeynep Akata. Posterior annealing: Fast calibrated uncertainty for regression. *arXiv preprint arXiv:2302.11012*, 2023.

Nina Volkmann, Johannes Brünger, Jenny Stracke, Claudius Zelenka, Reinhard Koch, Nicole Kemper, and Birgit Spindler. Learn to train: Improving training data for a neural network to detect pecking injuries in turkeys. *Animals*, 11(9):2655, 2021.

Nina Volkmann, Claudius Zelenka, Archana Malavalli Devaraju, Johannes Brünger, Jenny Stracke, Birgit Spindler, Nicole Kemper, and Reinhard Koch. Keypoint detection for injury identification during turkey husbandry using neural networks. *Sensors*, 22(14):5188, 2022.

C. Wah, S. Branson, P. Welinder, P. Perona, and S. Belongie. The Caltech-UCSD birds-200-2011 dataset. Technical Report CNS-TR-2011-001, California Institute of Technology, 2011.

Yeming Wen, Ghassen Jerfel, Rafael Muller, Michael W Dusenberry, Jasper Snoek, Balaji Lakshminarayanan, and Dustin Tran. Combining ensembles and data augmentation can harm your calibration. In *International Conference on Learning Representations (ICLR)*, 2021.

Ross Wightman. Pytorch image models, 2019.

Yongqin Xian, Bernt Schiele, and Zeynep Akata. Zero-shot learning - the good, the bad and the ugly. In *Computer Vision and Pattern Recognition (CVPR)*, 2017.

Donggeun Yoo and In So Kweon. Learning loss for active learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

Yang You, Jing Li, Sashank Reddi, Jonathan Hseu, Sanjiv Kumar, Srinadh Bhojanapalli, Xiaodan Song, James Demmel, Kurt Keutzer, and Cho-Jui Hsieh. Large batch optimization for deep learning: Training bert in 76 minutes. In *International Conference on Learning Representations (ICLR)*, 2020.

Ke Zou, Zhihao Chen, Xuedong Yuan, Xiaojing Shen, Meng Wang, and Huazhu Fu. A review of uncertainty estimation and its application in medical imaging. *arXiv preprint arXiv:2302.08119*, 2023.

A FAQ on URL benchmark and R-AUROC

A.1 Why AUROC with respect to R@1 and not accuracy?

First attempts in zero-shot learning literature measured the accuracy on unseen classes. This required learning not only an embedder, but also constructing classifier labels, usually through prototypes or attributes. Subsequent approaches in representation and deep metric learning seeked generally transferable embeddings, where no information is known about the test classes. So, it was impossible to give class logits and calculate an accuracy. Instead they measure the R@1. We have the same goal of transferring uncertainty estimates to unknown test conditions, so we built up the uncertainty benchmark on predicting whether the R@1 will be correct or wrong. This measures the risk inherent to the embedding, rather than the (classification) downstream task, which is unknown in advance. However, the risk in the embedding is a bottleneck to the potential subsequent classification layer and we indeed observe in Appendix C.3 that R@1 and accuracy are highly correlated, so that the difference is negligible.

A.2 Why a ranking-based metric (AUROC) and not probability-based calibration?

The AUROC measures if the predicted uncertainties are ranked such that the most uncertain have the most erroneous predictions. It is intriguing to use a different metric, which is to predict the probability of error directly as a value in $[0, 1]$. However, this metric fundamentally cannot be calibrated before the downstream task is known: In classification, we do not know how many classes there will be and beyond classification, our uncertainty metric should indicate the prediction risk, but without knowing the loss or its range in before, we cannot calibrate the uncertainty estimates. Therefore, the best strategy is to give a correctly *ranked* uncertainty metric, which is what AUROC measures. It can be calibrated into the $[0, 1]$ region for the downstream task at hand once data is available. Another point is that a $[0, 1]$ estimate has the trivial solution of always giving a random prediction the certainty being the prior over the classes. A ranking based metric cannot do that, as that would have an AUROC of 0.5. The AUROC thus forces the model to quantify *how* uncertain it is, not only that it *is* uncertain.

A.3 Does URL measure predictive, aleatoric, or epistemic uncertainty?

The R-AUROC in our setup measures the predictive uncertainty, in other words, the overall expected risk of a prediction. We believe this overall uncertainty is most relevant when seeking a reliable model. Note, however, that the AUROC uses a rank-based relative comparison of the uncertainty estimates. This means that a constant high uncertainty on out-of-distribution data does not suffice; instead, the model needs to quantify how uncertain it is, not just that it is uncertain. This is influenced both by the inherent ambiguity of the downstream sample (aleatoric) and how far it is from the upstream data (epistemic).

A.4 How is the AUROC implemented precisely?

We use the `TorchMetrics` implementation (Detlefsen et al., 2022). It applies the trapezoidal rule and uses every uncertainty value as a possible threshold.

A.5 Why CUB200-2011, CARS196, and Stanford Online Products?

Our definition of URL is agnostic of the datasets and can be applied to any downstream dataset. For this paper, we use CUB200-2011 (Wah et al., 2011), CARS196 (Krause et al., 2013), and Stanford Online Products (Song et al., 2016), because they are commonly used as zero-shot learning and representation learning datasets. We hope that this status prevents data-leakage to upstream pretraining datasets.

A.6 Can URL also be implemented for downstream datasets that don't have labels?

Throughout this paper, we measure the R@1 by seeing if the next neighbor of a test image has the same class label. This is to ensure the compatibility with representation learning benchmarks. One can also define a self-supervised R@1 (e.g., seeing if a crop of the same image is detected as belonging to that image). In this case, R-AUROC automatically generalizes to this form of supervision, as it still measures errors in R@1, regardless of how it was computed.

A.7 Can URL be applied beyond vision tasks?

Yes. URL can be deployed whenever there is a downstream dataset on which we measure a R@1.

B Reimplementations and hyperparameters

All methods benchmarked in this paper are re-implemented and hyperparameter tuned to ensure a consistent comparison. First, let us explain the hyperparameters shared by all approaches: As backbones, we use ResNet-50 (He et al., 2016) architectures with 224x224 inputs and 2048-dim max-pooled embeddings, and ViT-Medium (Dosovitskiy et al., 2021) with an input size of 256x256 split into 16x16 patches. All models are tuned for 32 epochs with a Lamb optimizer (You et al., 2020). 16 minibatches of size 128 are accumulated to reach a summed batch size of 2048, unless otherwise noted. The learning rate is fixed at the first epoch, then warmed up for five epochs and cooled down using a cosine scheduler (Loshchilov and Hutter, 2017). The learning rate is a hyperparameter for all approaches, searched over a range of $[0.0001, 0.01]$. The hyperparameter search is extended by more runs if the optimal value is close to a boundary. Let us now describe the approaches and their additional hyperparameters in detail. All optimal hyperparameters are reported in the code appendix.

B.1 Cross-Entropy (CE)

The **CE** baseline applies a cross-entropy loss to the class-logits output by a final linear layer appended to the embeddings. It has no hyperparameters except the learning rate.

B.2 InfoNCE

InfoNCE (Oord et al., 2018) uses two random crops per input image that are considered positive, whereas the remaining batch is considered negative. This results in a doubled VRAM usage, so that the batch size is reduced to 21 minibatches of 96 images each. **InfoNCE** uses an (inverse) temperature parameter t, tuned within $[8, 64]$. All embeddings e are normalized to lay on the unit sphere, where e_1^+ and e_2^+ denote the positive and e^- all negative embeddings. The final loss is

$$\mathcal{L} = -t \cdot {e_1^+}^\top e_2^+ + \log \sum_{e^- \in \text{Batch}} \exp\left(t \cdot {e_1^+}^\top e^-\right) + \log \sum_{e^- \in \text{Batch}} \exp\left(t \cdot {e_2^+}^\top e^-\right) . \tag{1}$$

B.3 MCInfoNCE

MCInfoNCE (Kirchhof et al., 2023) works similar to **InfoNCE**, except that it does not directly compare the embeddings, but samples from estimated posteriors $s_{1,i}^+ \sim \text{vMF}(e_1^+, \kappa(e_1^+))$, $s_{2,i}^+ \sim \text{vMF}(e_2^+, \kappa(e_2^+))$, $s_i^- \sim \text{vMF}(e^-, \kappa(e^-))$. The concentration, i.e., inverse uncertainty, κ is estimated from a 3-layer MLP attached to the embeddings. Its initial value is either randomly initialized or warmed up to 0.001, which is a hyperparameter. The second hyperparameter is its (inverse) temperature $t \in [8, 64]$. Like **InfoNCE**, it uses a reduced batch size of 21 times 96. The loss is obtained by calculating the **InfoNCE** loss for 16 samples s_i from the respective posteriors.

$$\mathcal{L} = \frac{1}{16} \sum_{i=1}^{16} -t \cdot {s_{1,i}^+}^\top s_{2,i}^+ + \log \sum_{s_i^- \in \text{Batch}} \exp\left(t \cdot {s_{1,i}^+}^\top s_i^-\right) + \log \sum_{s_i^- \in \text{Batch}} \exp\left(t \cdot {s_{2,i}^+}^\top s_i^-\right) \tag{2}$$

B.4 Expected Likelihood Kernel (ELK)

ELK uses a ProxyNCA formulation, as proposed in Kirchhof et al. (2022). Like **MCInfoNCE**, it parametrizes posteriors $\zeta = \text{vMF}(e, \kappa(e))$ from each image's normalized embeddings e and a 3-layer MLP for κ. Its initial value is either randomly initialized or warmed up to 0.001, which is a hyperparameter. **ELK** is supervised and learns vMF class distributions ρ_c for each class $c = 1, \ldots, C$. The concentrations of these classes are scaled up by the hyperparameter $t \in [8, 64]$, which takes a similar role to the inverse temperature in the previous losses. These are compared to the embedding posteriors via a distribution-to-distribution similarity function `elk_sim`. This is solved analytically, not requiring sampling. With ρ^* denoting the true class of the given sample, the loss can be written as

$$\mathcal{L} = -\texttt{elk\_sim}(\zeta, \rho^*) + \log \sum_{c=1}^{C} \exp\left(\texttt{elk\_sim}(\zeta, \rho_c)\right) \ . \tag{3}$$

B.5 Non-isotropic von Mises-Fisher (nivMF)

nivMF (Kirchhof et al., 2022) has the same hyperparameters and loss as **ELK**, except that the class proxy distributions ρ_c are non-isotropic vMFs. Since the expected-likelihoood between the image embeddings' vMFs and the classes' non-isotropic vMFs has no analytical solution, it is Monte-Carlo approximated with 16 samples.

$$\mathcal{L} = -\texttt{approx\_elk\_sim}(\zeta, \rho^*) + \log \sum_{c=1}^{C} \exp\left(\texttt{approx\_elk\_sim}(\zeta, \rho_c)\right) \ . \tag{4}$$

B.6 Hedged Instance Embeddings (HIB)

HIB (Oh et al., 2019), like **MCInfoNCE**, takes samples from estimated posteriors $s_{n,i} \sim$ vMF$(e_n, \kappa(e_n))$, where e_n are the image's L_2 normalized embeddings e_n and κ is estimated by a 3-layer MLP. Its initial value is either randomly initialized or warmed up to 0.001, which is a hyperparameter. **HIB** then calculates a matching probability by comparing the samples of every pair of images n, m in the batch: $p_{n,m} = \sum_{i=1}^{16} \text{sigmoid}\left(t \cdot s_{n,i}^\top s_{m,i} + b\right)$, where $t \in [8, 64]$ is a hyperparameter similar to the (inverse) temperature and $b \in [-8, 8]$ is a second hyperparameter. The matching probability should be high for images with the same label and low for images with different labels. Let $\mathcal{I}_{\text{same}}$ denote the pairs of images with the same label and $\mathcal{I}_{\text{different}}$ the pairs with different labels, both without self-matches. Then the loss is

$$\mathcal{L} = -\frac{1}{|\mathcal{I}_{\text{same}}|} \sum_{(n,m)\in\mathcal{I}_{\text{same}}} \log p_{n,m} + \frac{1}{|\mathcal{I}_{\text{different}}|} \sum_{(n,m)\in\mathcal{I}_{\text{different}}} \log p_{n,m} \ , \tag{5}$$

where $|\cdot|$ denotes the cardinality of the set. As opposed to the original implementation, we use cosine distances instead of L_2 distances and remove the prior regularizer. This is to make **HIB** more comparable to the other approaches in this paper. We also changed the second term from encouraging low log match probabilities for different labels $(-\log(1 - p_{n,m}))$ to discouraging high ones $(+\log p_{n,m})$, which stabilized training. **HIB** requires additional VRAM and thus uses 21 batches of size 96 (43 of size 48 on ViTs).

B.7 Heteroscedastic Classifier (HET-XL)

HET-XL (Collier et al., 2023) predicts a distribution in the embedding space for each image. It then takes samples and calculates a Monte Carlo estimate of the expected model output under the embedding distribution. As opposed to the other probabilistic embedding approaches from above, it operates in Euclidean space by predicting a Gaussian distribution $\mathcal{N}(\phi(x;\theta), \Sigma'(x;\theta_{\text{cov}}))$ with a low-rank approximation of the covariance matrix $\Sigma'(x;\theta_{\text{cov}}) = V(x)^\top V(x) + \text{diag}(d(x))$. $V(x)$ and $d(x)$ are output by a linear layer attached to the embeddings. The number of columns in V increases the rank of the low-rank approximation, but also the number of parameters that the final linear layer has to predict, and thus the memory requirements. We thus set this hyperparameter to 1 (exploratory experiments with a rank of 10 did not show increased performance). The final loss is

$$\mathcal{L}_{\text{cross-entropy}}\left(\mathbb{E}_{\epsilon'}\left[\text{softmax}_\tau(W^\top(\phi(x;\theta) + \epsilon'(x))\right], y\right) \quad \text{with} \quad \epsilon'(x) \sim \mathcal{N}\left(0, \Sigma'(x;\theta_{\text{cov}})\right) \ , \tag{6}$$

where the softmax temperature τ is a learnable parameter.

B.8 Spectral-normalized Neural Gaussian Processes (SNGP)

SNGP (Liu et al., 2020) predicts a Gaussian distribution

$$\mathcal{N}\left(\phi(x)^\top \beta, \phi(x)^\top \left(I + \Phi^\top \Phi\right)^{-1} \phi(x)\right) \tag{7}$$

over the class logits, which is cast into class probabilities via a mean-field approximation. These class probabilities are then trained under a **CE** loss. β is a learnable parameter matrix, $\phi(x) = \cos\left(Wh(x) + b\right)$ is a feature embedding based on frozen random parameters W and b, and $\Phi^\top\Phi$ is the empirical covariance matrix of the feature embeddings over the training dataset. The method also applies spectral normalization to the hidden weights in each layer in order to satisfy input distance awareness. We treat whether to apply spectral normalization through the network and whether to use layer normalization in the last layer as hyperparameters.

B.9 Direct Loss Prediction (Losspred)

Losspred trains a classifier under a cross-entropy loss and uses its uncertainty module κ, a 3-layer MLP attached to the embedding space, to predict the value of the cross-entropy loss. Both components are balanced by the hyperparameter $\lambda \in [0.01, 0.99]$:

$$\mathcal{L} = \mathcal{L}_{\text{cross-entropy}}(x, y) + \lambda \left(\kappa(x) - \mathcal{L}^{\text{detached}}_{\text{cross-entropy}}(x, y)\right)^2 \ . \tag{8}$$

The gradients of $\mathcal{L}^{\text{detached}}_{\text{cross-entropy}}$ inside the L_2 loss are detached to prevent fitting it to $\kappa(x)$, instead of the other way around. Besides λ, a second hyperparameter is whether or not to warm up κ to 0.001.

B.10 Deep Ensemble

Ensemble (Lakshminarayanan et al., 2017) has 10 classifier heads attached to the embedding space. Their logits are transformed into probabilities by a softmax and then averaged. This average is trained under a **CE** loss. **Ensemble** has no hyperparameters other than the learning rate.

B.11 MCDropout

MCDropout (Srivastava et al., 2014) trains with a dropout rate of $[0.05, 0.25]$, which is a hyperparameter. During inference time, it keeps the dropout activated to sample 10 logits and averages them like **Ensemble**.

C Additional results

C.1 Pretrained models already close the gap in terms of R@1

This section compares the models in terms of their R@1. To this end, we hyperparameter-tuned them with respect to R@1, and not R-AUROC. Similar to the main text, we also add an additional baseline that was trained on the downstream classes, where the original test split was split into equal sized train and test splits.

Fig. 6 shows these performances. As opposed to the R-AUROC, we can see that the gap between pretrained zero-shot and many-shot models is much tighter in terms of the R@1. The best pretrained ResNet-50 has an average R@1 of 0.48 vs. 0.54 when training on the downstream data.

Surprisingly, when it comes to R@1, **CE** is among the best two approaches both for ResNet and ViT backbones. In fact, three of the four best approaches on ViTs and two of the four on ResNets rely on **CE** as part of their loss. The approaches that do not rely on **CE** are probabilistic embeddings – **nivMF** on ResNet and ViT and **ELK** on ResNet.

C.2 R@1 and R-AUROC correlate negatively in most models

In extension to the plot that compared the best hyperparameter setup for R-AUROC to the best for R@1, Fig. 7 and Fig. 8 present the trade-offs for all tested hyperparameters.

The general picture is the same as in the comparison of best vs best: Most models show a trade-off between achieving the best R-AUROC and the best R@1. This is indicated by a negative rank correlation in 15 out of 22 models. Still, there are some models where the uncertainty estimation and prediction performances correlate moderately positively ($0.4 \leq$ Rank Corr. ≤ 0.72). These are **InfoNCE** and **MCInfoNCE** on ResNets, and **HET-XL**, **nivMF**, and **HIB** on ViTs.

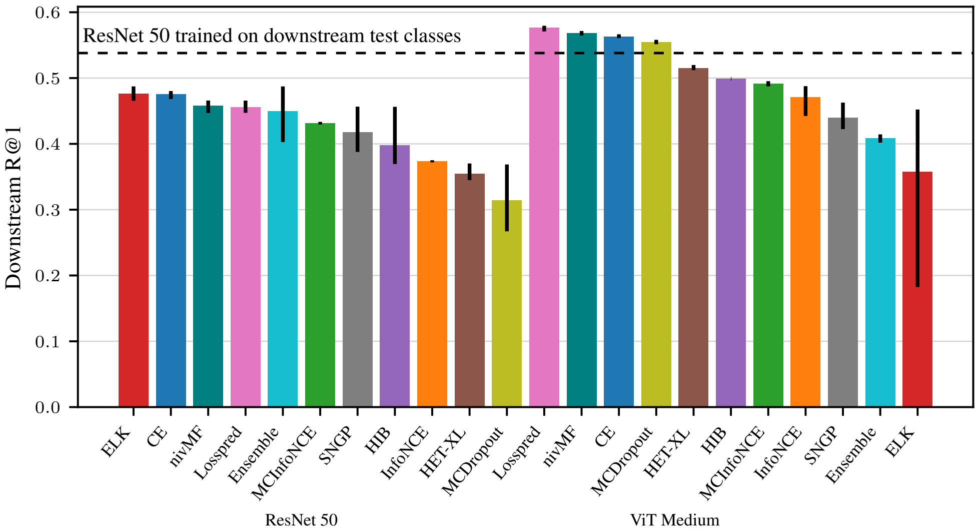


Figure 6: Pretrained models achieve an almost as good R@1 on unseen downstream data as models trained on the downstream data. Minimum, average, and maximum R@1 per model. Model hyperparameters were optimized w.r.t. R@1.

As mentioned in Appendix B, each approach was tested on 10 hyperparameters, with 2 additional runs on other seeds for the best hyperparameters. The reason that some plots show less than 12 points is that those had a R@1 < 0.1 on the downstream datasets and were excluded from the analysis. Some plots also show more than 12 points. This is because their best hyperparameter for R@1 was unlike that for R-AUROC, adding another 2 runs on different seeds for the R@1. Further, some approaches had optimal hyperparameters close to the original search bounds, such that the search was extended, leading to additional points in the plots.

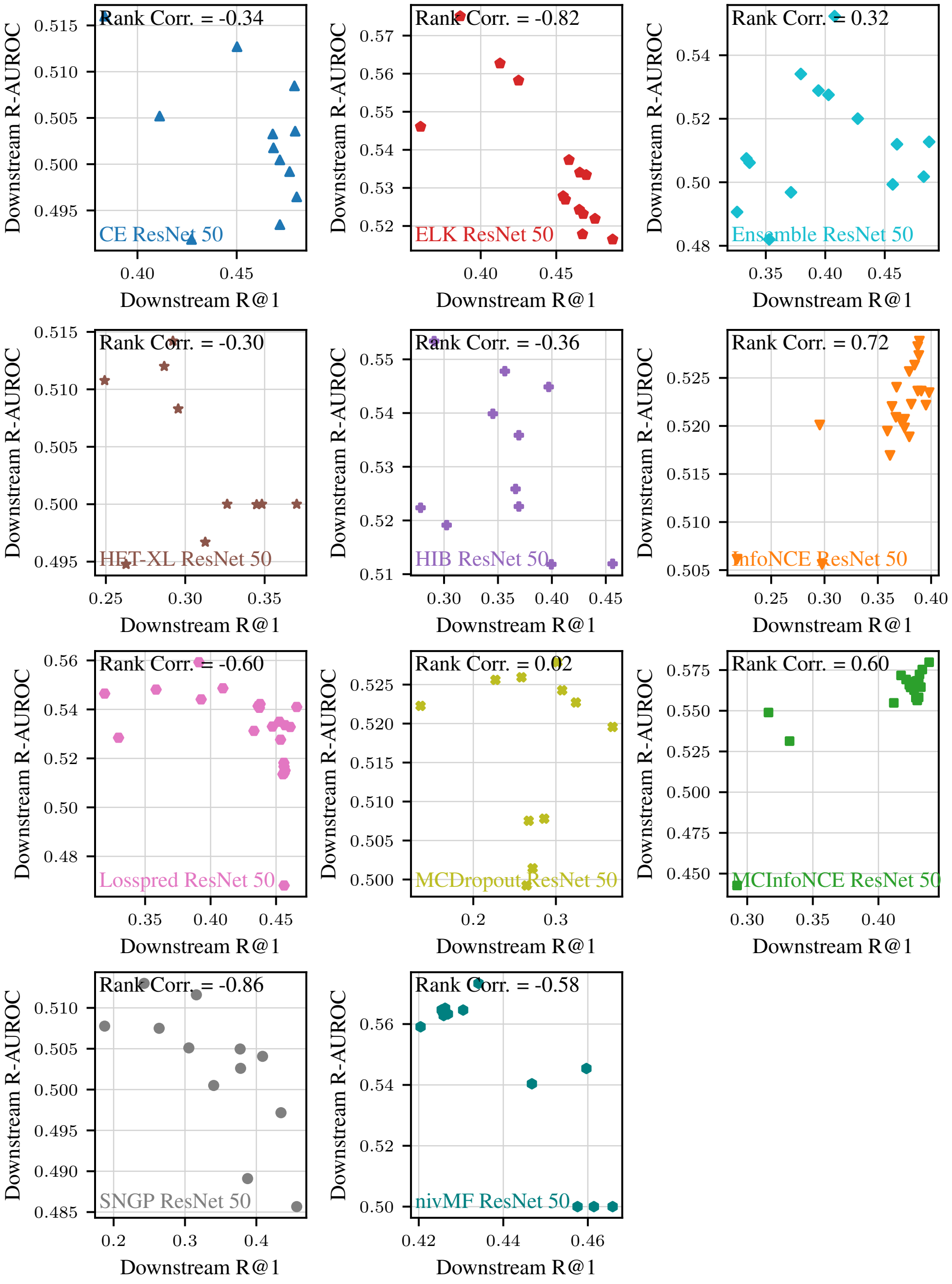


Figure 7: R@1 and R-AUROC are negatively correlated on seven out of ten model classes with ResNet backbones. Plot shows all tested hyperparameter combinations.

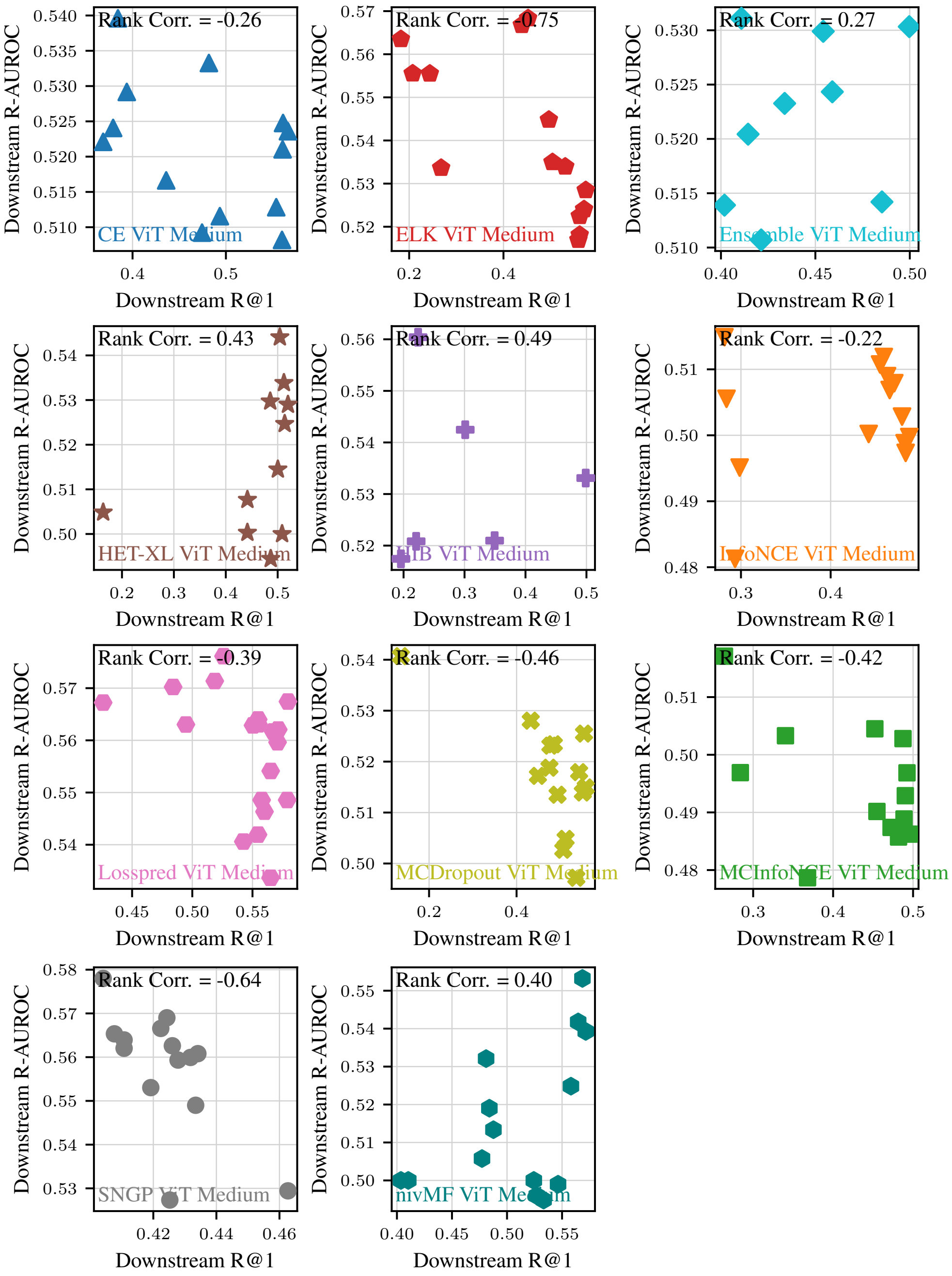


Figure 8: R@1 and R-AUROC are negatively correlated on seven out of ten model classes with ViT backbones. Plot shows all tested hyperparameter combinations.

C.3 Further correlation of up- and downstream metrics

This section reports further metrics for each model. These include a new metric (top-1 accuracy on upstream ImageNet) and all downstream metrics (R@1, R-AUROC, percentage where cropped image has higher uncertainty) on the upstream dataset.

Fig. 9 shows the pairwise correlations between all metrics, for every tested hyperparameter setting. Let us first consider the new metric, ImageNet top-1 accuracy, on its own. The best-performing models are a **CE** and a **nivMF** ViT, with an accuracy of 0.84 each. Other than them, **Losspred** and **HET-XL** on ViTs also have a high accuracy, similar to the results on the R@1 benchmark.

Regarding correlations between metrics, accuracy and R@1 are highly correlated, as expected. Further, models with a high accuracy or R@1 on ImageNet also have a high R@1 on the downstream data, resulting in the small gap explained in Appendix C.1. While up- and downstream R-AUROC do not correlate, up- and downstream percentage of cropped images having a higher uncertainty correlate nearly linearly. This reinforces that the percentage can be considered as a general notion of uncertainty. It should, however, be noted that **ELK** ViTs already achieve a performance of 0.99 on this metric, even on downstream data, such that it is not able to guide the field as well as R-AUROC.

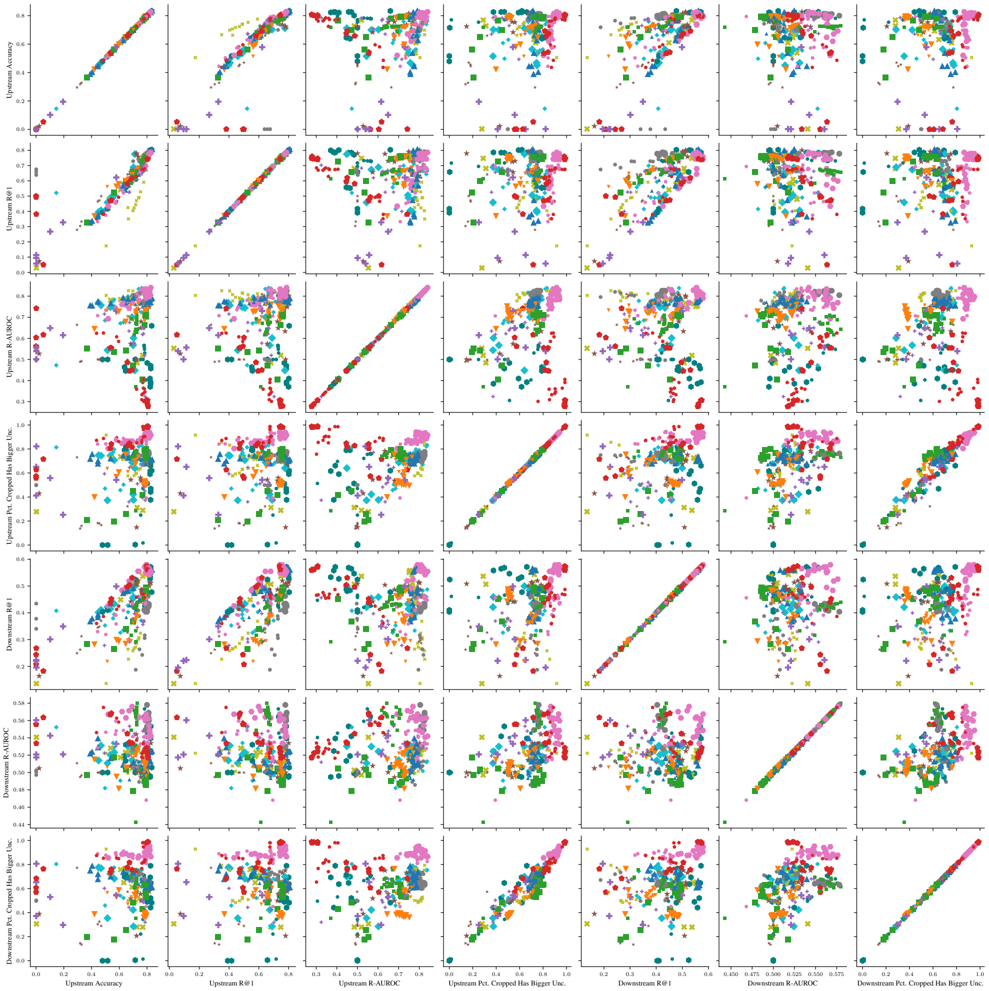


Figure 9: Correlations between several up- and downstream metrics across all models and hyperparameter choices.

C.4 Class-entropy is useful for OOD detection

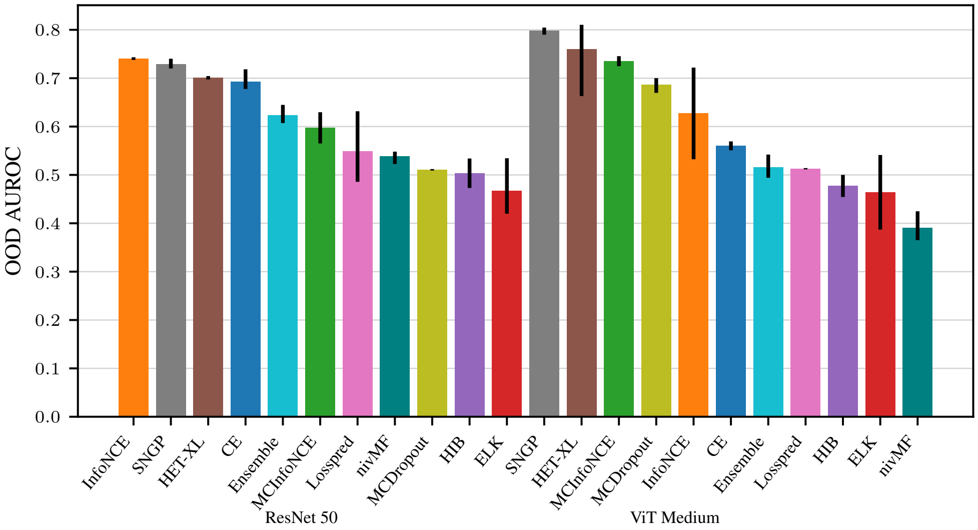


Figure 10: Cross-entropy and embedding norm-based uncertainty estimators have the best OOD detection capabilities. Averaged AUROC of distinguishing ImageNet vs CUB, ImageNet vs CARS, and ImageNet vs SOP. Error bars indicate minimum/maximum performance across seeds.

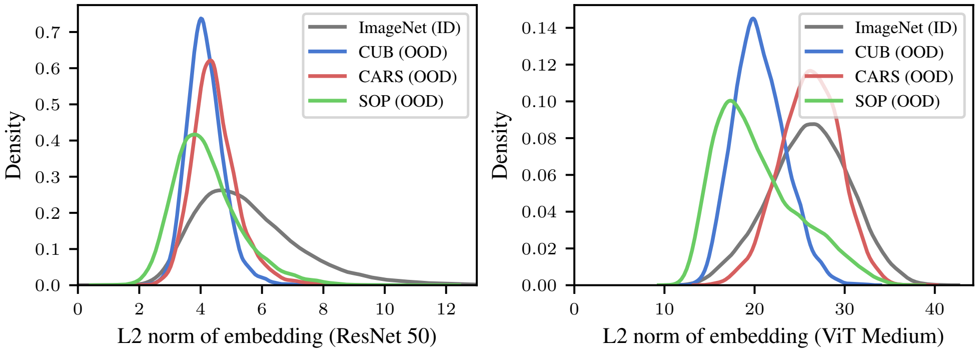


Figure 11: Embeddings of OOD images tend to have smaller L_2-norms than embeddings of ID images. Embedding norms are from InfoNCE models that use the (inverse) norm as uncertainty estimate (without using them during training) and reach an OOD AUROC of 0.73.

In this section, we study if uncertainties on downstream data are generally higher than for the upstream data. To this end, we perform out-of-distribution detection experiments: Using the same pretrained models as before, we calculate their uncertainties on downstream dataset samples and on an equally sized set of upstream samples. We quantify how well the predicted uncertainties distinguish whether the sample was from a downstream dataset (1) or not (0) by calculating the AUROC on ImageNet vs CUB, ImageNet vs CARS, and ImageNet vs SOP, and averaging them across the datasets.

Fig. 10 shows the average result across all seeds, along with minimum and maximum performances. Generally, pretrained models perform differently than in the URL benchmark in the main text. This is because the OOD task benchmarks epistemic uncertainty (i.e., OOD data just has to have generally high uncertainties). On the other hand, the URL benchmark tests predictive uncertainties on OOD data. There, it is not enough to predict high values on OOD data but the models need to differentiate within them, which lays more focus on aleatoric uncertainty. In more detail, models that directly predict variances or losses do not provide as good OOD performance. Models that use class-entropy as uncertainty estimates (**SNGP**, **HET-XL**, **CE**, **Ensemble**, and, only on ViTs, **MCDropout**) and also **InfoNCE**, which uses the norm of the embedding vector, do work well.

There is an intuitive explanation for this. The latter models implicitly provide epistemic uncertainty estimates by construction: ResNets and ViTs embed inputs with less known features closer to the origin. As an example, Fig. 11 shows that for **InfoNCE**, the embedding norms of OOD samples are smaller. Note that this is no trained behaviour; **InfoNCE** only trains on normalized embeddings, so the norms occur naturally. In **InfoNCE**, we use this embedding norm directly as uncertainty estimator. But the same happens in models that use the class entropy: Embeddings with small norm lay close to the origin, where they lead to uniform distributions over the classes, i.e., a high entropy.

In summary, the OOD experiment reveals that probabilistic embeddings and loss prediction methods provide aleatoric uncertainty estimates, whereas models that explicitly or implicitly use the embedding norm provide good epistemic uncertainty estimates. Some models, like **SNGP**, provide a mixture and are good in both tasks.

C.5 Uncertainties on mixtures of in- and out-of-distribution data

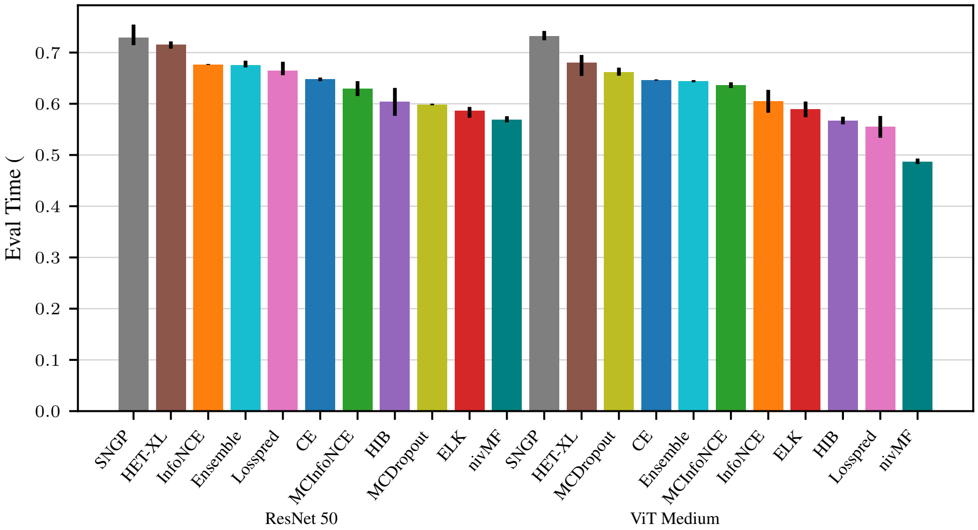


Figure 12: Cross-entropy based uncertainties provide the best mix of OOD detection capability and ordering within ID and OOD uncertainties. R-AUROC on mixed ImageNet+CUB, ImageNet+CARS, and ImageNet+SOP data. Error bars indicate minimum/maximum performance across seeds.

In some applications, models might encounter a mixture of in-distribution and out-of-distribution data. In this case, the model both needs to assign higher uncertainties to the OOD data and it needs to differentiate the uncertainties within both the ID and the OOD sets. This blends the OOD detection of the previous experiment with traditional ID calibration and the OOD calibration of the URL benchmark.

Fig. 12 measures the R-AUROC on a mixture of upstream and downstream data. As in the previous experiment, we use 50/50 splits of ImageNet+CUB, ImageNet+CARS, and ImageNet+SOP, and average across those combinations. We find that **HET-XL** and **SNGP**, which previously performed well on both OOD detection and the URL benchmark, also perform well on this mixed task. The remaining models tend to follow the ranking of the OOD benchmark rather than that of URL. This indicates that good epistemic uncertainty estimation outweighs aleatoric uncertainty estimation in this task. This is intuitive, because the R-AUROC measures how likely it is that a wrong prediction has a higher uncertainty than a correct one. In such a 50/50 split of ID and OOD data, the capability to distinguish ID from OOD data, and thus data with generally less errors from data with generally more errors, thus leads to a higher R-AUROC.

This serves to demonstrate that the quality of an uncertainty estimate always depends on the task and setup at hand. While OOD detection and ID calibration are ideally both reflected within the very same predictive uncertainty value, both are of different importance depending on the data mixture.

C.6 Few-shot uncertainties starting from pretrained models

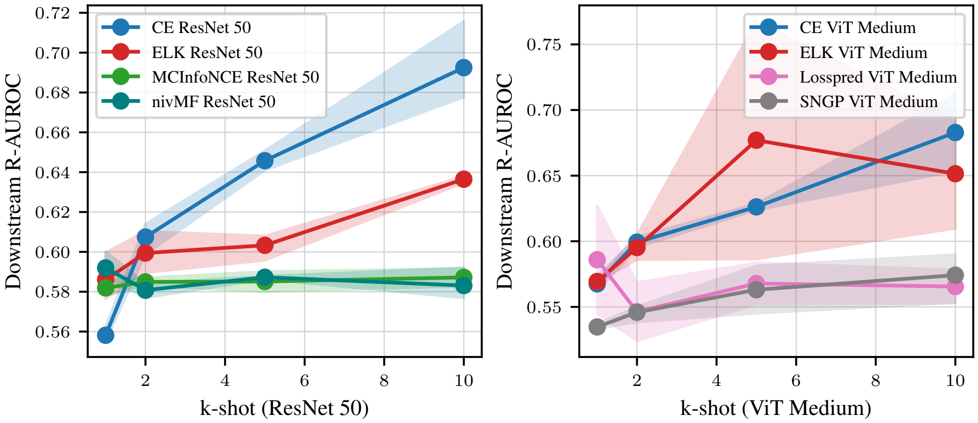


Figure 13: Cross-entropy training on the downstream test data reaches the zero-shot AUROC of 0.6 at 2-5 samples. R-AUROC of pretrained models when finetuned on the downstream test classes.

In this section, we provide an initial attempt on a few-shot experiment. We use both the best three pretrained models (**ELK**, **MCInfoNCE**, and **nivMF** for ResNet 50; **ELK**, **Losspred**, and **SNGP** for ViT Medium) along with the **CE** model as pretrained checkpoints and continue to train under the same hyperparameters and losses on $k \in \{1, 2, 5, 10\}$ samples of each class of the downstream datasets. This is done on the test classes of the downstream classes, which were separated into disjoint train and test samples for this experiment. We train on each CUB/CARS/SOP separately and average their performance.

Fig. 13 shows the minimum, maximum and average performance across the seeds. First, we can see that the normal cross entropy training reaches the zero-shot R-AUROC of the currently best pretrained models at around 2 samples per class (totaling in 200 samples on CUB, 196 on CARS, and 22636 on SOP). This again demonstrates the point that the challenge URL addresses is unsolved yet. It also shows that knowledge of the specific downstream task can increase the uncertainty estimators quality even at only a few samples. Second, we find that most pretrained models increase their performance as well. This, however, happens not for all models and is highly noisy. We attribute this to the simple setup of our experiment, which uses the same hyperparameters as for the pretraining and performs standard training as opposed to specialized few-shot methods. Finding best practices to tune pretrained uncertainties to downstream few-shot tasks is a promising undertaking for future research.

C.7 Uncertainty estimation does not add significant computational costs

Table 1 reports the computational complexity during all benchmarks. It shows the number of parameters as proxy for RAM usage, the duration of the first training epoch and evaluation, and the time needed for each sample during evaluation. These results were collected on the go and there are possible confounders such as network storage workload. We thus recommend to interpret them as rough indicators. Note also that ViTs were run on NVIDIA A100 GPUs, and ResNets on NVIDIA RTX2080TIs (except **HET-XL**on ResNet, due to RAM usage).

First, we see that explicit uncertainty estimation does not come at a high RAM cost. The 3-layer MLP serving as uncertainty head for **MCInfoNCE**, **ELK**, **nivMF**, **HIB**, and **Losspred** adds 4.2M parameters to a ResNet or 2.6M to a ViT. The difference is due to the ViT's lower-dimensional embedding space. Bigger increases occur only for ensembles, which uses additional classifiers with $10 \cdot 2.1$M parameters on ResNets and each $10 \cdot 0.5$M on ViTs.

	Model	Parameters (Millions)	Epoch Time (s)	Inference time per sample (ms)
ResNet 50 on RTX 2080TI	**CE**	25.6	5971	3.7
	InfoNCE	23.5	6703	3.7
	MCInfoNCE	27.7	6656	3.8
	ELK	29.8	6703	3.8
	nivMF	29.8	6256	3.8
	HIB	29.8	7279	3.8
	HET-XL (on A100)	33.9	(4189)	(2.3)
	Losspred	29.8	6526	3.7
	MCDropout	25.6	7105	13.1
	Ensemble	46.0	6002	3.7
	SNGP	28.7	7632	3.7
ViT Medium on A100	**CE**	38.9	4922	2.8
	InfoNCE	38.3	7804	2.7
	MCInfoNCE	41.0	9883	2.4
	ELK	41.5	4838	2.7
	nivMF	41.5	4121	2.4
	HIB	41.5	3950	2.5
	HET-XL	39.4	4205	2.4
	Losspred	41.5	3814	2.6
	MCDropout	38.9	4721	9.6
	Ensemble	44.0	4107	2.5
	SNGP	42.0	4811	2.9

Table 1: Computational costs of all approaches. Epoch times include training on ImageNet as well as evaluating on all eight downstream datasets.

Training times should be interpreted with caution due to the aforementioned network storage. However, in general uncertainty estimates do not seem to exceedingly increase the train time, with between -22% and +28% over the **CE** baseline. Taking multiple samples during training (**MCInfoNCE**, **nivMF**, **HIB**, **HET-XL**) also does not systematically increase runtime. This is likely due to their efficient sampling implementations (Kirchhof et al., 2022; Davidson et al., 2018; Ulrich, 1984). The only consistent runtime cost occurs when training unsupervised models (**InfoNCE**, **MCInfoNCE**), which need to forward propagate two augmentations of each sample to obtain have positive pairs.

Similarly, providing an uncertainty estimate at inference takes only up to 0.1 additional milliseconds on ResNets, because the uncertainties are all calculated within a single forward pass, and sampling was only required for the losses during training. The only exception here is **MCDropout**, which requires making 10 full forward passes during inference, which increases the time by a factor of roughly 3.6. The factor is not 10, because loading the data and storing the results is part of the measured elapsed time.

In summary, we find that uncertainty estimates only have small computational costs if they are implemented in a forward fashion.

D

Pretrained Visual Uncertainties

This appendix contains the full paper and appendix discussed in Chapter 5, reproduced with permission.

Michael Kirchhof, Mark Collier, Seong Joon Oh, and Enkelejda Kasneci. Pretrained visual uncertainties, 2024. First published in Arxiv. Available from https://arxiv.org/abs/2402.16569. Reproduced under a Creative Commons Attribution 4.0 International License (`https://creativecommons.org/licenses/by/4.0/`). Under submission.

Pretrained Visual Uncertainties

Michael Kirchhof[1] Mark Collier[2] Seong Joon Oh[3] Enkelejda Kasneci[4]

Abstract

Accurate uncertainty estimation is vital to trustworthy machine learning, yet uncertainties typically have to be learned for each task anew. This work introduces the first pretrained uncertainty modules for vision models. Similar to standard pretraining this enables the zero-shot transfer of uncertainties learned on a large pretraining dataset to specialized downstream datasets. We enable our large-scale pretraining on ImageNet-21k by solving a gradient conflict in previous uncertainty modules and accelerating the training by up to 180x. We find that the pretrained uncertainties generalize to unseen datasets. In scrutinizing the learned uncertainties, we find that they capture aleatoric uncertainty, disentangled from epistemic components. We demonstrate that this enables safe retrieval and uncertainty-aware dataset visualization. To encourage applications to further problems and domains, we release all pretrained checkpoints and code under https://github.com/mkirchhof/url.

1. Introduction

With every prediction comes the risk of an error. Uncertainty estimates quantify this expected error in order to defer predictions and catch errors before they happen, a key requirement for trustworthy machine learning (Mucsányi et al., 2023). Uncertainty quantification has seen tremendous advances in recent years, bringing principled methods such as Gaussian processes (Liu et al., 2020) and probabilistic embeddings (Oh et al., 2019; Kirchhof et al., 2023a; Kim et al.; Nakamura et al., 2023) to large-scale computer vision (Tran et al., 2022; Dehghani et al., 2023; Collier et al., 2023). Recent benchmarks reveal that they excel at their metrics and are ready for application (Galil et al., 2023a;b). However, there is a lack of widespread adoption of uncertainty methods by practitioners. The hurdle is that modern uncertainty quantification methods can be complex, making them difficult to implement and increasing the inference costs. So what if uncertainty estimates were as easy to access as being shipped along with every pretrained model?

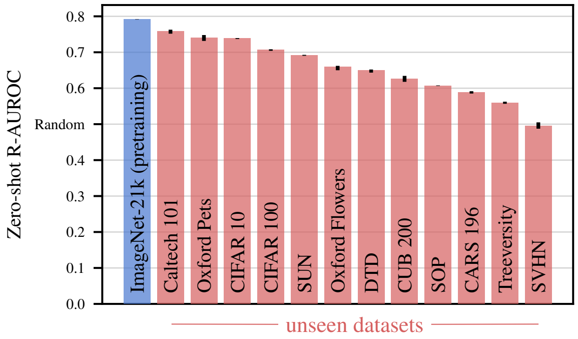


Figure 1. Our pretrained uncertainties generalize to unseen datasets. The R-AUROC measures the quality of uncertainty estimates on zero-shot datasets, see Section 4.

We seek a method that is simple to implement and train, even on large scales, and most importantly does not interfere with the main objective of a practitioner's model. One group of recent methods stands out on these terms: Feed-forward uncertainties (Cui et al., 2023; Kirchhof et al., 2022; Yoo & Kweon, 2019; Oh et al., 2019). They an auxiliary uncertainty head to a deep network that is evaluated alongside every forward pass. Not only is this cheap and simple to implement, it was also recently found that its uncertainty estimates transfer well (Kirchhof et al., 2023b).

Our work solves a remaining gradient conflict of these feed-forward uncertainties to guarantee non-interference with the main objective. We also implement massive caching in our pretraining pipeline, reducing the train time by a factor of up to 180x. This enables us to scale up both the pretraining dataset to ImageNet-21k Winter-2021 (ImageNet-21k-W) (Deng et al., 2009) and the vision backbone to large Vision Transformers (Dosovitskiy et al., 2021).

Figure 1 shows that our pretrained uncertainties now transfer beyond the train dataset to unseen datasets, outperforming previous zero-shot uncertainties (Kirchhof et al., 2023b). We find that the learned uncertainties are generalizable no-

[*]Equal contribution [1]University of Tübingen, Germany [2]Google Research, Switzerland [3]University of Tübingen, Tübingen AI Center, Germany [4]TUM University, Munich, Germany. Correspondence to: Michael Kirchhof <michael dot kirchhof at uni dash tuebingen dot de>.

Preliminary work.

tions of aleatoric uncertainty disentangled from epistemic uncertainty. This enables multiple use-cases: Beyond error prediction, we showcase novel applications like safe retrieval and uncertainty-aware dataset visualization. To facilitate widespread adoption, we release checkpoints for our pretrained uncertainties along with efficient code to pretrain them for arbitrary model architectures.

In summary, our contributions are:

- We develop a method which learns pretrained uncertainties that transfer zero-shot.
- This is based on fixing a gradient conflict in previous feed-forward uncertainties and speeding up the training by 180x, enabling large-scale pretraining (Section 3.3).
- Our uncertainties represent aleatoric uncertainty, disentangled from epistemic uncertainty (Section 4.5).
- We apply the uncertainties to improve the reliability of retrieval and to aid data visualization (Section 5).

2. Related Work

Large-scale uncertainty quantification. Uncertainty quantification has recently been scaled rapidly. Within one year, the largest vision models capable to perform uncertainty quantification grew from 1B (Tran et al., 2022) to 22B parameters (Dehghani et al., 2023). Benchmarks have increased accordingly. Previous surveys on out-of-distribution detection of 32x32 sized images (Ovadia et al., 2019) have been scaled to real-world images (Galil et al., 2023a) and to several additional tasks (Galil et al., 2023b). One such task is the uncertainty-aware representation learning (URL) benchmark that pretrains an uncertainty estimator and then tests zero-shot uncertainties on unseen datasets (Kirchhof et al., 2023b). Our work sets a new state-of-the-art on this task. In particular, we provide pretrained uncertainty modules for large computer vision models, independent from the classes or specific task of a dataset.

Feed-forward uncertainties. State-of-the-art models attempt to give such transferable uncertainties by moving away from classifier-layer uncertainties and towards uncertainties in the representation space (Collier et al., 2023). This approach falls under the category of feed-forward or deterministic uncertainties (Postels et al., 2022). They have a specialized uncertainty module that outputs predicted uncertainties during the forward pass of the model at minimal computational costs, which enables scaling. A variational take on this are probabilistic embeddings (Oh et al., 2019; Chun, 2023; Kim et al.; Nakamura et al., 2023) that output a variance estimate to give a distribution of possible representations instead of just one. This has recently been proven to recover the aleatoric uncertainty of the true posterior (Kirchhof et al., 2023a) and improve retrieval performance (Karpukhin et al., 2022). As opposed to such indirect approaches, a second group of feed-forward approaches makes uncertainty quantification a direct regression task (Yoo & Kweon, 2019; Cui et al., 2023; Lahlou et al., 2023; Laves et al., 2020). In initial experiments, we found this direct approach to scale better. We use it as a starting point to develop our pretrained uncertainties in the next section, overcoming some remaining challenges to enable scaling.

3. Developing Pretrained Uncertainties

In this section we set out the desired properties of our pretrained uncertainties and then extend a popular feed-forward uncertainty method to satisfy these properties in the large-scale pretraining case.

3.1. Basic Principles

We develop pretrained uncertainties from basic principles for scalability and ease of use, in order of importance:

(i) **Non-interference with primary task**. Adding pretrained uncertainties to a model should not worsen the performance of the pretrained model's primary objective, e.g., its accuracy.

(ii) **Generalization**. The predicted uncertainty estimates should reflect general forms of uncertainty that transfer to unseen datasets and tasks beyond the pretraining data and task.

(iii) **Flexible adjustment**. The uncertainties should be general enough to adapt to new tasks and/or datasets that downstream practitioners might introduce.

(iv) **Minimal overhead**. Providing uncertainties add only minimal runtime and memory usage to the main task prediction model.

(v) **Scalable optimization**. Training should converge stably to ensure scalability to large pretraining corpora.

In summary, we seek a *download and forget* approach that requires minimal interventions from practitioners.

3.2. Recap: Loss Prediction

We now introduce a simple yet general uncertainty method that we build our method upon. From a decision theory perspective, giving an uncertainty estimate means predicting how wrong one thinks one's estimate is. The key is that any task's level of wrongness is defined by its loss $\mathcal{L}_{\text{task}}$. So in loss prediction (Yoo & Kweon, 2019; Cui et al., 2023; Lahlou et al., 2023; Laves et al., 2020), the model has an additional module u that predicts the model's own loss at

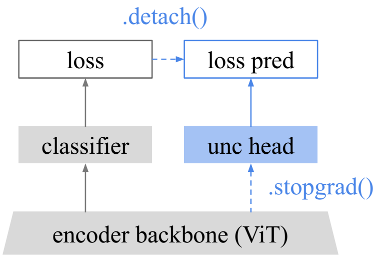


Figure 2. Pretrained uncertainties are returned by an auxiliary head (blue) that is trained to predict the classification loss of each image.

each of its predictions. This is learned via a L_2 loss between u and the (gradient-detached, det.) main task loss $\mathcal{L}_{\text{task}}^{\text{det.}}(y, f(x))$ at every sample (x, y). This is trained along with the main task (Kirchhof et al., 2023b), yielding the combined objective $\mathcal{L}$:

$$\mathcal{L} = \mathcal{L}_{\text{task}}(y, f(x)) + (u(e(x)) - \mathcal{L}_{\text{task}}^{\text{det.}}(y, f(x)))^2 \,. \quad (1)$$

The uncertainty module u is implemented as a small MLP head $u(e(x))$ on top of the model representations $e(x)$. This makes it cheap to compute during the forward pass, fulfilling the minimal overhead principle (iv). Loss prediction's uncertainties also adapt to any loss, fulfilling the flexibility principle (iii), and transfer well (Kirchhof et al., 2023b), fulfilling the generalization principle (ii).

Yet, loss prediction's implementation has a limitation: Figure 3a depicts a conflict between the uncertainty and the classification task. Their gradients interact negatively with one another, deteriorating the joint backbone and violating the non-interference principle (i). To resolve it, the current implementation stops early, roughly at epoch 12 in the plot. However, this early stopping is at odds with the scalability principle (v). Below, we fix these issues.

3.3. Enhancing Loss Prediction

We introduce four changes to the above loss prediction:

1. Stopgrad. As visualized in Figure 2, we add a stopgrad behind the uncertainty module. This prevents its gradients from flowing to the backbone and interfering with the classification head. This strictly ensures the non-interference principle (i), and, indeed, the training now converges robustly, see Figure 3b. This way, uncertainties can be trained in parallel to the main task, as opposed to only in post-hoc.

2. No early stopping. With the gradient conflict resolved, there is no more need for early stopping. The uncertainty head converges to its maximum at the end of the training in Figure 3b, making the training scalable as per principle (v).

3. Cache everything. Since the classification head and backbone are now independent from the uncertainty head, we pretrain and then freeze them before training the uncertainty module. Only the uncertainty objective remains:

$$\mathcal{L} = (u(e(x)) - \mathcal{L}_{\text{task}}^{\text{det.}}(y, f(x)))^2 \quad (2)$$

This can be optimized efficiently: The uncertainty module uses only the representations $e(x)$ as inputs, and, likewise, the task loss depends only on them via $f(x) = c(e(x))$, where c is the classifier layer. So, we do not need to load the images x or run them through the backbone, but can cache the representations $e(x)$ of the whole training process once (all epochs, including random augmentations). When learning the uncertainty module on top of a pretrained model, this increases the train speed by a factor of 180x and reduces the memory usage so far that we can pretrain uncertainties even for large models on single GPUs (or even CPUs). This paves the way for scalability: After caching the representations once, training the uncertainty module of a ViT-Large for seven ImageNet-21k-W epochs takes 2:26 hours on a single V100 GPU as opposed to 18 days with the standard loss prediction implementation.

4. Scale-free uncertainties. With the current L_2 loss, the uncertainty module is trained to match the scale of the pretraining loss. However, a downstream user might switch to a different loss on a different scale, which would introduce destructive gradients during finetuning. Thus, we switch to the ranking-based objective of Yoo & Kweon (2019):

$$\mathcal{L} = \max(0, \mathbb{1}_{\mathcal{L}} \cdot (u(e(x_1)) - u(e(x_2)) + m)), \quad (3)$$

$$\text{s.t. } \mathbb{1}_{\mathcal{L}} := \begin{cases} +1\,, \text{if } \mathcal{L}_{\text{task}}^{\text{det.}}(y_1, f(x_1)) > \mathcal{L}_{\text{task}}^{\text{det.}}(y_2, f(x_2)) \\ -1\,, \text{else} \end{cases} \quad (4)$$

For every pair of images x_1 and x_2, the indicator function compares which image has the higher primary task loss $\mathcal{L}_{\text{task}}^{\text{det.}}$. Then, the uncertainty u of that sample is forced to be higher than that of the other sample, by a margin of at least $m = 0.1$. This unties the uncertainty values from the scale of the task loss, improving on the flexibility principle (iii).[1]

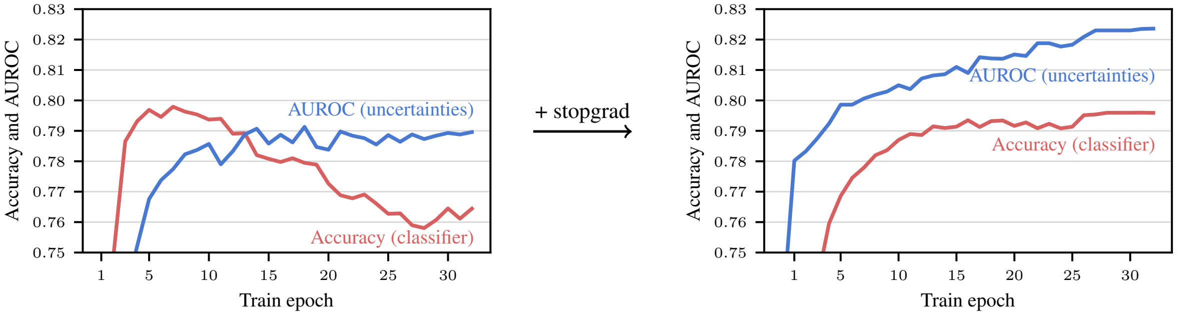
4. Experiments

We now study the main objective of pretrained uncertainties, their performance on downstream datasets. Additionally, we investigate which types of uncertainties they represent.

4.1. Experimental Setup

We are interested in how good our uncertainties perform on unseen datasets. This challenging zero-shot transfer is simple with our pretrained uncertainties, our enhanced loss

[1] Being scale-free also means being uncalibrated. However, this is not a disadvantage for pretrained uncertainties, because during pretraining the downstream task is unknown, hence it is impossible to be calibrated for the unseen downstream task in the first place.



(a) Vanilla Loss Prediction

(b) Ours: Loss Prediction + Stopgrad

Figure 3. (a) The uncertainty and classification heads of Loss Prediction are in conflict. We solve this in (b) by adding a stopgrad. It ensures that the uncertainty head's gradients do not interfere with those of the classifier head, stabilizing the performance of both. The uncertainty and classifier heads were finetuned on ImageNet-1k on a pretrained (but unfrozen) ViT-Base backbone.

prediction module outputs an uncertainty $u(x)$ for every image x from the downstream task. In order to measure the quality of these uncertainty estimates, we follow URL (Kirchhof et al., 2023b) in using the representation AUROC (R-AUROC) metric. It performs a 1-nearest neighbor classification on the representations $e(x)$ on all images of a downstream dataset. The uncertainties $u(x)$ should then be higher for images that are misclassified, which is quantified by the area under the ROC curve between the predicted uncertainties and whether or not the representation is correct in the sense that it is placed next to another representations of the same class. The R-AUROC can benchmark uncertainties when the classes are unseen during training, but if classes are seen during training, it is highly correlated with a conventional classification AUROC (Kirchhof et al., 2023b). In all experiments we also report the 1-nearest neighbor accuracy (Recall@1) from representation learning to quantify the retrieval performance of the representations and verify the non-interference principle (i) above.

We focus on Vision Transformers (Dosovitskiy et al., 2021) of several sizes and report results for the ViT-Base unless otherwise noted. Their backbone and classifiers were already pretrained by (Steiner et al., 2021) on ImageNet-21k-Winter-2021 (ImageNet-21k-W) (Deng et al., 2009) in `timm` (Wightman, 2019), so we only train the uncertainty module. As we show in Section 4.4, our approach is robust to architecture and optimizer hyperparameters, so we use the default values reported in Appendix A, inter alia a lightweight 2-layer MLP with width 512 for the uncertainty head. We train each model on five seeds and report the median as well as the distance to the maximum or minimum, whichever is larger, to provide an interpretable means to judge the variation.

As datasets, we use ImageNet-21k-W for pretraining and twelve datasets that span a variety of natural image domains for zero-shot transfer. Three are used in the URL benchmark, namely CUB-200-2011 (Wah et al., 2011), CARS196 (Krause et al., 2013), and Stanford Online Products (SOP) (Song et al., 2016). Seven are the natural images datasets of the Visual Task Adaption Benchmark (VTAB) (Zhai et al., 2020), namely Caltech101 (Fei-Fei et al., 2004), Oxford IIIT Pets (Parkhi et al., 2012), CIFAR100 (Krizhevsky, 2009), Scene Understanding 397 (SUN) (Xiao et al., 2010), Oxford Flowers 102 (Nilsback & Zisserman, 2008), Describable Textures (DTD) (Cimpoi et al., 2014), and Street View House Numbers (SVHN) (Netzer et al., 2011). The remaining two are CIFAR10 (Krizhevsky, 2009) and Treeversity#1 (Schmarje et al., 2022).

4.2. Pretrained Uncertainties Generalize

We first test the generalization principle (ii) on the twelve unseen datasets. Figure 1 shows that our pretrained uncertainties generalize well on eleven of the twelve datasets. The best R-AUROCs (Caltech101: 0.758 ± 0.006, Oxford Pets: 0.740 ± 0.008, and CIFAR10: 0.739 ± 0.002) are close to that of the pretraining dataset (0.791 ± 0.001). In Appendix B we find that the zero-shot R-AUROC is higher on datasets that are closer to the domain spanned by ImageNet-21k-W, as one would expect from a pretrained model. This implies that further scaling the pretraining corpus, which is possible with our efficient training, may further benefit performance. The performance also depends on the granulatity of the zero-shot dataset. SVHN for example demands fine-grained house number disambiguation. It is harder to assign a pretrained uncertainty to such specialized tasks without knowing them in advance.

4.3. A New State-of-the-art

How do these results compare to the transfer performances of other methods in the field? The URL benchmark (Kirchhof et al., 2023b) has recently tested the R-AUROC of eleven

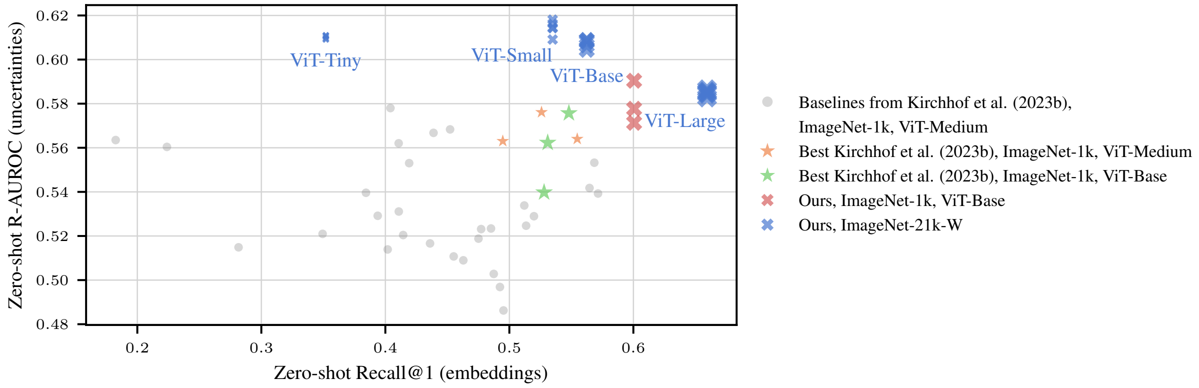
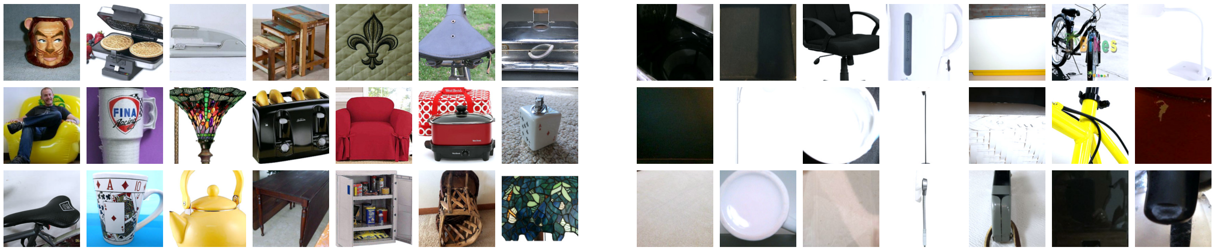


Figure 4. Our pretrained uncertainties outperform the approaches in the URL benchmark (Kirchhof et al., 2023b). The URL benchmark trained ViT-Mediums on ImageNet-1k. We reimplement its best approach (orange) on ViT-Base (green), then enhance it with our changes (red), and finally scale the training of ours to ImageNet-21k with various ViT sizes (blue). Each dot is one seed.



(a) Most certain images ($u(x) \leq 0.04$)

(b) Most uncertain images ($u(x) \geq 0.38$)

Figure 5. Pretrained uncertainties separate clear from ambiguous images on Stanford Online Products, a zero-shot dataset.

approaches on the CUB, CARS, and SOP datasets, averaged. URL tested the methods on ViT-Medium, a niche architecture that does not have ImageNet-21k checkpoints available. Thus, we switch to ViT-Base and reimplement URL's best performing method, the original loss prediction from Section 3.2. We reuse their codebase for compatibility.

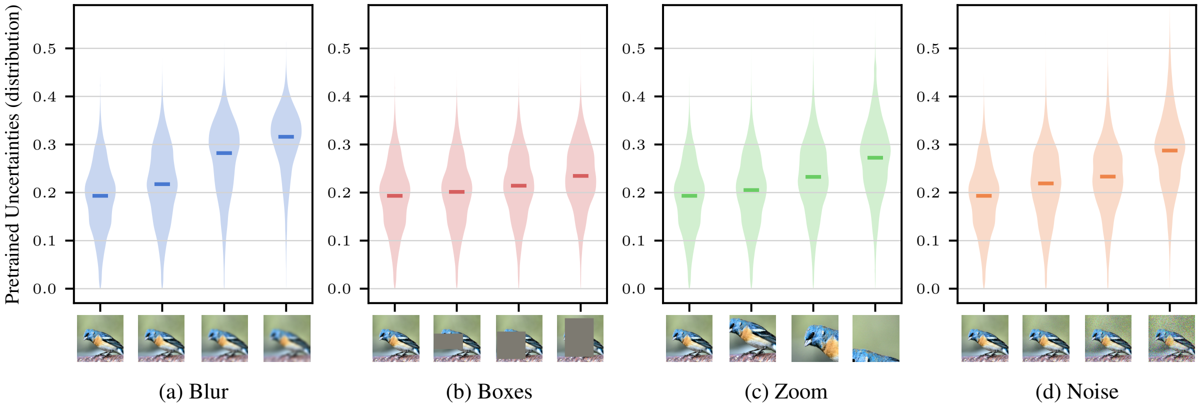
Figure 4 shows the results, both in terms of R-AUROC and Recall@1. First, we find that our ViT-Base loss prediction reimplementation (red stars) achieves comparable performance to the original ViT-Medium backbone (orange stars). We then enhance the original loss prediction with our changes from Section 3.3 (green crosses). We find that the Recall@1 increases by 0.065 ± 0.012 because the backbone is no longer deteriorated by the uncertainty module gradients. The Recall@1 is now constant because we only train the uncertainty head anymore, keeping the pretrained backbone frozen. This does not restrict the R-AUROC of the uncertainty head, it in fact increases slightly by 0.021 ± 0.030. Finally, we scale from ImageNet-1k to ImageNet-21k-W pretraining (blue crosses). This increases the R-AUROC again by 0.028 ± 0.014 on the ViT-Base. Because we now use a different ImageNet-21k-W pretrained checkpoint, its Recall@1 changes. Upon training uncertainty modules for various ViT sizes, we find that they all have higher R-AUROC than the previous state-of-the-art. This demonstrates the generality of our approach.

4.4. Negative Results: Simple Beats Complex

Before we continue, we share some negative results. In Appendix C, we experiment with several techniques to further improve our method, including softening the loss function, uncertainty-induced training data augmentations, architecture changes, optimizer modifications, and initialization schemes. However, none of them significantly improve the uncertainties beyond the method presented in Section 3.3. Thus, to avoid adding unnecessary complexity, we decide to keep our approach as clean and simple as it is.

4.5. Pretrained Uncertainties ≈ Aleatoric Uncertainty

If pretrained uncertainties work on unseen datasets, then which uncertainties do they capture? In this section, we



(a) Blur (b) Boxes (c) Zoom (d) Noise

Figure 6. Pretrained uncertainties grow as images are deteriorated. Distributions and medians over unseen datasets (CUB, CARS, SOP). Note that except zooming, the pretrained model is not exposed to these data augmentations during training.

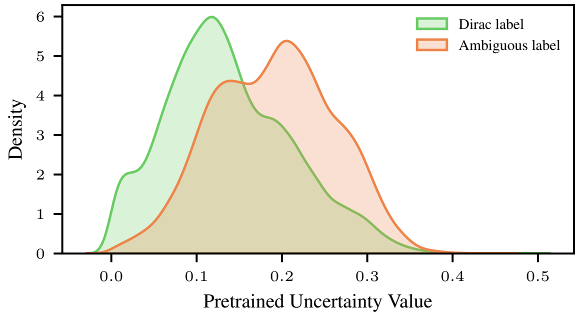


Figure 7. Pretrained uncertainties are systematically higher for ImageNet ReaL-H images with multiple possible labels.

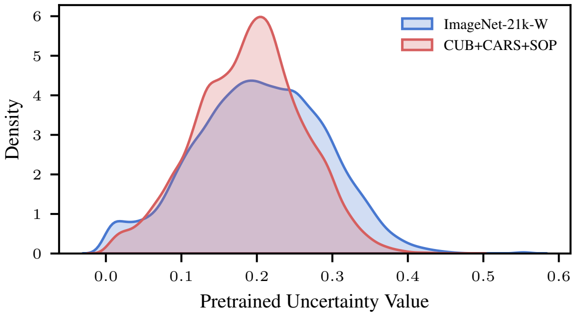


Figure 8. Pretrained uncertainties are consistent between train and unseen datasets, indicating the absence of epistemic uncertainty.

find that they model primarily aleatoric uncertainty and are mostly invariant to epistemic uncertainty. We conduct all analyses on unseen datasets and the ViT-Base model, unless otherwise noted.

To form a working hypothesis, we give some randomly selected examples with low and high predicted uncertainty in Figure 5. Although the network was not trained on this task, eBay product images, it correctly gives low uncertainties to clear and high uncertainties to ambiguous images. Not even a human expert or a Bayes classifier will be able to reduce the ambiguous images' uncertainty to zero, their ambiguity is intrinsic. This is known as aleatoric uncertainty. We hypothesize that pretrained uncertainties represent this form of uncertainty. Below, we investigate this hypothesis.

Human ambiguities. Aleatoric uncertainty is what is left even when an expert makes a prediction, for example a human annotator. So, we compare the model uncertainties to those of human annotators. ImageNet-1k ReaL-H (Beyer et al., 2020) re-collected labels for the 50,000 images in the ImageNet-1k validation set.[2] While clear images kept their original Dirac label, annotators gave multiple or no labels to an image if it was ambiguous. Figure 7 shows that pretrained uncertainties are systematically higher on images the human annotators considered ambiguous (AUROC 0.701). This reinforces the aleatoric uncertainty hypothesis.

Interventional study. Second, we run an interventional experiment to induce aleatoric uncertainty. We deteriorate the images of the unseen datasets by blurring, overlaying with grey boxes, zooming in strongly, and adding Gaussian noise. Except zooming, these transformations were not applied during pretraining. Figure 6 shows that each of these transformations increases the pretrained uncertainties, the more strongly we deteriorate the images. This is additional evidence for the aleatoric uncertainty hypothesis.

No sign of epistemic uncertainty. We now consider the opposite hypothesis: Besides aleatoric uncertainty, do pre-

[2] Our pretraining dataset ImageNet-21k-W covers the classes of ImageNet-1k but neither its validation images nor any soft labels.

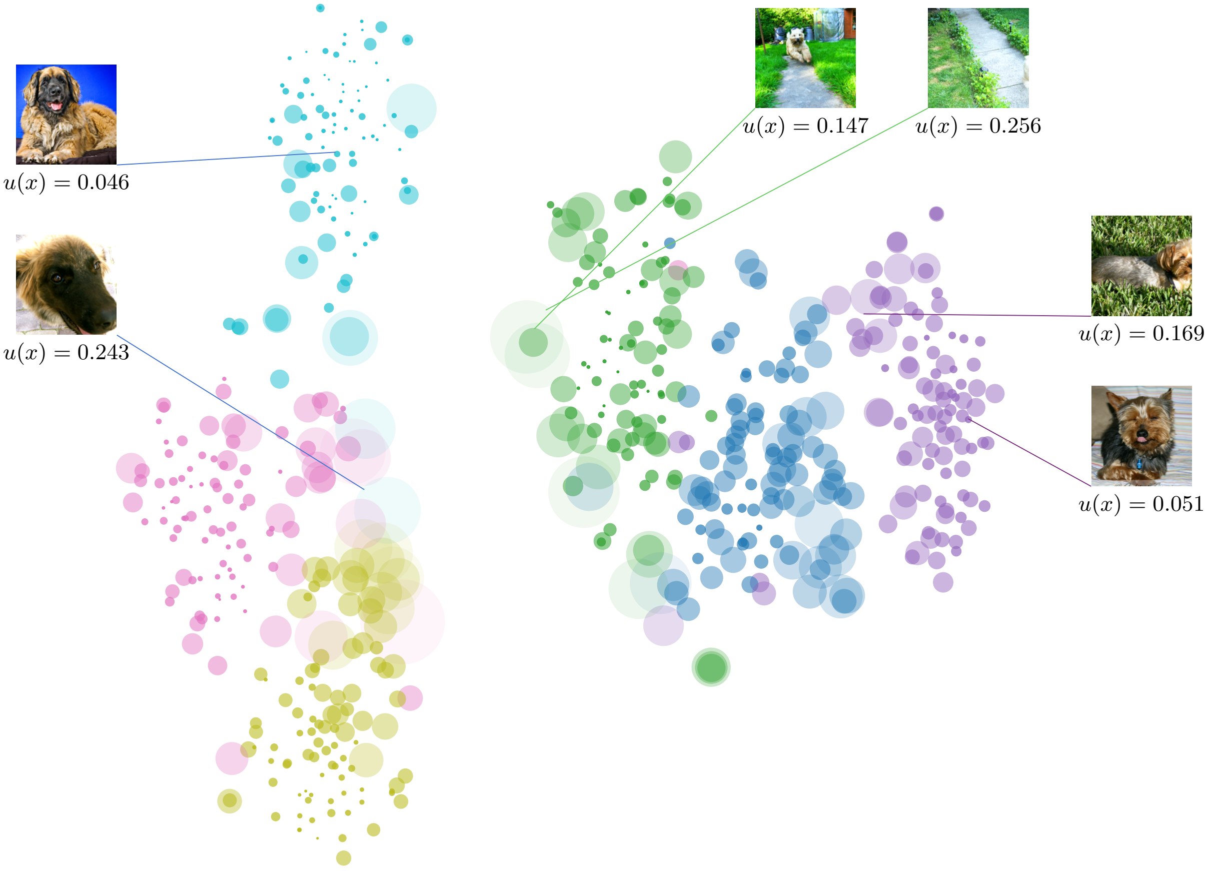


Figure 9. Visualizing pretrained uncertainties makes it easy to identify outliers in a tSNE plot. Images with high uncertainty $u(x)$ are larger and transparent. Six classes of the zero-shot Oxford Pets dataset.

trained uncertainties comprise epistemic uncertainty? This uncertainty arises when a model has not seen an input before. Note that this would be detrimental to a pretrained uncertainty model, since it is intended to be used (exclusively) on unseen datasets where every image would be highly epistemically uncertain, drowning out the aleatoric signal. To test for epistemic uncertainty, we compare the pretrained uncertainties of in-distribution pretraining images to those of out-of-distribution images from unseen datasets. Figure 8 shows that the uncertainties on the pretraining data are similarly distributed to the unseen dataset (pairwise AUROC 0.503 ± 0.004). This suggests they are primarily capturing aleatoric uncertainty.

In summary, our results suggest that pretrained uncertainties quantify the amount of aleatoric uncertainty in an image, both in- and out-of-distribution, without being confounded by epistemic uncertainty. This constitutes significant progress for the ongoing efforts to disentangle epistemic from aleatoric uncertainty (Wimmer et al., 2023; Valdenegro-Toro & Mori, 2022).

5. Application Examples

In this section we showcase two applications that are unlocked by our new pretrained uncertainties.

5.1. Uncertainty-aware tSNE

Pretrained representations are often used to visualize datasets using methods like tSNE (van der Maaten & Hinton, 2008) or UMAP (McInnes et al., 2018). With pretrained uncertainties, we can now communicate inhererent ambiguities and explain outliers in these plots.

Figure 9 shows a tSNE visualization of six dog breeds in Oxford Pets, an unseen dataset. If an image has a high pretrained uncertainty, it is plotted as a larger and increasingly transparent circle. The core region of each class cluster, typified by many small opaque circles, is now visually distinct from border regions and outliers, which are typified by collections of large transparent circles. By inspecting the images that underlie each representation, we verify that the core regions with low uncertainties comprise prototypical images whereas more uncertain images are often cropped out or in a camera angle that makes the exact dog breed ambiguous. We can also see that images lying in other classes' regions are often highly uncertain. Such misclassifications can be prevented by using our uncertainty-enhanced tSNE plots by allowing practitioners to understand and adjust the data preprocessing and filtering.

5.2. Safe Retrieval

This outlier identification can also be automated and utilized to make image retrieval more robust, enabling safe retrieval.

Consider again the Oxford Pets dataset in Figure 9. If we add a new dog image and search for its nearest neighbor, existing next-neighbor retrieval systems (Douze et al., 2024) may match it to an ambiguous image since these tend to lay at border regions. Similarly, if our new image itself is ambiguous, it is likely misplaced and existing systems will match it an arbitrary class. We can utilize pretrained uncertainties to tackle both of these problems by

1. Rejecting queries that are uncertain, and
2. Removing ambiguous images in the existing dataset, making it impossible to match to them.

As an example, we reject and/or clean the 10% most uncertain images per class in Oxford Pets. Table 1 shows that a typical cosine-distance based next-neighbour search achieves a Recall@1 of 0.772 ± 0.000, or in other words a rate of 0.228 ± 0.000 wrong retrievals. Refusing to retrieve images when the input query is uncertain reduces this error rate by 14% to 0.196 ± 0.003 and cleaning the dataset from ambiguous images as potential retrieval partners reduces it by an additional 17%. These improvements are not only observed on unseen dataset but also on data that the retrieval system is familiar with. When using the ImageNet-1k validation set, whose classes were seen during ImageNet-21k-W pretraining (but whose validation images are unseen), deferring ambiguous queries reduces the error rate by 10% and cleaning the database reduces another 10%. All of these improvements are obtained automatically and fully unsupervised - we do not need to know the ground truth label of either the input or the database, since pretrained uncertainties can be computed for any input.

These are only first demonstrations of the opportunities that pretrained uncertainties offer. We anticipate further applications building up on pretrained uncertainties. For example, recent literature proposes retrieving a set of potential neighbours that is close to an ambiguous input with respect to its representation uncertainty (Kirchhof et al., 2023a). This can be implemented with pretrained uncertainty since they give uncertainties about representations. Similarly, conformal prediction (Angelopoulos & Bates, 2022) can view our pretrained uncertainties as a scoring function and calibrate its uncertainty predictions to downstream datasets. To facilitate future research, we provide all pretrained uncertainty checkpoints and code under the link in the abstract.

1-NN error	Oxford Pets	ImageNet-1k
Full datasets	0.228 ± 0.000	0.382 ± 0.000
+ clean queries	0.196 ± 0.003	0.343 ± 0.001
+ clean database	0.163 ± 0.003	0.307 ± 0.001

Table 1. Pretrained uncertainties reduce the error rate of next-neighbour retrieval systems by rejecting ambiguous queries and/or removing ambiguous images from the database.

6. Conclusion

This work introduces a pretrained uncertainty module for computer vision models that is simple, cheap and scalable. We demonstrate its scalability by pretraining on ImageNet-21k-W. Our pretrained uncertainties method gives state-of-the-art zero shot uncertainty estimates on unseen datasets i.e. without finetuning. In future work, we anticipate scaling to even larger pretraining datasets as well as extending the method to pretraining objectives beyond classification and beyond the vision domain. We expect our fixes to the vanilla loss prediction method, that eliminate interference between uncertainty prediction and the main task, to also help in other feed-forward uncertainty quantifiers. By providing pretrained checkpoints, we intend to support applications similar to enhanced visualization and safe retrieval.

Impact Statement

Our uncertainties are intended to capture errors before they happen and reveal uncertainties that would remain undetected when images are solely expressed as representation vectors. This makes models more safe and trustworthy by allowing them to fulfill their tasks with less errors. We enable an easier access to these uncertainties by our ease-of-use principles and providing plug-and-play checkpoints. We see this as a positive impact on both the community and society. As with all general-purpose machine learning advancements, this assumes that a practitioner does not develop a model with a harmful task, which is beyond our sphere of influence. Additionally, we encourage researchers to follow our example of finding ways to significantly reduce training costs for a lower energy consumption during training. We provide our code standalone to help start these efforts.

References

Angelopoulos, A. N. and Bates, S. A gentle introduction to conformal prediction and distribution-free uncertainty quantification. *arXiv preprint arXiv:2107.07511*, 2022.

Beyer, L., Hénaff, O. J., Kolesnikov, A., Zhai, X., and Oord, A. v. d. Are we done with ImageNet? *arXiv preprint arXiv:2006.07159*, 2020.

Chen, X., Liang, C., Huang, D., Real, E., Wang, K., Liu, Y., Pham, H., Dong, X., Luong, T., Hsieh, C.-J., et al. Symbolic discovery of optimization algorithms. *arXiv preprint arXiv:2302.06675*, 2023.

Chun, S. Improved probabilistic image-text representations. *arXiv preprint arXiv:2305.18171*, 2023.

Cimpoi, M., Maji, S., Kokkinos, I., Mohamed, S., and Vedaldi, A. Describing textures in the wild. In *Computer Vision and Pattern Recognition (CVPR)*, 2014.

Collier, M., Jenatton, R., Mustafa, B., Houlsby, N., Berent, J., and Kokiopoulou, E. Massively scaling heteroscedastic classifiers. *arXiv preprint arXiv:2301.12860*, 2023.

Cui, P., Zhang, D., Deng, Z., Dong, Y., and Zhu, J. Learning sample difficulty from pre-trained models for reliable prediction. In *Neural Information Processing Systems (NeurIPS)*, 2023.

Dehghani, M., Djolonga, J., Mustafa, B., Padlewski, P., Heek, J., Gilmer, J., Steiner, A. P., Caron, M., Geirhos, R., Alabdulmohsin, I., et al. Scaling vision transformers to 22 billion parameters. In *International Conference on Machine Learning (ICML)*, 2023.

Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., and Fei-Fei, L. ImageNet: A large-scale hierarchical image database. In *Computer Vision and Pattern Recognition (CVPR)*, 2009.

Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., Uszkoreit, J., and Houlsby, N. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations (ICLR)*, 2021.

Douze, M., Guzhva, A., Deng, C., Johnson, J., Szilvasy, G., Mazaré, P.-E., Lomeli, M., Hosseini, L., and Jégou, H. The Faiss library. *arXiv preprint arXiv:2401.08281*, 2024.

Fei-Fei, L., Fergus, R., and Perona, P. Learning generative visual models from few training examples: An incremental Bayesian approach tested on 101 object categories. *Computer Vision and Pattern Recognition Workshop*, 2004.

Galil, I., Dabbah, M., and El-Yaniv, R. A framework for benchmarking class-out-of-distribution detection and its application to ImageNet. In *The Eleventh International Conference on Learning Representations (ICLR)*, 2023a.

Galil, I., Dabbah, M., and El-Yaniv, R. What can we learn from the selective prediction and uncertainty estimation performance of 523 ImageNet classifiers? In *International Conference on Learning Representations (ICLR)*, 2023b.

Hendrycks, D., Mu, N., Cubuk, E. D., Zoph, B., Gilmer, J., and Lakshminarayanan, B. AugMix: A simple method to improve robustness and uncertainty under data shift. In *International Conference on Learning Representations (ICLR)*, 2020.

Karpukhin, I., Dereka, S., and Kolesnikov, S. Probabilistic embeddings revisited. *arXiv preprint arXiv:2202.06768*, 2022.

Kim, E., Jung, D., Park, S., Kim, S., and Yoon, S. Probabilistic concept bottleneck models. In *International Conference on Machine Learning (ICML)*.

Kingma, D. P. and Ba, J. Adam: A method for stochastic optimization. *International Conference on Learning Representations (ICLR)*, 2015.

Kirchhof, M., Roth, K., Akata, Z., and Kasneci, E. A non-isotropic probabilistic take on proxy-based deep metric learning. In *European Conference on Computer Vision (ECCV)*, 2022.

Kirchhof, M., Kasneci, E., and Oh, S. J. Probabilistic contrastive learning recovers the correct aleatoric uncertainty of ambiguous inputs. *International Conference on Machine Learning (ICML)*, 2023a.

Kirchhof, M., Mucsányi, B., Oh, S. J., and Kasneci, E. Url: A representation learning benchmark for transferable uncertainty estimates. *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks*, 2023b.

Krause, J., Stark, M., Deng, J., and Fei-Fei, L. 3D object representations for fine-grained categorization. In *Conference on Computer Vision and Pattern Recognition (CVPR) Workshop*, 2013.

Krizhevsky, A. Learning multiple layers of features from tiny images. Technical report, 2009.

Lahlou, S., Jain, M., Nekoei, H., Butoi, V. I., Bertin, P., Rector-Brooks, J., Korablyov, M., and Bengio, Y. DEUP: Direct epistemic uncertainty prediction. *Transactions on Machine Learning Research (TMLR)*, 2023. ISSN 2835-8856.

Laves, M.-H., Ihler, S., Fast, J. F., Kahrs, L. A., and Ortmaier, T. Well-calibrated regression uncertainty in medical imaging with deep learning. In *Medical Imaging with Deep Learning*, pp. 393–412. PMLR, 2020.

Liu, J., Lin, Z., Padhy, S., Tran, D., Bedrax Weiss, T., and Lakshminarayanan, B. Simple and principled uncertainty estimation with deterministic deep learning via distance awareness. *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

Loshchilov, I. and Hutter, F. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017.

McInnes, L., Healy, J., Saul, N., and Großberger, L. UMAP: Uniform manifold approximation and projection. *Journal of Open Source Software*, 3(29):861, 2018. doi: 10.21105/joss.00861. URL https://doi.org/10.21105/joss.00861.

Mucsányi, B., Kirchhof, M., Nguyen, E., Rubinstein, A., and Oh, S. J. Trustworthy machine learning, 2023.

Nakamura, H., Okada, M., and Taniguchi, T. Representation uncertainty in self-supervised learning as variational inference. In *International Conference on Computer Vision (ICCV)*, 2023.

Netzer, Y., Wang, T., Coates, A., Bissacco, A., Wu, B., and Ng, A. Y. Reading digits in natural images with unsupervised feature learning. 2011.

Nilsback, M.-E. and Zisserman, A. Automated flower classification over a large number of classes. In *Proceedings of the Indian Conference on Computer Vision, Graphics and Image Processing*, 2008.

Oh, S. J., Gallagher, A. C., Murphy, K. P., Schroff, F., Pan, J., and Roth, J. Modeling uncertainty with hedged instance embeddings. In *International Conference on Learning Representations (ICLR)*, 2019.

Ovadia, Y., Fertig, E., Ren, J., Nado, Z., Sculley, D., Nowozin, S., Dillon, J., Lakshminarayanan, B., and Snoek, J. Can you trust your model's uncertainty? Evaluating predictive uncertainty under dataset shift. *Advances in Neural Information Processing Systems (NeurIPS)*, 32, 2019.

Parkhi, O. M., Vedaldi, A., Zisserman, A., and Jawahar, C. V. Cats and dogs. In *Computer Vision and Pattern Recognition (CVPR)*, 2012.

Postels, J., Segù, M., Sun, T., Sieber, L. D., Van Gool, L., Yu, F., and Tombari, F. On the practicality of deterministic epistemic uncertainty. In *International Conference on Machine Learning (ICML)*, 2022.

Schmarje, L., Grossmann, V., Zelenka, C., Dippel, S., Kiko, R., Oszust, M., Pastell, M., Stracke, J., Valros, A., Volkmann, N., et al. Is one annotation enough? A data-centric image classification benchmark for noisy and ambiguous label estimation. *arXiv preprint arXiv:2207.06214*, 2022.

Song, H. O., Xiang, Y., Jegelka, S., and Savarese, S. Deep metric learning via lifted structured feature embedding. In *Computer Vision and Pattern Recognition (CVPR)*, 2016.

Steiner, A., Kolesnikov, A., , Zhai, X., Wightman, R., Uszkoreit, J., and Beyer, L. How to train your ViT? Data, augmentation, and regularization in vision transformers. *arXiv preprint arXiv:2106.10270*, 2021.

TorchVision. Torchvision: Pytorch's computer vision library. *GitHub repository: https://github.com/pytorch/vision*, 2016.

Tran, D., Liu, J., Dusenberry, M. W., Phan, D., Collier, M., Ren, J., Han, K., Wang, Z., Mariet, Z., Hu, H., et al. Plex: Towards reliability using pretrained large model extensions. *arXiv preprint arXiv:2207.07411*, 2022.

Valdenegro-Toro, M. and Mori, D. S. A deeper look into aleatoric and epistemic uncertainty disentanglement. In *Computer Vision and Pattern Recognition Workshops (CVPRW)*, 2022.

van der Maaten, L. and Hinton, G. Visualizing data using t-SNE. *Journal of Machine Learning Research (JMLR)*, 9(86):2579–2605, 2008.

Wah, C., Branson, S., Welinder, P., Perona, P., and Belongie, S. The Caltech-UCSD birds-200-2011 dataset. Technical Report CNS-TR-2011-001, California Institute of Technology, 2011.

Wightman, R. PyTorch image models, 2019.

Wimmer, L., Sale, Y., Hofman, P., Bischl, B., and Hüllermeier, E. Quantifying aleatoric and epistemic uncertainty in machine learning: Are conditional entropy and mutual information appropriate measures? In *Uncertainty in Artificial Intelligence (UAI)*, 2023.

Xiao, J., Hays, J., Ehinger, K. A., Oliva, A., and Torralba, A. Sun database: Large-scale scene recognition from abbey to zoo. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2010.

Yoo, D. and Kweon, I. S. Learning loss for active learning. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

Yun, S., Han, D., Oh, S. J., Chun, S., Choe, J., and Yoo, Y. Cutmix: Regularization strategy to train strong classifiers with localizable features. In *International Conference on Computer Vision (ICCV)*, 2019.

Zhai, X., Puigcerver, J., Kolesnikov, A., Ruyssen, P., Riquelme, C., Lucic, M., Djolonga, J., Pinto, A. S., Neumann, M., Dosovitskiy, A., Beyer, L., Bachem, O., Tschannen, M., Michalski, M., Bousquet, O., Gelly, S., and Houlsby, N. A large-scale study of representation learning with the visual task adaptation benchmark, 2020.

Zhang, H., Cisse, M., Dauphin, Y. N., and Lopez-Paz, D. Mixup: Beyond empirical risk minimization. In *International Conference on Learning Representations (ICLR)*, 2018.

A. Training details

Architecture. With d_e denoting the dimensionality of the (flattened) embeddings $e(x)$ of each ViT size, our pretrained uncertainty module has the following size across all ViT sizes: `Linear(`d_e`, 512), LeakyReLU (negative slope 0.01), Linear(512, 512), LeakyReLU (negative slope 0.01), Linear(512, 1), Softplus(`β`=1, threshold=20)`. The softplus in the end is to ensure that all uncertainties are strictly positive. This could be dropped since our uncertainties are scale free, but we added it for convenience of interpretation.

Optimizer. We train on ImageNet-21k-W for 460 episodes of 200,000 images, corresponding to roughly seven full epochs. We use a cosine learning rate scheduler that warms up the learning rate from 0.0001 to 0.0028 for 25 episodes and then decays it down to 1e-8 for the remaining episodes. We use an AdamW (Loshchilov & Hutter, 2017) optimizer with $\beta_1 = 0.8$ and $\beta_2 = 0.95$. We apply weight decay of strength 0.0001. These settings are constant for all experiments, without any hyperparameter tuning.

Augmentations. We use the torchvision (TorchVision, 2016) augmentations that `timm` applies by default. Inter alia, all images are cropped to 224x224 pixels. We first apply a `RandomResizedCropAndInterpolation(size=(224, 224), scale=(0.08, 1.0), ratio=(0.75, 1.3333), interpolation=bilinear bicubic)`, and then randomly add `RandomHorizontalFlip(`p`=0.5)` and with $p = 0.4$ a `ColorJitter(brightness=[0.6, 1.4], contrast=[0.6, 1.4], saturation=[0.6, 1.4], hue=None)`.

B. Transfer analysis

In this section, we analyze which datasets our pretrained uncertainties transfer to. We hypothesize that they behave similarly to the classifier head for ImageNet-21k. To test this, we use the entropy of the 21k class predictions of the classifier head as uncertainties and test its R-AUROC. We find that our pretrained uncertainties achieve a similar performance on all datasets. This indicates that pretrained uncertainties works on datasets similar enough to ImageNet-21k that its classifier is also informative. Note that this classifier method is not applicable to provide pretrained uncertainties in practice since it requires maintaining a heavy classifier head (17M parameters for ViT-Base), violating principle (iv), is not scalable to datasets with more classes, violating principle (v), and is not available outside ImageNet-21k classification, violating principle (iii).

Dataset	Pretrained Uncertainties	21k Classifier Entropy
ImageNet-21k	0.791 ± 0.001	0.798
Caltech 101	0.758 ± 0.006	0.808
Oxford Pets	0.740 ± 0.008	0.724
CIFAR 10	0.739 ± 0.002	0.716
CIFAR 100	0.706 ± 0.002	0.696
SUN	0.691 ± 0.002	0.697
Oxford Flowers	0.659 ± 0.009	0.679
Describable Textures	0.649 ± 0.006	0.610
CUB 200	0.626 ± 0.008	0.608
Stanford Online Products	0.607 ± 0.001	0.591
CARS 196	0.589 ± 0.003	0.554
Treeversity	0.560 ± 0.003	0.565
SVHN	0.495 ± 0.005	0.524

Table 2. Pretrained uncertainties perform similarly to using the entropy of the ImageNet-21k classifier head as uncertainty estimate (note that this is impractical due to its size and violating the first principles in Section 3.1). This implies that pretrained uncertainties cover roughly the classes that ImageNet-21k also covers.

C. Simple beats Complex: Negative Results

We test multiple adjustments to our loss, architecture and optimizer in Table 3. We report the average R-AUROC on the unseen datasets CUB, CARS, and SOP as in the URL protocol, evaluated for five seeds on a ViT-Base pretrained on ImageNet-21k-W.

The first change regards the loss function. Currently, when comparing two images, it always requires one image to have a pretrained uncertainty of at least 0.1 larger than the other one. In the formula below, we add an 'approximately equal' category, such that images whose ground-truth loss is within a leeway l are not required to have different loss values:

$$\mathbb{1}_{\mathcal{L}} := \begin{cases} +1 & \text{, if } \mathcal{L}_{\text{task}}^{\text{det.}}(y_1, f(x_1)) > l + \mathcal{L}_{\text{task}}^{\text{det.}}(y_2, f(x_2)) \\ -1 & \text{, if } \mathcal{L}_{\text{task}}^{\text{det.}}(y_1, f(x_1)) + l < \mathcal{L}_{\text{task}}^{\text{det.}}(y_2, f(x_2)) \\ 0 & \text{, else} \end{cases} . \tag{5}$$

However, at several values of the allowed leeway l, this does not change the performance outside the margin of error of the baseline method (0.608 ± 0.004).

Second, we change the size of the uncertainty head MLP. By default it has 2 hidden layers of width 512. We either shrink it to 1 hidden layer with width 256 or enlargen it to 3 hidden layers of width 1024. While there is a slight trend favoring smaller heads, it does not exceed the margin of chance.

Third, we briefly experimented with initializing the uncertainty module with zero-weights. However, this failed to train at all, which is theoretically expected.

Fourth, we add strong augmentations that add different types of aleatoric uncertainty to half of the train dataset. None of these increase the performance, with some even deteriorating it. While this might seem counterintuitive, we presume such artificial sources of uncertainty do not reflect the uncertainties occuring on real images.

Last, we experiment with optimizers other than our default AdamW with cosine learning rate scheduler. While Lion collapses after less than one epoch, SGD performs slightly better than the baseline. However, it might be a false positive, especially taking multiple testing into account. Indeed, the reason we did not select SGD for the main paper is that it did not systematically outperform AdamW during our preliminary experiments on the validation splits with less seeds. The test splits were held strictly secret until the writing of the paper. We suggest future researchers to experiment with replacing their advanced optimizers by SGD.

Method	R-AUROC
Default	0.608 ± 0.004
Softened loss ($l = 0.001$)	0.607 ± 0.006
Softened loss ($l = 0.01$)	0.607 ± 0.005
Softened loss ($l = 0.1$)	0.609 ± 0.005
Smaller uncertainty module	0.611 ± 0.002
Larger uncertainty module	0.606 ± 0.004
Initialize uncertainty module with zero	0.500 ± 0.000
AugMix (Hendrycks et al., 2020) for 50% of train data	0.606 ± 0.005
CutMix (Yun et al., 2019) for 50% of train data	0.575 ± 0.002
MixUp (Zhang et al., 2018) for 50% of train data	0.552 ± 0.009
Blurred images for 50% of train data	0.609 ± 0.003
Small crops for 50% of train data	0.610 ± 0.004
Box overlay for 50% of train data	0.606 ± 0.001
Adam optimizer (Kingma & Ba, 2015)	0.609 ± 0.002
Lion optimizer (Chen et al., 2023)	0.500 ± 0.000
SGD optimizer	0.614 ± 0.005
Step learning rate scheduler	0.607 ± 0.003

Table 3. No change to loss, architecture, optimizer, or data augmentation improves the performance. R-AUROC averaged across CUB, CARS, and SOP as in the URL protocol, for five seeds on a ViT-Base.