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

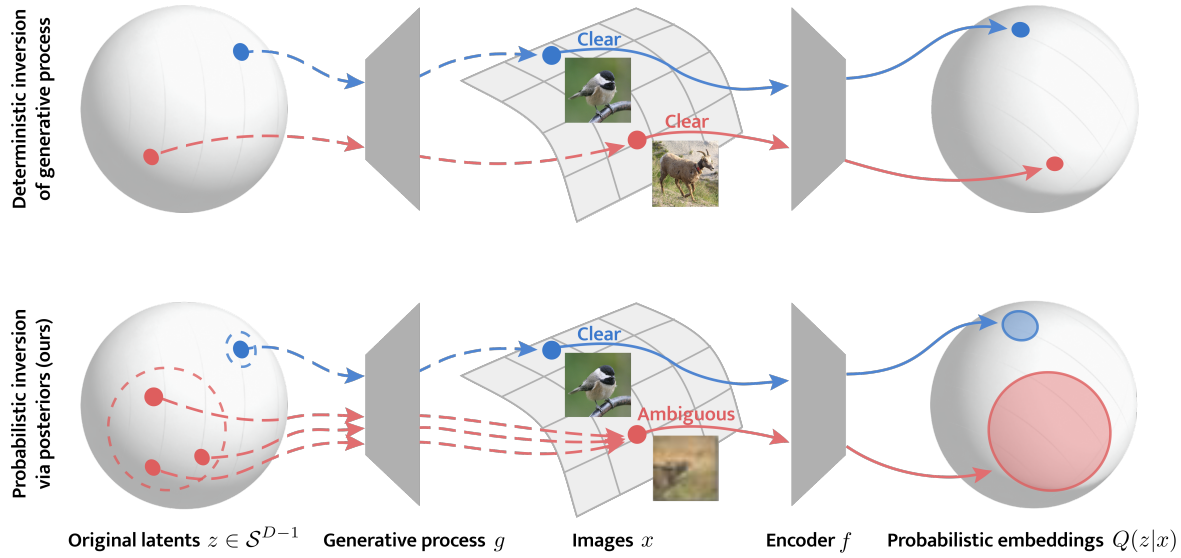

Figure 1. Deteministic encoders embed images to points in the latent space. This recovers the latent vectors that generated them (dashed), up to a rotation (top). However, if an image is ambiguous there are multiple possible latents that could have generated it (bottom). An encoder trained with MCInfoNCE 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 $\text{vMF}(z; \mu(x), \kappa(x))$:

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

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 $P(z|x) = \text{vMF}(z; \mu(x), \kappa(x))$ 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 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	$\hat{\kappa}(x) \to \infty$	

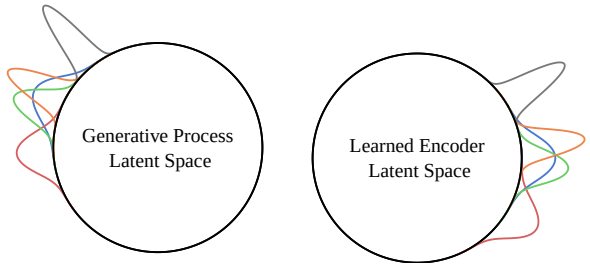

Figure 2. Five posteriors of the generative process and the encoder trained in a run with a 2D latent space. The encoder correctly predicts the posteriors of the generative process, up to a rotation: Rank corr. between $\hat{\mu}(x)$ and the true $\mu(x)$ is 1.00 ± 0.00 (RMSE 0.05 ± 0.00) and that of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

(Li et al., 2021; Shi & Jain, 2019). We sample contrastive training data $(x, x^+, (x_m^-)_{m=1,\dots,M})$ 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.

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

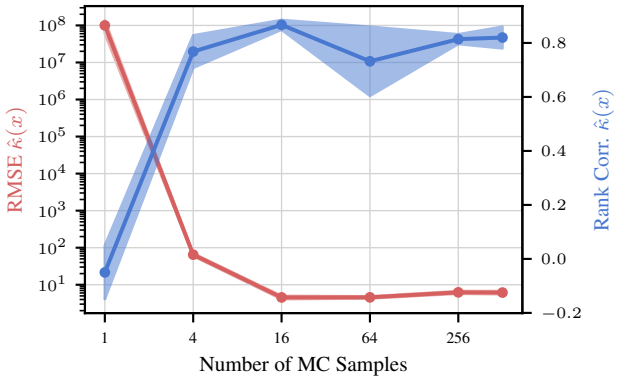

Figure 3. The marginal likelihood approximation bias diminishes with sufficient MC samples. Mean ± std. err. for five seeds.

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 $\pm$ 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

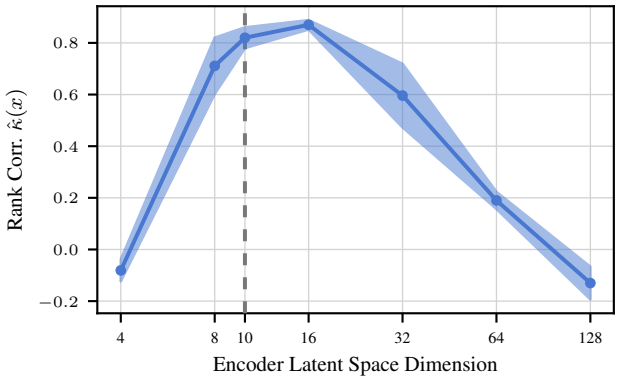

Figure 4. MCInfoNCE learns good $\hat{\kappa}(x)$ even when the encoder latent space dimension mismatches the true generative dimensionality ($D = 10$). Mean $\pm$ std. err. for five seeds.

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

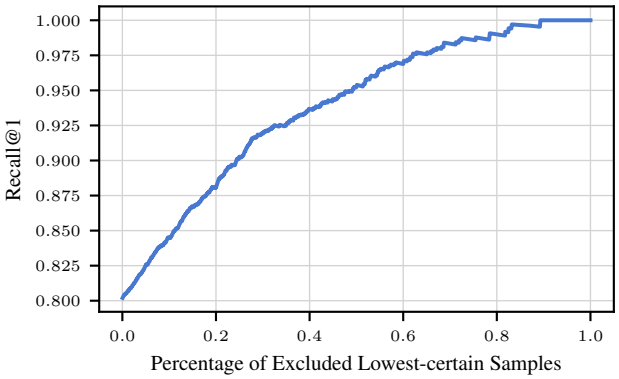

Figure 5. Rejecting images with low certainty values $\hat{\kappa}(x)$ improves the performance on the remaining data monotonically with the threshold. This shows that $\hat{\kappa}(x)$ is predictive of performance.

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

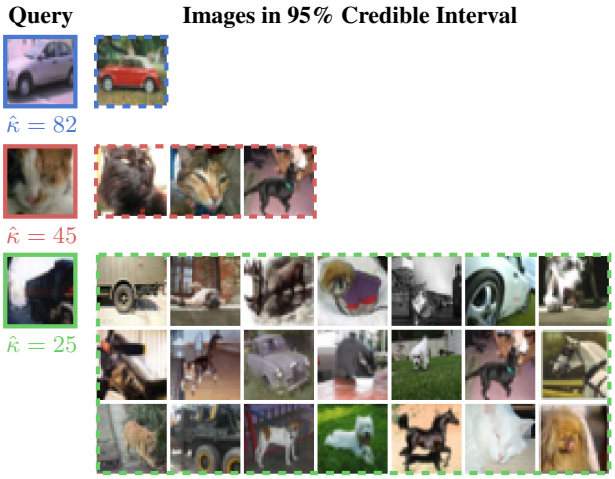

Figure 6. We use an image's posterior to define the credible interval that its latents lie in with a given probability. Clear query images (top) have small credible intervals containing images of the same class as the query. More ambiguous queries (bottom) return larger credible intervals with images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

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.

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 $\text{CI}_p(x) \subset \mathcal{Z}$ contain the true generative latent z of x with a user-defined probability $p \in [0, 1]$, i.e., $P(z \in \text{CI}_p(x)) = p$ 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.

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

$$\prod_{m=1}^{M} Q(z_m^-|x_m^-) \frac{e^{\kappa_{\text{pos}} z^\top z^+}}{\frac{1}{M} e^{\kappa_{\text{pos}} z^\top z^+} + \frac{1}{M} \sum_{m=1}^{M} e^{\kappa_{\text{pos}} z^\top z_m^-}} dz_1^- \dots z_M^- dz^+ dz dx_1^- \dots dx_M^- dx^+ dx \tag{15}$$

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.

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

$$\prod_{m=1}^{M} Q(z_m^-|x_m^-) \frac{e^{\kappa_{\text{pos}} z^\top z^+}}{\frac{1}{M} e^{\kappa_{\text{pos}} z^\top z^+} + \frac{1}{M} \sum_{m=1}^{M} e^{\kappa_{\text{pos}} z^\top z_m^-}} dz_1^- \dots z_M^- dz^+ dz dx_1^- \dots dx_M^- dx^+ dx \tag{17}$$

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

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

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

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

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

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

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

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

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

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

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

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

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:

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

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.

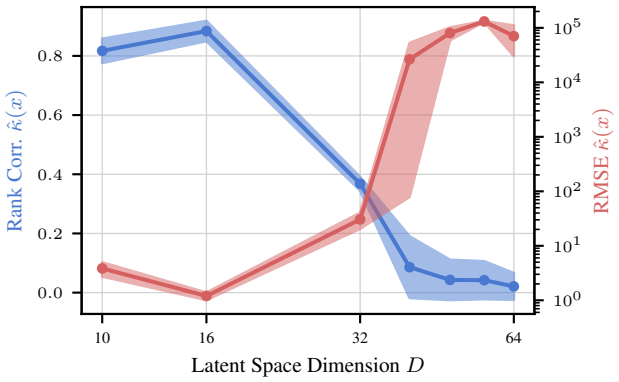

Figure 7. The metrics worsen if the generative process has a latent space of dimension $D \geq 40$. This is likely not due to MCInfoNCE, but a limitation of the contrastive setup of our controlled experiment. Mean $\pm$ std. err. for five seeds.

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 $\pm$ 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.

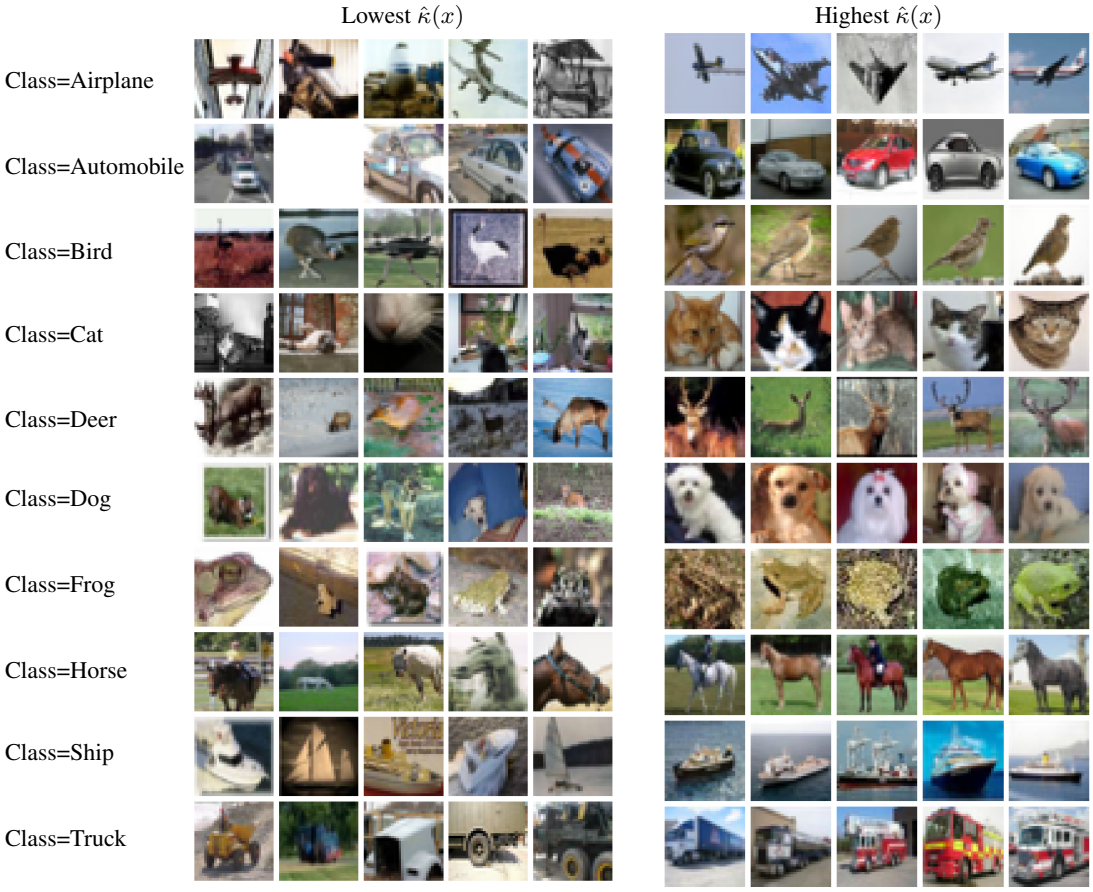

Figure 8. Images for which MCInfoNCE predicts the highest aleatoric uncertainty , i.e., lowest $\hat{\kappa}(x)$, (left) per class qualitatively look more ambiguous than those with the highest predicted $\hat{\kappa}(x)$ (right).