PROBABILISTIC CONTRASTIVE LEARNING WITH EXPLICIT CONCENTRATION ON THE HYPERSPHERE

Anonymous authors
Paper under double-blind review

ABSTRACT

Contrastive learning is predominantly deterministic, limiting its effectiveness in noisy and uncertain environments. We propose a probabilistic contrastive learning framework inspired by the von Mises-Fisher (*vMF*) distribution, embedding representations on a hyperspherical space. To address numerical instability, we introduce an *unnormalized and regularized vMF* distribution, preserving essential properties with theoretical guarantees. The concentration parameter, κ, serves as an interpretable measure of aleatoric uncertainty. Empirical evaluations show a strong correlation between estimated κ and unseen data corruption severity, enabling effective failure analysis and enhancing out-of-distribution detection without modeling epistemic uncertainty. From a fresh perspective, our approach introduces a flexible alignment mechanism for improved uncertainty estimation in high-dimensional spaces while remaining compatible with existing contrastive learning frameworks.

1 INTRODUCTION

Self-supervised contrastive learning has significantly narrowed the gap between unsupervised and supervised learning across various domains, including vision (Chen et al., 2020; Chen & He, 2021; Caron et al., 2021; Zbontar et al., 2021) and multimodal learning (Hager et al., 2023). Despite these notable achievements, current methods still fall short in critical aspects necessary for decision-making in high-stakes applications. In domains such as medical diagnosis (Azizi et al., 2021) and autonomous driving (Kaya et al., 2022), where decisions can have serious consequences, accurately estimating uncertainty, either from data or models, is essential.

Traditional contrastive learning methods are predominantly *deterministic* and lack mechanisms to gauge uncertainty, limiting their utility in scenarios where understanding the model's confidence is crucial. Previous attempts to incorporate uncertainty estimation have primarily utilized Gaussian distributions (Kingma et al., 2015; Gal et al., 2016; Upadhyay et al., 2023), which may not align well with hyperspherical contrastive representations (Bachman et al., 2019; Tian et al., 2020; He et al., 2020; Chen & He, 2021). Recent research has begun exploring geometric properties of contrastive representations (Wang & Isola, 2020; Wang & Liu, 2021; Ge et al., 2023), prompting a shift towards probabilistic models better suited to these spaces.

Probabilistic embedding approaches involve encoders generating distributions within the latent space, rather than deterministic point estimates. These approaches generally fall into two categories: (1) transforming traditional loss functions into probabilistic formats by aggregating the loss across predicted probabilistic embeddings (Scott et al., 2021; Roads & Love, 2021; Kirchhof et al., 2023), and (2) employing distribution-to-distribution metrics to replace point-to-point distances in loss calculations, with the Expected Likelihood Kernel (ELK) (Shi & Jain, 2019) being particularly effective. Recently, a Monte-Carlo sampling-based *InfoNCE* loss (Kirchhof et al., 2023) was proposed to train encoders to predict probabilistic embeddings and learn correct posteriors. Despite these innovations, these approaches face limitations such as numerical instability and implicit uncertainty modeling.

To address these limitations, we leverage the von Mises-Fisher (*vMF*) distribution (Fisher, 1953), which is well-suited for data on the hypersphere and aligns closely with the intrinsic structure of most contrastive learning representations. The *vMF* distribution is parameterized by a mean direction $\boldsymbol{\mu}$ and a concentration parameter κ, where κ controls the spread of the distribution. As shown

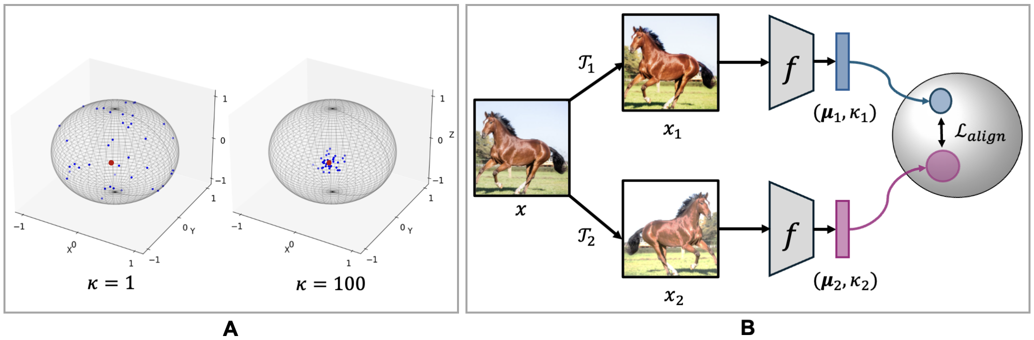

Figure 1: **A.** The *vMF* distribution with a fixed mean vector and varied concentration values κ on a sphere. **B.** Aligning two *vMF* distributions, (μ_1, κ_1) and (μ_2, κ_2) from two different views, is a key challenge in probabilistic contrastive learning, which our method effectively addresses.

in Figure 1(A), higher κ values indicate lower dispersion around the mean, serving as a direct measure of uncertainty. Explicitly learning κ at the sample level is critical for directly quantifying the confidence of the learned representations. A key challenge in probabilistic contrastive learning is the alignment of two *vMF* distributions, as shown in Figure 1(B).

In our work, we present a fresh perspective on modeling probabilistic embeddings by introducing an *unnormalized* and *regularized vMF* distribution, enabling smoother and more stable training. This approach replaces the complex normalization constant of *vMF* with an ℓ_2 regularization, addressing numerical instability issues in high-dimensional spaces and acting as an effective regularizer. Moreover, our method incorporates a probabilistic embedding alignment loss that flexibly adjusts the alignment strength based on embedding dispersion, allowing for both weak and strong alignments depending on uncertainty levels.

Our contributions are as follows: (1) We propose a *vMF*-based probabilistic contrastive learning framework that effectively captures uncertainty in hyperspherical spaces, supported by theoretical guarantees on similarity ranking preservation. (2) We develop a novel embedding alignment loss that accounts for both direction and concentration, providing flexible alignment based on embedding dispersion. This loss is compatible with existing contrastive learning methods. (3) By replacing the normalization constant of the *vMF* distribution with an ℓ_2 regularization term, our approach mitigates numerical instability and acts as a natural regularizer, enhancing training stability and performance. (4) We empirically demonstrate our framework's effectiveness in quantifying degrees of corruption and failure analysis during test time, as well as its potential in enhancing representations for out-of-distribution (OOD) detection.

2 Method

2.1 Preliminaries

Contrastive learning aims to encode semantically similar data points close together and dissimilar points far apart in an embedding space in a *deterministic* manner. A common approach involves creating positive and negative pairs: for a data point $\boldsymbol{x}$, two augmented views $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ are generated. The objective is to maximize the similarity of these positive pairs while minimizing the similarity with other data points (negative pairs). This is formalized using a loss function such as the *SimCLR* framework (Chen et al., 2020), with the contrastive loss defined as:

$$\mathcal{L}_{\text{contrastive}} = -\log \frac{\exp(\text{sim}(\boldsymbol{z}_i, \boldsymbol{z}_j)/\tau)}{\sum_{k=1}^{N} \exp(\text{sim}(\boldsymbol{z}_i, \boldsymbol{z}_k)/\tau)}, \tag{1}$$

where $\boldsymbol{z}_i$ and $\boldsymbol{z}_j$ are the embeddings of $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$, $\text{sim}(\cdot)$ denotes cosine similarity, and τ is a temperature parameter.

von Mises-Fisher distribution (Fisher, 1953) is a probability distribution on the unit sphere in $\mathbb{R}^n$, suitable for modeling data on an n-dimensional hypersphere. Its probability density function is:

$$p(\boldsymbol{x}; \boldsymbol{\mu}, \kappa) = C(\kappa) \exp(\kappa \boldsymbol{\mu}^\top \boldsymbol{x}), \tag{2}$$

where $\boldsymbol{x}$ lies on the unit sphere, $\boldsymbol{\mu}$ is the mean direction, κ is the concentration parameter, and $C(\kappa) = \frac{\kappa^{\frac{n}{2}-1}}{(2\pi)^{\frac{n}{2}} I_{\frac{n}{2}-1}(\kappa)}$ is the normalization constant involving the modified Bessel function $I_\nu(\kappa)$ (Watson, 1922). The concentration parameter κ controls the dispersion around the mean direction; higher κ implies less dispersion.

An **overflow issue** arises in the *vMF* distribution due to the rapid growth of $I_\nu(\kappa)$ with increasing κ, especially in high-dimensional spaces where $\nu = \frac{n}{2} - 1$. This can lead to numerical instability during model training with gradient-based optimization methods (Banerjee et al., 2005).

2.2 UNNORMALIZED AND REGULARIZED *vMF* DISTRIBUTION

Unnormalized and simplified form. To mitigate overflow issues inherent in the traditional *vMF* distribution, we adopt an *unnormalized* form that omits the normalization constant:

$$\psi(\boldsymbol{x}; \boldsymbol{\mu}, \kappa) = \exp(\kappa \boldsymbol{\mu}^\top \boldsymbol{x}). \tag{3}$$

Despite being unnormalized, $\psi(\boldsymbol{x}; \boldsymbol{\mu}, \kappa)$ retains the essential directional and concentration properties, making it suitable for relative comparisons within the loss function used in contrastive learning.

Theoretical guarantee: preserving similarity ranking. To ensure that the unnormalized *vMF* distribution maintains the relative ordering of similarities between embeddings, we present the following proposition:

Proposition 1. *For any two embeddings $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$, if $p(\boldsymbol{x}_1; \boldsymbol{\mu}_1, \kappa_1) > p(\boldsymbol{x}_2; \boldsymbol{\mu}_2, \kappa_2)$, then:*

$$\exp(\kappa_1 \boldsymbol{\mu}_1^\top \boldsymbol{x}_1) > \exp(\kappa_2 \boldsymbol{\mu}_2^\top \boldsymbol{x}_2),$$

thereby preserving the ranking of similarities between embeddings even in the unnormalized form.

Proof outline: Since $C(\kappa)$ is a positive scaling factor dependent solely on κ and the dimensionality d, the relative ordering of $p(\boldsymbol{x}_1; \boldsymbol{\mu}_1, \kappa_1)$ and $p(\boldsymbol{x}_2; \boldsymbol{\mu}_2, \kappa_2)$ is primarily governed by the exponential terms $\exp(\kappa_1 \boldsymbol{\mu}_1^\top \boldsymbol{x}_1)$ and $\exp(\kappa_2 \boldsymbol{\mu}_2^\top \boldsymbol{x}_2)$. By omitting $C(\kappa)$, the unnormalized form $\psi(\boldsymbol{x}; \boldsymbol{\mu}, \kappa)$ retains the relative ordering, ensuring that the ranking of similarities between embeddings is preserved. The complete proof is provided in Appendix A.

Log-likelihood and regularization. Consider the log-likelihood of the unnormalized *vMF* distribution for a data point $\boldsymbol{x}$ on the unit sphere with mean direction $\boldsymbol{\mu}$ and concentration parameter κ:

$$\mathcal{L}(\boldsymbol{\mu}, \kappa) = \log \psi(\boldsymbol{x}; \boldsymbol{\mu}, \kappa) = \kappa \boldsymbol{\mu}^\top \boldsymbol{x}. \tag{4}$$

Since $C(\kappa)$ is omitted, κ can become excessively large during optimization as the original $C(\kappa)$ acts as a natural regularizer. Hence, a new regularization approach is necessary. To address this, we explore two regularization techniques:

(1) **Approximation-based regularization:** Using the large κ approximation of the modified Bessel function $I_\nu(\kappa) \approx \frac{e^\kappa}{\sqrt{2\pi\kappa}}$, we derive a regularizer from the original *vMF* distribution:

$$\mathcal{L}_{\text{reg}_1} = \kappa - \frac{d-1}{2} \log \kappa, \tag{5}$$

where d is the dimension of the embedding space. This regularizer naturally emerges from the log-likelihood of the *vMF* distribution under the approximation.

(2) ℓ_2 **regularization:** Alternatively, we propose a standard ℓ_2 regularizer:

$$\mathcal{L}_{\text{reg}_2} = \lambda \kappa^2, \tag{6}$$

where $\lambda > 0$ is a hyperparameter controlling the regularization strength. While this deviates further from the original likelihood model, it offers smoother, convex gradients.

Empirically, we find that the ℓ_2 regularizer provides more stable training dynamics and superior performance across various tasks. The gradient of this regularizer with respect to κ, given by $\frac{\partial \mathcal{L}_{\text{reg}_2}}{\partial \kappa} = 2\lambda\kappa$, introduces a linear restoring force that grows with κ, effectively preventing it from becoming excessively large during training.

Probabilistic interpretation. From a Bayesian perspective, regularization can be interpreted as placing a prior on the parameter. The ℓ_2 regularization term corresponds to a Gaussian prior on κ:

$$P(\kappa) \propto \exp(-\lambda\kappa^2). \tag{7}$$

This prior assumes that κ is more likely to take smaller values, aligning with the nature of the *vMF* distribution to avoid extreme concentrations.

In summary, this unnormalized and regularized *vMF* distribution serves as the foundation for our probabilistic contrastive learning framework, enabling the model to effectively capture uncertainty.

2.3 Probabilistic Contrastive Learning on the Hypersphere

Unit sphere normalization. To enhance sensitivity to angular differences, we project each data point $\boldsymbol{x}$ onto the unit sphere:

$$\boldsymbol{z} = \frac{f(\boldsymbol{x})}{\|f(\boldsymbol{x})\|}, \tag{8}$$

where $f(\boldsymbol{x})$ is the encoder network's output. This normalization ensures that similarities are based solely on directionality, aligning with the use of cosine similarity in contrastive learning (Chen et al., 2020; Chen & He, 2021; Grill et al., 2020).

Probabilistic embedding alignment. We incorporate the *vMF* distribution into the contrastive learning framework by modeling embeddings with mean directions $\boldsymbol{\mu}$ and concentration parameters κ. Given a batch of input images, two augmented views $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$ are generated. The encoder $f(\cdot)$ outputs $(\boldsymbol{\mu}_1, \kappa_1)$ and $(\boldsymbol{\mu}_2, \kappa_2)$, ensuring $\boldsymbol{\mu}$ is normalized via: $\boldsymbol{\mu} = \frac{\boldsymbol{\mu}'}{\|\boldsymbol{\mu}'\|}$.

In the contrastive learning context, we propose a *probabilistic embedding alignment loss* for the distributions of two augmented views (positive pairs). The alignment of $\boldsymbol{\mu}_2$ given $(\boldsymbol{\mu}_1, \kappa_1)$ and $\boldsymbol{\mu}_1$ given $(\boldsymbol{\mu}_2, \kappa_2)$ can be formulated as:

$$L_{\text{a}}(\boldsymbol{\mu}_1, \kappa_1, \boldsymbol{\mu}_2, \kappa_2) = \exp[(\kappa_1 + \kappa_2) \cdot \boldsymbol{\mu}_1^T \boldsymbol{\mu}_2] \propto \exp(\kappa_1 \cdot \cos(\theta)) \cdot \exp(\kappa_2 \cdot \cos(\theta)), \tag{9}$$

where θ is the angle between $\boldsymbol{\mu}_1$ and $\boldsymbol{\mu}_2$, and $\cos(\theta)$ can be computed as the dot product between $\boldsymbol{\mu}_1$ and $\boldsymbol{\mu}_2$ due to their normalization. The loss is then defined as the negative log-alignment:

$$\mathcal{L}_{\text{align}}(\kappa_1, \kappa_2, \boldsymbol{\mu}_1, \boldsymbol{\mu}_2) = -\lambda_{\text{align}} \cdot (\kappa_1 + \kappa_2)\boldsymbol{\mu}_1^T \boldsymbol{\mu}_2, \tag{10}$$

where λ_{align} controls the strength of the loss. This loss emphasizes the exponential alignment of embeddings based on their dot product, scaled by the sum of their concentration parameters. Unlike the *MC-InfoNCE* loss (Kirchhof et al., 2023), our loss *directly* links the strength of the alignment to the uncertainty of the embeddings, as represented by κ. Intuitively, it encourages tight alignment when uncertainty is low. The analysis of gradient behavior of Eq. 10 is provided in the Appendix.

Final loss. To maintain discriminative embeddings, we combine the *probabilistic embedding alignment* loss, the ℓ_2 regularization, and the original contrastive loss:

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{align}}(\kappa_1, \kappa_2) + \mathcal{L}_{\text{reg}}(\kappa_1, \kappa_2) + \mathcal{L}_{\text{contrastive}}(\boldsymbol{\mu}_1, \boldsymbol{\mu}_2). \tag{11}$$

This combined loss ensures that embeddings are both discriminative and uncertainty-aware, ensuring that the model learns embeddings that are tightly aligned when confident and appropriately dispersed when uncertain.

Connection between κ and uncertainty. κ serves as a key indicator of uncertainty in our framework. High κ values signify tightly clustered embeddings, indicating low aleatoric uncertainty, while low κ reflects dispersed embeddings, corresponding to higher aleatoric uncertainty arising from data noise or corruption. Moreover, κ also captures epistemic uncertainty: in regions with limited data, hard samples, or OOD inputs, lower κ values represent increased uncertainty, reflecting the model's lack of confidence in its learned representations.

Algorithm 1 Probabilistic Contrastive Learning with Embedding Alignment

```
for each batch $\{x^i\}_{i=1}^N$ do
    $\{(x_1^i, x_2^i)\}_{i=1}^N \leftarrow$ Augment($\{x^i\}_{i=1}^N$) % Generate positive pairs
    for each positive pair $(x_1, x_2)$ do
        $(\mu_1, \kappa_1), (\mu_2, \kappa_2) \leftarrow f(x_1), f(x_2)$ % Obtain embeddings and concentrations
        $\kappa_1, \kappa_2 \leftarrow$ Softplus($\kappa_1, \kappa_2$) % Ensure positive $\kappa$ values
        $\mathcal{L}_{align} \leftarrow -\lambda_{align}(\kappa_1 + \kappa_2)\mu_1^\top \mu_2$ % Compute alignment loss
        $\mathcal{L}_{contrastive} \leftarrow \mathcal{L}_{contrastive}(\mu_1, \mu_2)$ % Compute contrastive loss
        $\mathcal{L}_{reg} \leftarrow \lambda_\kappa(\kappa_1^2 + \kappa_2^2)$ % Compute regularization loss
        $\mathcal{L}_{total} \leftarrow \mathcal{L}_{align} + \mathcal{L}_{contrastive} + \mathcal{L}_{reg}$ % Combine losses
        Backpropagate and update model parameters
    end for
end for
```

This dual role of κ is intricately tied to data augmentation in contrastive learning, which introduces variability and perturbations to the training data. As a result, the embedding dispersion varies: high-quality, less perturbed data maintain high κ values, while heavily augmented or noisy data lead to lower κ values. During training, the model adjusts κ based on the consistency and quality of augmented data, enabling it to adapt to diverse and uncertain input regions.

3 Related Work

Representation learning on the unit hypersphere has its advantages in representation quality and interpretability (Nickel & Kiela, 2017; Davidson et al., 2018; Govindarajan et al., 2023). Theoretical analysis has shown that such methods learn alignment and uniformity properties asymptotically on the hypersphere (Wang & Isola, 2020). It has been therefore widely adopted by the popular contrastive learning approaches (Bachman et al., 2019; Tian et al., 2020; He et al., 2020; Chen & He, 2021). Hyperspherical latent spaces in variational autoencoders have demonstrated superior performance over Euclidean counterparts (Davidson et al., 2018; Xu & Durrett, 2018).

Hyperspherical face embeddings have outperformed their unnormalized counterparts (Liu et al., 2017; Wang et al., 2017). Recently, contrastive learning on the hypersphere has been shown effective in out-of-distribution detection (Ming et al., 2022). The consistent empirical success across diverse applications and nice geometric properties underscores the hypersphere's uniqueness as a feature space. In the context of our work, we extend this exploration to the realm of uncertainty estimation within these hyperspherical spaces.

Aleatoric uncertainty is inherent in many vision problems, such as object recognition (Kendall & Gal, 2017; Shi & Jain, 2019) and semantic segmentation (Monteiro et al., 2020; Kahl et al., 2024), where stochasticity in image acquisition (*e.g.,* noise and imaging artifacts) incurs uncertainties in prediction. Other tasks with ambiguous input data include 3D reconstruction from 2D input (Chen et al., 2021) or from noisy sensor (Meech & Stanley-Marbell, 2021).

To facilitate the systematic study of aleatoric uncertainty, the widely-applied benchmark proposed by Hendrycks & Dietterich (2019) quantifies the severity of data corruptions (*e.g.,* imaging noise, distortions caused by compression, etc.) into different corruption levels (Hendrycks & Dietterich, 2019). In this work, we demonstrate that the estimated concentration parameters κ's closely *correlate* with the corruption levels.

Probabilistic embedding are emerging approaches that involve encoders generating distributions within the latent space, rather than deterministic point estimates. Such approaches to probabilistic embeddings diverge into two primary categories: The first method transforms traditional loss functions into probabilistic formats by aggregating the entire loss across the spectrum of predicted probabilistic embeddings (Scott et al., 2021; Roads & Love, 2021; Kirchhof et al., 2023). Another strategy employs distribution-to-distribution metrics to substitute the conventional point-to-point distances in loss calculations, with the Expected Likelihood Kernel (Shi & Jain, 2019) standing out as a particularly effective technique. Notably, it has recently shown its efficacy even in contexts involving high-dimensional embedding spaces (Kirchhof et al., 2022). Recently, a Monte-Carlo sampling-based *InfoNCE* loss (Kirchhof et al., 2023) was proposed to train the encoder to predict probabilistic embeddings and to learn the correct posteriors. In our work, we present a fresh perspective on modeling such a probabilistic embedding by introducing the unnormalized *vMF* and a regularization term, enabling a smoother training.

4 Experiments and Results

4.1 Experimental Setup

Quantifying the level of data corruption. CIFAR-10-C (Hendrycks & Dietterich, 2018) is a well-established benchmark dataset for evaluating model robustness in a controlled environment. It contains 18 image corruption types based on the original CIFAR-10 (Krizhevsky et al., 2009). Our key assumption is that the corrupted data have higher inherent aleatoric uncertainty compared to the uncorrupted one. We therefore assume that higher degrees of corruption would result in higher uncertainties (lower concentration κ). We use *Spearman Correlation* as an evaluation metric to quantify if a model could capture this connection. We use the non-parametric, ranking-based Spearman correlation (rather than Pearson) as the relationship between the variables is highly nonlinear (i.e., we test for their monotonicity). Some corruptions are shown in Figure 2. The details of the corruption are in Appendix B

OOD detection. From CIFAR-10, CIFAR-100, and MNIST (LeCun, 1998), we generate six in-domain and out-of-domain pairs for the OOD detection tasks, as shown in Table 2. Area Under the Receiver Operating Characteristic curve (AUROC) is used for the detection accuracy following the practice from (Kuan & Mueller, 2022). For this task, we train three different models on the three domains from scratch. The learned κ is treated as a one-dimensional feature (or anomaly score) to enhance the features for OOD detection.

Failure analysis. To evaluate the effectiveness of our uncertainty estimates, we perform a three-step failure analysis. First, we pre-train the probabilistic encoder using a contrastive learning approach. Second, we train a linear classifier on the learned embeddings (mean directions) with labels from the training set and assess its accuracy on the test set. Third, we categorize the test samples into two groups: (1) correctly classified and (2) misclassified. By comparing the κ values between these groups, we investigate whether lower κ values are associated with misclassifications, thereby demonstrating the utility of our uncertainty measures in identifying uncertain predictions.

4.2 Baselines

To quantify the uncertainty in representations, we compare our method with the following baselines which are briefly described.

Model ensembles (Huang et al., 2016). We train multiple deterministic models with different initializations and evaluate the empirical variance in their representations. The variance across the ensemble serves as a measure of uncertainty, where high variance indicates lower confidence in embeddings.

MC dropout (Gal & Ghahramani, 2016). This approach quantifies uncertainty by enabling dropout during inference and performing multiple forward passes through the network. The variance of the predictions from these passes estimates the uncertainty, with higher variance reflecting greater uncertainty.

Differential Entropy (Malinin & Gales, 2018) (DE). This baseline measures the entropy of the continuous probability distributions of the embeddings. Higher entropy values indicate greater uncertainty in the model's representations.

Expected likelihood kernel (Shi & Jain, 2019) (ELK). ELK replaces traditional point-to-point distances with distribution-to-distribution metrics. This probabilistic approach measures similarity between embeddings based on their underlying distributions, enhancing uncertainty estimation in contrastive learning frameworks.

Hedged instance embeddings (Oh et al., 2018) (HIB). HIB models embeddings as random variables trained under the variational information bottleneck principle. By hedging the location of each

Table 1: **Spearman correlation between kappa values and the levels of corruption**. As the severity of corruption increases, κ decreases, implying higher uncertainty in the representations. $+$ and $-$ indicate that the correlations are expected to be *positive* and *negative*, respectively. We use the ranking-based Spearman correlation rather than Pearson as the relationship between the variables is highly nonlinear (monotonic).

Methods	Brightness	Contrast	Defocus Blur	Elastic Transform	Fog	Frost	Gaussian Blur	Gaussian Noise	Glass Blur
Model ensembles $(+)$	-0.829	-0.943	-0.486	-0.829	-0.943	-1.000	-0.657	-1.000	-0.714
MC dropout $(+)$	-1.000	-1.000	-0.600	-0.943	-1.000	-1.000	-0.829	-1.000	-0.486
DE (Malinin & Gales, 2018) $(+)$	-0.829	-0.943	-0.486	-0.829	-0.943	-0.943	-0.657	-0.943	-0.714
ELK (Shi & Jain, 2019) $(-)$	-0.714	-0.829	-0.371	-0.657	-0.943	-0.714	-0.657	-0.714	-0.714
HIB (Oh et al., 2018) $(-)$	-0.829	-0.943	-0.371	-0.829	-0.943	-0.714	-0.657	-0.714	-0.714
MCInfoNCE $(-)$	$\mathbf{-1.000}$	$\mathbf{-1.000}$	$\mathbf{-0.429}$	$\mathbf{-0.943}$	$\mathbf{-1.000}$	$\mathbf{-1.000}$	-0.486	$\mathbf{-1.000}$	-0.714
Ours $(-)$	$\mathbf{-1.000}$	$\mathbf{-1.000}$	$\mathbf{-0.429}$	$\mathbf{-0.943}$	$\mathbf{-1.000}$	$\mathbf{-1.000}$	$\mathbf{-0.771}$	-0.600	$\mathbf{-0.771}$
	Impulse Noise	JPEG Comp.	Motion Blur	Pixelate	Saturate	Snow	Spatter	Speckle Noise	Zoom Blur
Model ensembles $(+)$	-1.000	-1.000	-0.943	-1.000	-0.371	-0.829	-0.829	-1.000	-0.522
MC dropout $(+)$	-1.000	-1.000	-1.000	-1.000	-0.543	-0.829	-0.829	-1.000	-0.714
DE (Malinin & Gales, 2018) $(+)$	-0.829	-0.943	-0.943	-0.943	-0.522	-0.829	-0.657	-0.910	-0.657
ELK (Shi & Jain, 2019) $(-)$	-0.486	-0.943	-0.943	-0.829	-0.486	-0.829	$\mathbf{-0.829}$	-0.829	-0.543
HIB (Oh et al., 2018) $(-)$	-0.829	-0.829	-0.829	$\mathbf{-0.943}$	-0.371	-0.657	-0.714	-0.829	-0.486
MCInfoNCE $(-)$	$\mathbf{-0.943}$	$\mathbf{-1.000}$	$\mathbf{-0.943}$	-0.829	-0.371	-0.483	$\mathbf{-0.829}$	$\mathbf{-1.000}$	0.600
Ours $(-)$	$\mathbf{-0.943}$	$\mathbf{-1.000}$	$\mathbf{-0.943}$	$\mathbf{-0.943}$	$\mathbf{-0.714}$	$\mathbf{-0.943}$	$\mathbf{-0.829}$	$\mathbf{-1.000}$	$\mathbf{-0.943}$

input in the embedding space, HIB explicitly captures uncertainty arising from ambiguous inputs, enhancing performance in image matching and classification tasks.

MCInfoNCE (Kirchhof et al., 2023). We adapt the Monte Carlo sampling-based *InfoNCE* loss to the *SimCLR* framework for a fair comparison. This method leverages sampling to approximate expectations over the latent space, facilitating uncertainty estimation while maintaining compatibility with contrastive learning objectives.

4.3 Training

Architecture. The encoder network contains two projection heads for the mean direction $\boldsymbol{\mu}$ and κ based on *ResNet50* (He et al., 2016). The projection head for the $\boldsymbol{\mu}$ is realized through a sequential arrangement of layers, starting with a linear transformation from the 2048-dimensional *ResNet50* feature space to an intermediate 512-dimensional space, followed by batch normalization and *ReLU* activation, and finally projecting down to a d-dimensional representation. d is set to 128 for all experiments except the study on dimension in Table 4.4. In parallel, the κ parameter is estimated through a separate head, mirroring the structure but diverging in its final output to produce a single scalar value per input. κ is then passed through a *softplus* function (Nair & Hinton, 2010), ensuring its non-negativity and adherence to the constraints of a concentration parameter in a probabilistic setting. The codes of the neural architecture are in the Appendix.

Optimization. Following the *SimCLR* configuration, our data augmentation includes random cropping, resizing, color jittering, and horizontal flipping. We train all models on CIFAR-10's training set for 1000 epochs to quantify corruption levels. Hyper-parameters λ_{align} and λ_{reg} are adjusted for optimal training loss and stability, with λ_{reg} fixed at 0.005 across experiments due to observed training stability. The λ_{align} parameter, dictating alignment loss strength, inversely affects representation discriminativeness and, if increased, may cause training instability. A practical approach involves starting with a low value, like 0.01, and incrementally adjusting up to a saturation point where the total training loss stabilizes; here, λ_{align} is set to 0.05 for *SimCLR*. We provide an ablation study of λ_{reg} and λ_{align} in Appendix E. Training codes are in the Appendix.

4.4 Results

κ captures fine-grained aleatoric uncertainty. We validate our framework on CIFAR-10-C (Hendrycks & Dietterich, 2019), focusing on κ correlates with varying levels of data corruption, providing a probabilistic interpretation of uncertainty in contrastive learning. Table 1 shows *Spearman* correlation coefficients between κ and different corruption types. Our method shows strong correlations, especially for brightness, contrast, and defocus blur, surpassing model ensembles and

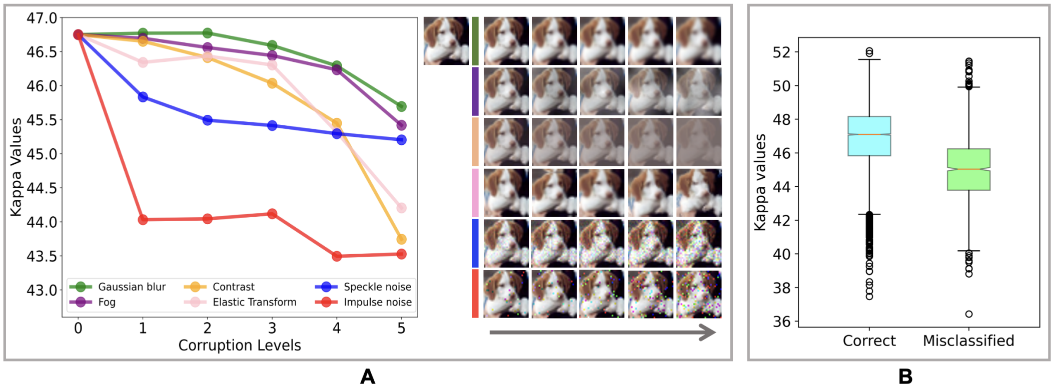

Figure 2: **A.** Decreasing κ implies less concentration and therefore more uncertainty in the representation (*left*). The associated image corruption is from mild to severe (*right*). **B.** The two groups of kappa values (i.e., correctly classified and misclassified) from the test set are significantly different.

Table 2: **AUROC scores for OOD detection.** F_{res} refers to using a k-NN classifier (k=5) based on ResNet-18 features. $F_{\text{res}}+\kappa$ denotes the enhancement through the concatenation of κ with the original features.

In-domain	OOD	F_{res}	κ	$F_{\text{res}}+\kappa$
CIFAR-10	CIFAR-100	0.9658	0.8162	**0.9677**
CIFAR-10	MNIST	0.9929	0.6783	**0.9937**
CIFAR-100	CIFAR-10	0.8653	0.6312	**0.8794**
CIFAR-100	MNIST	0.9769	0.9390	**0.9774**
MNIST	CIFAR-10	0.9993	0.9979	**0.9999**
MNIST	CIFAR-100	0.9998	0.9951	**1.0000**

Table 3: **Extension to other contrastive learning methods**. 'Correlation' refers to the average of Spearman correlations in Tab. 1.

Methods	SimCLR	SimSiam	BYOL	SwaV
Correlation	-0.883	-0.846	-0.835	-0.865

Table 4: **The effect of embedding dimensions** with fixed λ_{align} and λ_{κ}. 'Correlation' refers to the average of Spearman correlations in Tab. 1.

Dimension	64	128	256	384
Correlation	-0.768	-0.883	-0.844	-0.901

MC dropout, which fail to capture fine-grained uncertainty for most corruptions. In contrast, model ensembles, MC dropout, and DE exhibit unexpected negative correlations, failing to capture increasing uncertainty under corruption.

By comparing *MC-InfoNCE* and our method, we observe that *MC-InfoNCE* achieves general good-quality estimation but fails to quantify semantics-related corruptions (such as Gaussian blur and Zoom blur). The formulation of *MC-InfoNCE* enforces the κ for the positive pair to be identical. HIB (Oh et al., 2018), although effective in managing ambiguous inputs, is less responsive to severe noise-based distortions. In contrast, our method learns a data-dependent κ that adapts dynamically to new corruptions, providing more reliable uncertainty estimates across diverse scenarios. This highlights the robustness of our approach in quantifying aleatoric uncertainty compared to traditional ensemble-based methods. Figure 2(A) visually demonstrates this adaptability, highlighting how our framework enhances uncertainty estimation as corruption intensifies.

κ enables failure analysis. To empirically validate the model's potential in failure analysis, we analyzed the outcome of the CIFAR-10 test set, which includes 10,000 samples. We divided the predictions into two groups: correctly classified (8,554 κ values) and misclassified (1,446 κ values). The distribution of the two groups is shown in Figure 2(B). Through bootstrapping (50 iterations, each with 100 randomly sampled observations) and applying the Mann-Whitney U test, we sought to robustly compare κ values between the two groups. Our analysis yielded p-values ranging from 6.42×10^{-20} to 1.15×10^{-6}, which strongly suggests a meaningful difference in κ values between correctly and incorrectly classified samples, indicating the model's potential in failure detection within practical settings.

κ enhances OOD detection. Since κ captures inherent characteristics of the data, it may manifest as epistemic uncertainty. The efficacy of κ as a self-supervised image feature to *enhance* OOD detection methods is evident from the results presented in Table 2, showcasing consistently superior

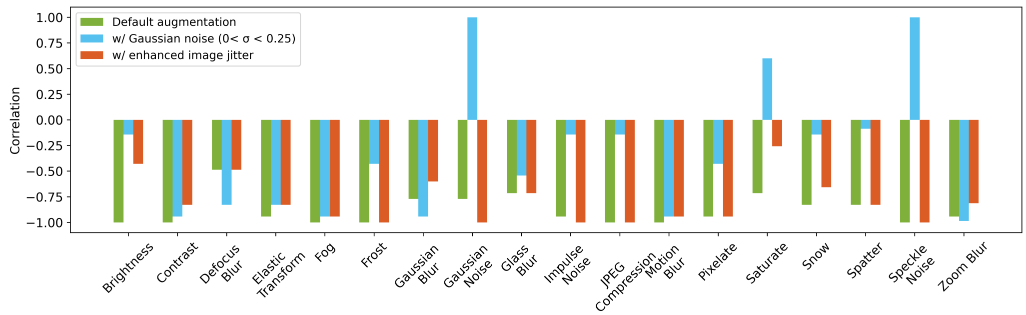

Figure 3: **Additional augmentation degrades the quality of uncertainty estimation for specific types of corruptions**. For instance, introducing Gaussian noise during training causes the correlation with both Gaussian and Speckle noise to shift from negative to positive.

AUROC values by a simple concatenation with existing features. When compared against *ResNet* feature-based baselines derived from supervised learning approaches as discussed in (Ming et al., 2022) – the addition of κ consistently enhances performance. This improvement highlights κ's capacity to capture aleatoric uncertainty that varied between dataset distributions, thereby validating its utility in strengthening OOD detection methods.

κ partially reflects internal augmentations. It is known that internal data augmentation during training enables models to learn invariance to those augmentations. Yet, how the concentration parameter κ reacts to such augmentations, remains unexplored. We add two types of data augmentations one at a time to test the response of κ. Initially, as evidenced by the ■ bars in Figure 3, the default data augmentations do not weaken the sensitivity of κ. Further introducing Gaussian noise (σ<0.25) into the data augmentation pipeline allows the model to adjust effectively, making κ less sensitive to both Gaussian and speckle noise, as indicated by the ■ bars. Furthermore, despite the default augmentation regime, enhancing the image color jittering including brightness (0.3→0.4), contrast (0.3→0.4), saturation (0.3→0.4), and hue (p = 0.2 → p = 0.3), κ continues to be reactive to these changes. However, intensifying these augmentations leads to significant shifts in the correlations associated with brightness and similar aspects, highlighted by the ■ bars. This suggests the existence of a 'saturation point,' beyond which further augmentation fails to meaningfully influence κ's assessment of uncertainty. Consequently, to preserve κ's efficacy in uncertainty quantification, our framework advises against the use of overly strong augmentations.

Integrating uncertainty without losing much discriminativeness. Our framework not only models aleatoric uncertainty but also maintains the discriminativeness inherent in contrastive learning models. An analysis depicted on the left panel of Figure 4 compares the top-1 classification accuracy on the CIFAR-10 test set and the quality of uncertainty estimation across 1000 training epochs. Despite a modest performance decrease (2%) compared to the deterministic approach, our method exhibits training stability and surpasses the accuracy of the MC sampling-based method (Kirchhof et al., 2023), demonstrating our model's effectiveness. Furthermore, the right panel of Figure 4 showcases the consistent performance of our framework in uncertainty estimation. Notably, even in the early stage of training (at the epoch of 200), our model provides high-quality uncertainty estimations.

Adaptability to different methods and dimensions. We adapt our framework to other established contrastive learning methods such as *SimSiam* (Chen et al., 2020), *BYOL* (Grill et al., 2020), and *SwaV* (Caron et al., 2020)), in a manner of adapting to *SimCLR*. Table 3 demonstrates our framework's versatility, particularly with *SimSiam* and *BYOL*, which train using only positive pairs. As shown in Table 4, the compatibility of our framework different dimensions of the embedding space further attests to its adaptability. More discussions on the results in Appendix D.

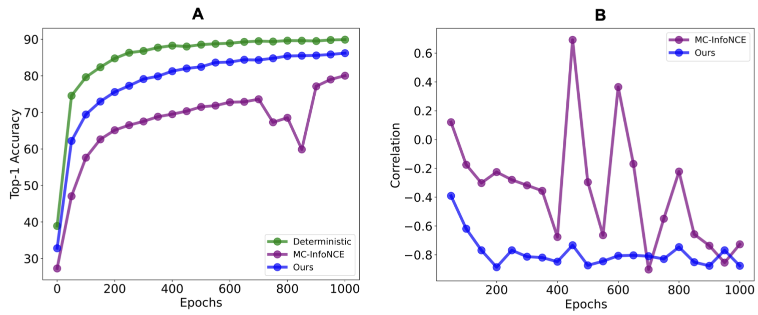

Figure 4: **A.** Comparison of top-1 classification accuracy on the downstream task over the 1000 training epochs. The deterministic approach represents the original *SimCLR* approach that learns a one-to-one mapping from an image to a representation. **B.** Comparison of correlation between κ and levels of data corruption (i.e., uncertainty estimation quality) over the 1000 training epochs.

5 Discussion

Our study demonstrates the efficacy of the concentration parameter κ in uncertainty estimation, failure analysis, and OOD detection within contrastive learning frameworks. Empirical results show that κ effectively captures aleatoric uncertainty by quantifying the dispersion of embeddings in the *vMF* distribution. Additionally, κ indirectly captures epistemic uncertainty by exhibiting greater variability for OOD samples and failure cases. Theoretically, we show that the unnormalized *vMF* distribution preserves the ranking of similarities between embeddings, which is critical for contrastive learning. By introducing an alignment loss that leverages the concentration parameter κ, we offer a flexible mechanism that adapts alignment strength based on uncertainty.

However, our study is limited to small-scale datasets such as CIFAR-10-C, and has not yet been evaluated on larger, more complex datasets like ImageNet. Scaling κ and the alignment mechanism to handle these environments remains a challenge (Zhang et al., 2023; Lu et al., 2024), which we aim to address in future work.

Future research will explore integrating κ with other OOD detection methods and extending its application to domains such as healthcare. Additionally, investigating alternative approaches to managing the normalization constant $C_n(\kappa)$, and extending our framework to non-contrastive methods like MAE (He et al., 2022), multi-modal settings, and higher-dimensional data types, represent promising avenues for further development.

Potential broader impact. Integrating uncertainty estimation into contrastive learning has significant implications for critical applications such as autonomous driving and medical diagnosis. Our framework supports the development of transparent and accountable AI systems (Kim & Doshi-Velez, 2021), enhancing decision-making by providing interpretable confidence levels. Improved uncertainty estimation mitigates risks in high-stakes environments by alerting users to low-confidence predictions, thereby fostering trust and reliability. Future work will focus on applying this method to other tasks, including classification and segmentation across various domains, further promoting robustness and reliability in AI systems.

References

Shekoofeh Azizi, Basil Mustafa, Fiona Ryan, Zachary Beaver, Jan Freyberg, Jonathan Deaton, Aaron Loh, Alan Karthikesalingam, Simon Kornblith, Ting Chen, et al. Big self-supervised models advance medical image classification. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 3478–3488, 2021.

Philip Bachman, R Devon Hjelm, and William Buchwalter. Learning representations by maximizing mutual information across views. *Advances in neural information processing systems*, 32, 2019.

Arindam Banerjee, Inderjit S Dhillon, Joydeep Ghosh, Suvrit Sra, and Greg Ridgeway. Clustering on the unit hypersphere using von mises-fisher distributions. *Journal of Machine Learning Research*, 6(9), 2005.

Mathilde Caron, Ishan Misra, Julien Mairal, Priya Goyal, Piotr Bojanowski, and Armand Joulin. Unsupervised learning of visual features by contrasting cluster assignments. *Advances in neural information processing systems*, 33:9912–9924, 2020.

Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 9650–9660, 2021.

Hansheng Chen, Yuyao Huang, Wei Tian, Zhong Gao, and Lu Xiong. Monorun: Monocular 3d object detection by reconstruction and uncertainty propagation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 10379–10388, 2021.

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, pp. 1597–1607. PMLR, 2020.

Xinlei Chen and Kaiming He. Exploring simple siamese representation learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 15750–15758, 2021.

Tim R. Davidson, Luca Falorsi, Nicola De Cao, Thomas Kipf, and Jakub M. Tomczak. Hyperspherical variational auto-encoders. *arXiv preprint arXiv:1804.00891*, 2018.

Ronald A. Fisher. Dispersion on a sphere. *Proceedings of the Royal Society of London. Series A. Mathematical and Physical Sciences*, 217(1130):295–305, 1953.

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

Yarin Gal et al. Uncertainty in deep learning. 2016.

Songwei Ge, Shlok Mishra, Simon Kornblith, Chun-Liang Li, and David Jacobs. Hyperbolic contrastive learning for visual representations beyond objects. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 6840–6849, 2023.

Hariprasath Govindarajan, Per Sidén, Jacob Roll, and Fredrik Lindsten. Dino as a von mises-fisher mixture model. In *The Eleventh International Conference on Learning Representations, ICLR 2023*, 2023.

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. *Advances in neural information processing systems*, 33:21271–21284, 2020.

Paul Hager, Martin J Menten, and Daniel Rueckert. Best of both worlds: Multimodal contrastive learning with tabular and imaging data. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 23924–23935, 2023.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 9729–9738, 2020.

Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 16000–16009, 2022.

Dan Hendrycks and Thomas Dietterich. Benchmarking neural network robustness to common corruptions and perturbations. In *International Conference on Learning Representations*, 2018.

Dan Hendrycks and Thomas Dietterich. Benchmarking neural network robustness to common corruptions and perturbations. *arXiv preprint arXiv:1903.12261*, 2019.

Gao Huang, Yixuan Li, Geoff Pleiss, Zhuang Liu, John E Hopcroft, and Kilian Q Weinberger. Snapshot ensembles: Train 1, get m for free. In *International Conference on Learning Representations*, 2016.

Kim-Celine Kahl, Carsten T Lüth, Maximilian Zenk, Klaus Maier-Hein, and Paul F Jaeger. Values: A framework for systematic validation of uncertainty estimation in semantic segmentation. *arXiv preprint arXiv:2401.08501*, 2024.

Berk Kaya, Suryansh Kumar, Carlos Oliveira, Vittorio Ferrari, and Luc Van Gool. Uncertainty-aware deep multi-view photometric stereo. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 12601–12611, 2022.

Alex Kendall and Yarin Gal. What uncertainties do we need in bayesian deep learning for computer vision? *Advances in neural information processing systems*, 30, 2017.

Been Kim and Finale Doshi-Velez. Machine learning techniques for accountability. *AI Magazine*, 42(1):47–52, 2021.

Durk P Kingma, Tim Salimans, and Max Welling. Variational dropout and the local reparameterization trick. *Advances in neural information processing systems*, 28, 2015.

Michael Kirchhof, Enkelejda Kasneci, and Seong Joon Oh. Probabilistic contrastive learning recovers the correct aleatoric uncertainty of ambiguous inputs. *arXiv preprint arXiv:2302.02865*, 2023.

Michael Kirchhof et al. A non-isotropic probabilistic take on proxy-based deep metric learning. In *European Conference on Computer Vision*, Cham, 2022. Springer Nature Switzerland.

Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009.

Johnson Kuan and Jonas Mueller. Back to the basics: Revisiting out-of-distribution detection baselines. *arXiv preprint arXiv:2207.03061*, 2022.

Yann LeCun. The mnist database of handwritten digits. *http://yann. lecun. com/exdb/mnist/*, 1998.

Weiyang Liu, Yandong Wen, Zhiding Yu, Ming Li, Bhiksha Raj, and Le Song. Sphereface: Deep hypersphere embedding for face recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 212–220, 2017.

Haodong Lu, Dong Gong, Shuo Wang, Jason Xue, Lina Yao, and Kristen Moore. Learning with mixture of prototypes for out-of-distribution detection. *arXiv preprint arXiv:2402.02653*, 2024.

Andrey Malinin and Mark Gales. Predictive uncertainty estimation via prior networks. *Advances in neural information processing systems*, 31, 2018.

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

Yifei Ming, Yiyou Sun, Ousmane Dia, and Yixuan Li. How to exploit hyperspherical embeddings for out-of-distribution detection? In *The Eleventh International Conference on Learning Representations*, 2022.

Miguel Monteiro, Loïc Le Folgoc, Daniel Coelho de Castro, Nick Pawlowski, Bernardo Marques, Konstantinos Kamnitsas, Mark van der Wilk, and Ben Glocker. Stochastic segmentation networks: Modelling spatially correlated aleatoric uncertainty. *Advances in neural information processing systems*, 33:12756–12767, 2020.

Vinod Nair and Geoffrey E. Hinton. Rectified linear units improve restricted boltzmann machines. In *Proceedings of the 27th International Conference on International Conference on Machine Learning*, pp. 807–814. Omnipress, 2010.

Maximilian Nickel and Douwe Kiela. Poincaré embeddings for learning hierarchical representations. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2017.

Seong Joon Oh, Kevin Murphy, Jiyan Pan, Joseph Roth, Florian Schroff, and Andrew Gallagher. Modeling uncertainty with hedged instance embedding. *arXiv preprint arXiv:1810.00319*, 2018.

Brett D Roads and Bradley C Love. 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.

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*, pp. 10612–10622, 2021.

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

Yonglong Tian, Dilip Krishnan, and Phillip Isola. Contrastive multiview coding. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XI 16*, pp. 776–794. Springer, 2020.

Uddeshya Upadhyay, Shyamgopal Karthik, Massimiliano Mancini, and Zeynep Akata. Probvlm: Probabilistic adapter for frozen vison-language models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 1899–1910, 2023.

Feng Wang and Huaping Liu. Understanding the behaviour of contrastive loss. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 2495–2504, 2021.

Feng Wang, Xiang Xiang, Jian Cheng, and Alan Loddon Yuille. Normface: L2 hypersphere embedding for face verification. In *Proceedings of the 25th ACM international conference on Multimedia*, pp. 1041–1049, 2017.

Tongzhou Wang and Phillip Isola. Understanding contrastive representation learning through alignment and uniformity on the hypersphere. In *International Conference on Machine Learning*, pp. 9929–9939. PMLR, 2020.

George Neville Watson. *A treatise on the theory of Bessel functions*, volume 2. The University Press, 1922.

Jiacheng Xu and Greg Durrett. Spherical latent spaces for stable variational autoencoders. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 2018.

Jure Zbontar, Li Jing, Ishan Misra, Yann LeCun, and Stéphane Deny. Barlow twins: Self-supervised learning via redundancy reduction. In *International Conference on Machine Learning*, pp. 12310–12320. PMLR, 2021.

Jingyang Zhang, Jingkang Yang, Pengyun Wang, Haoqi Wang, Yueqian Lin, Haoran Zhang, Yiyou Sun, Xuefeng Du, Kaiyang Zhou, Wayne Zhang, et al. Openood v1. 5: Enhanced benchmark for out-of-distribution detection. *arXiv preprint arXiv:2306.09301*, 2023.

APPENDIX

A PROOF OF PRESERVING SIMILARITY RANKING WITH UNNORMALIZED *vMF* DISTRIBUTION

Proposition 1. *For any two embeddings x_1 and x_2, if $p(x_1; \mu_1, \kappa_1) > p(x_2; \mu_2, \kappa_2)$, then:*

$$\exp(\kappa_1 \mu_1^\top x_1) > \exp(\kappa_2 \mu_2^\top x_2),$$

*thus preserving the ranking of similarities between embeddings, even in the unnormalized form of the von Mises-Fisher (*vMF*) distribution.*

Proof.

1. Recall that the probability density function of the normalized *vMF* distribution is given by:
$$p(x; \mu, \kappa) = C(\kappa) \exp(\kappa \mu^\top x),$$
where $C(\kappa)$ is the normalization constant, defined as:
$$C(\kappa) = \frac{\kappa^{d/2-1}}{(2\pi)^{d/2} I_{d/2-1}(\kappa)},$$
and $I_\nu(\kappa)$ is the modified Bessel function of the first kind of order ν.
2. Given the assumption $p(x_1; \mu_1, \kappa_1) > p(x_2; \mu_2, \kappa_2)$, we express this inequality as:
$$C(\kappa_1) \exp(\kappa_1 \mu_1^\top x_1) > C(\kappa_2) \exp(\kappa_2 \mu_2^\top x_2).$$
3. Taking the natural logarithm of both sides (which preserves the inequality) gives:
$$\ln(C(\kappa_1)) + \kappa_1 \mu_1^\top x_1 > \ln(C(\kappa_2)) + \kappa_2 \mu_2^\top x_2.$$
4. Rearranging this inequality, we obtain:
$$\kappa_1 \mu_1^\top x_1 > \kappa_2 \mu_2^\top x_2 + \ln(C(\kappa_2)) - \ln(C(\kappa_1)).$$
5. Exponentiating both sides (which again preserves the inequality) yields:
$$\exp(\kappa_1 \mu_1^\top x_1) > \exp(\kappa_2 \mu_2^\top x_2) \cdot \exp(\ln(C(\kappa_2)) - \ln(C(\kappa_1))).$$
6. Simplifying the right-hand side, we get:
$$\exp(\kappa_1 \mu_1^\top x_1) > \exp(\kappa_2 \mu_2^\top x_2) \cdot \frac{C(\kappa_2)}{C(\kappa_1)}.$$
7. Note that $C(\kappa) > 0$ for $\kappa > 0$. Define $\alpha = \frac{C(\kappa_2)}{C(\kappa_1)}$, where $\alpha > 0$. Therefore, we can rewrite the inequality as:
$$\exp(\kappa_1 \mu_1^\top x_1) > \alpha \cdot \exp(\kappa_2 \mu_2^\top x_2).$$
8. Since $\alpha > 0$, we conclude:
$$\exp(\kappa_1 \mu_1^\top x_1) > \exp(\kappa_2 \mu_2^\top x_2),$$
thereby proving that the unnormalized *vMF* distribution preserves the ranking of similarities between embeddings.

□

Corollary 1. *The unnormalized form of the* vMF *distribution retains the relative ordering of embedding similarities.*

Remark. This theoretical result provides a strong justification for employing the unnormalized *vMF* distribution in contrastive learning. In high-dimensional settings, computing the normalization constant $C(\kappa)$ becomes computationally expensive and prone to overflow due to the exponential growth of the Bessel function. By utilizing the unnormalized form, we avoid these computational burdens while preserving the essential ranking properties of embeddings, leading to more efficient and numerically stable optimization.

B DESCRIPTION OF CORRUPTION TYPES FROM CIFAR-10-C

Table 5: Types of image corruption and their descriptions from CIFAR-10-C (Hendrycks & Dietterich, 2019).

Type	Description
Gaussian Noise	Often occurs in conditions of poor lighting and adds random fluctuations to pixel values.
Shot Noise	Represents electronic noise emerging due to the inherent discreteness of light, leading to pixel-level variability.
Impulse Noise	Similar to the color version of salt-and-pepper noise, arises from bit errors and manifests as isolated pixel outliers.
Defocus Blur	Occurs when images are not in sharp focus, resulting in a slight blurriness.
Frosted Glass Blur	Resembles the effect seen through frosted glass surfaces, introducing a diffuse and obscured appearance.
Motion Blur	Created by rapid camera movements, causing objects to appear streaked or elongated.
Zoom Blur	Results from quickly moving the camera towards an object, causing a radial blurring effect.
Snow	An obstruction in visual perception, characterized by the presence of white or colored specks in the image.
Frost	Ice crystals on lenses or windows disrupt image clarity, leading to a frosted appearance.
Fog	Cloaks objects in images, simulated using the diamond-square algorithm, resulting in a hazy and obscured view.
Brightness	Affected by variations in daylight intensity, causing overall illumination changes.
Contrast	Depends on lighting conditions and the object's inherent color, leading to alterations in image contrast.
Elastic Transformations	Lead to stretching or contracting of small regions in an image, distorting local features.
Pixelation	A consequence of enlarging a low-resolution image, causing blocky artifacts due to limited pixel information.
JPEG Compression	A lossy method that reduces image size and can introduce artifacts such as blockiness and blurring.

C ANALYSIS OF THE GRADIENTS FROM $\mathcal{L}_{\text{ALIGN}}$

The gradient of $\mathcal{L}_{\text{align}}$ w.r.t. $\boldsymbol{\mu}_1$.

Given L_{a} in Eq. 9, the gradient of its log w.r.t. $\boldsymbol{\mu}_1$ can be obtained by differentiating the loss function w.r.t. $\boldsymbol{\mu}_1$. Its gradient can be expanded as follows:

$$\nabla_{\boldsymbol{\mu}_1} \log L_{\text{a}} = \quad \nabla_{\boldsymbol{\mu}_1} \left[\kappa_1 \cdot \cos(\theta) + \kappa_2 \cdot \cos(\theta)\right] \tag{12}$$

Now, $\cos(\theta) = \boldsymbol{\mu}_1^T \boldsymbol{\mu}_2$, and its gradient w.r.t. $\boldsymbol{\mu}_1$ is $\boldsymbol{\mu}_2$. Plug this into the gradient of $\mathcal{L}_{\text{align}}$, we get:

$$\nabla_{\boldsymbol{\mu}_1} \mathcal{L}_{\text{align}} = -\lambda_{\text{align}} \cdot (\kappa_1 \boldsymbol{\mu}_2 + \kappa_2 \boldsymbol{\mu}_2) \tag{13}$$

This gradient aligns $\boldsymbol{\mu}_1$ towards $\boldsymbol{\mu}_2$, similar to those with existing contrastive losses. More importantly, however, the *strength* of this alignment effect is *controlled* by the estimated concentration parameters κ_1 and κ_2 (*i.e.,* the estimated uncertainties) of both $\boldsymbol{\mu}_1$ and $\boldsymbol{\mu}_2$. Smaller κ's indicate more uncertainties and lead to looser alignment. Compared with conventional contrastive losses which naively align positive pairs regardless of the severity of corruptions in the input, our $\mathcal{L}_{\text{align}}$ yields a more flexible latent space that is aware of the severity of corruptions in the input.

The gradient of $\mathcal{L}_{\text{align}}$ w.r.t. κ_1. Similarly, we can compute the gradient of $\mathcal{L}_{\text{align}}$ w.r.t. κ_1 as follows:

$$\nabla_{\kappa_1} \mathcal{L}_{\text{align}} = -\lambda_{\text{align}} \cdot \boldsymbol{\mu}_1^T \boldsymbol{\mu}_2 \tag{14}$$

Eq. 14 implies that a closer cosine distance between $\boldsymbol{\mu}_1$ and $\boldsymbol{\mu}_2$ encourages a stronger increase in κ_1, indicating reduced uncertainty. The increasing effect on κ_1 weakens as the distance between $\boldsymbol{\mu}_1$

and $\boldsymbol{\mu}_2$ grows. Meanwhile, when the angle between $\boldsymbol{\mu}_1$ and $\boldsymbol{\mu}_2$ surpasses $\frac{\pi}{2}$, the gradient encourages a reduction in κ_1 instead, hence an increase in predicted uncertainty. Of note, κ's would not grow uninformatively large as they are bounded by the ℓ_2 regularization (Eq. **??**) at the same time.

D Discussion on Network Complexity, Embedding Dimensions, and Learning Frameworks

Table 6 further demonstrates the versatility of our approach across different network architectures, including *ResNet18*, *ResNet34*, and *ResNet50*. Our method consistently achieves strong correlation coefficients, illustrating that the introduction of κ does not compromise the discriminative nature of the embeddings. Instead, it enriches the model's representation by providing a probabilistic dimension that captures uncertainty directly related to the data's intrinsic characteristics.

The compatibility of our framework with established contrastive learning methods, such as *SimSiam* (Chen et al., 2020), *BYOL* (Grill et al., 2020), and *SwaV* (Caron et al., 2020), further attests to its adaptability. Table 3 demonstrates our framework's versatility, particularly with *SimSiam* and *BYOL*, which train using only positive pairs. Across these methods, our approach consistently achieves strong correlation coefficients, underscoring the substantial promise of our design. This extension is not merely a testament to the flexibility of our approach but also promises to broaden the applicability of contrastive learning models in handling diverse applications.

In Table 4, we investigate the effect of embedding dimensions on κ's capability to quantify uncertainty. With embedding dimensions set at 64, 128, 256, and 384, our framework demonstrates a nuanced performance variation, indicated by the correlation coefficients -0.768, -0.883, -0.844, and -0.901, respectively. The optimal performance at 128 and 384 dimensions suggests a critical balance between dimensionality and the model's ability to effectively capture uncertainty.

Table 6: **The effect of network complexity** with fixed λ_{align}, λ_κ, and number of embedding dimension (dim. = 128). 'Correction' refers to the average of 18 Spearman correlations from the types of corruption listed in Table 1.

Architecture	ResNet18	ResNet34	ResNet50
Correlation	-0.908	-0.876	-0.883

Table 7: Ablation study on the effect of varying λ_{align} on Spearman correlation and top-1 accuracy.

λ_{align}	Spearman Correlation	Top-1 Accuracy
0.001	-0.844	0.860
0.005	-0.857	0.862
0.01	-0.884	0.854
0.02	-0.869	0.849
0.04	-0.884	0.845
0.1	-0.870	0.831

Table 8: Ablation study on the effect of varying λ_{reg} while fixing $\lambda_{\text{align}} = 0.05$. 'NaN' indicates that the κ is constant for all samples when κ is not well regularized (i.e., small λ_{reg}).

λ_{reg}	Spearman Correlation	Top-1 Accuracy
0.0005	NaN	0.10
0.001	NaN	0.10
0.002	-0.831	0.868
0.004	-0.884	0.865
0.01	-0.862	0.854
0.02	-0.862	0.858
0.04	-0.853	0.864

E Ablation Study on Hyper-parameters

We conducted ablation experiments to investigate the impact of the regularization parameters λ_{align} and λ_{reg} on training stability and performance. The Spearman correlation and top-1 accuracy on the test set of CIFAR-10 are reported in Tables 7 and 8, which demonstrate the effect of varying λ_{align} and λ_{reg}, respectively.

Effect of λ_{align}. In Table 7, we observe that increasing λ_{align} leads to a slight deterioration in embedding quality, as indicated by the drop in top-1 accuracy. However, the Spearman correlation remains relatively stable across different values of λ_{align}, suggesting that this regularization term stabilizes training without significantly affecting the relative ranking of embeddings in terms of similarity.

Effect of λ_{reg}. Table 8 illustrates the impact of varying λ_{reg} while keeping λ_{align} fixed at 0.05. Weak regularization (e.g., $\lambda_{\text{reg}} < 0.001$) leads to instability during training, reflected in the significantly lower top-1 accuracy. On the other hand, stronger regularization results in only a slight decrease in the correlation coefficient, while still maintaining competitive performance.

F MC-InfoNCE with the *SimCLR* contrastive loss

```
from torch import nn
import torch
from vmf_sampler import VonMisesFisher
from utils_mc import pairwise_cos_sims, pairwise_l2_dists, log_vmf_norm_const
import torch
import torch.nn as nn

class MCSimCLR(nn.Module):
    def __init__(self, kappa_init=16, n_samples=64, temperature=0.5, device=torch.device('cuda:0')):
        super().__init__()
        self.n_samples = n_samples
        self.kappa = torch.nn.Parameter(torch.ones(1, device=device) * kappa_init, requires_grad=True)
        self.temperature = temperature

    def forward(self, mu1, kappa1, mu2, kappa2):
        # Draw samples from the von Mises-Fisher distribution
        samples1 = VonMisesFisher(mu1, kappa1).rsample(torch.Size([self.n_samples]))
        samples2 = VonMisesFisher(mu2, kappa2).rsample(torch.Size([self.n_samples]))
        # Concatenate positive samples for contrastive loss calculation
        samples = torch.cat([samples1, samples2], dim=1)  # [n_MC, 2 * batch, dim]
        # Compute similarity matrix
        sim_matrix = torch.exp(torch.matmul(samples, samples.transpose(2, 1)) / self.temperature)
        # Create mask to zero-out self-similarities (diagonal elements)
        batch_size = mu1.size(0)
        mask = ~torch.eye(2 * batch_size, device=sim_matrix.device, dtype=bool).repeat(self.n_samples, 1, 1)
        sim_matrix = sim_matrix.masked_select(mask).view(self.n_samples, 2 * batch_size, -1)
        # Similarities for the positive pairs)
        pos_sim = torch.exp(torch.sum(samples1 * samples2, dim=2) / self.temperature)
        pos_sim = torch.cat([pos_sim, pos_sim], dim=1) #Duplicate pos_sim
        loss = -torch.log(pos_sim / sim_matrix.sum(dim=2))
        loss = loss.mean()
        return loss
```

G ARCHITECTURE

```python
import torch
import torch.nn as nn
import torch.nn.functional as F
from torchvision.models import resnet50, resnet18, resnet34
class ProbabilisticModel(nn.Module):
    def __init__(self, feature_dim=128):
        super(ProbabilisticModel, self).__init__()

        # Define the layers of the ResNet model
        self.f = []
        for name, module in resnet50().named_children():
            if name == 'conv1':
                module = nn.Conv2d(3, 64, kernel_size=3, stride=1, padding=1, bias=False)
            if not isinstance(module, nn.Linear) and not isinstance(module, nn.MaxPool2d):
                self.f.append(module)
        self.f = nn.Sequential(*self.f)

        # Projection head for feature
        self.g = nn.Sequential(
            nn.Linear(2048, 512, bias=False),
            nn.BatchNorm1d(512),
            nn.ReLU(inplace=True),
            nn.Linear(512, feature_dim, bias=True)
        )
        # Additional layer for kappa (concentration parameter)
        self.kappa_head = nn.Sequential(
            nn.Linear(2048, 512, bias=False),
            nn.BatchNorm1d(512),
            nn.ReLU(inplace=True),
            nn.Linear(512, 1, bias=True)  # Outputs kappa for each sample
        )

    def forward(self, x):
        x = self.f(x)
        feature = torch.flatten(x, start_dim=1)
        out = self.g(feature)
        kappa = self.kappa_head(feature)  # Compute kappa for each sample
        # Normalize the feature vector and return it with variance and kappa
        return F.normalize(out, dim=-1), F.softplus(kappa.squeeze(-1))

```

H TRAINING

```python

# train for one epoch to learn the mean vector mu and kappa
def train(net, data_loader, train_optimizer):
    net.train()
    total_loss, total_num, train_bar = 0.0, 0, tqdm(data_loader)
    epsilon = 1e-6  # Small constant for numerical stability
    align_strength = 0.05  # Hyperparameter to regularize the embedding alignment loss
    kappa_reg_strength = 0.005  # Hyperparameter for the regularization strength
    simclr_strength = 1  # Hyperparameter for the strength of SimCLR loss

    for pos_1, pos_2, target in train_bar:
        pos_1, pos_2 = pos_1.to(device), pos_2.to(device)

        mean_1, kappa_1 = net(pos_1)
        mean_2, kappa_2 = net(pos_2)

        # Compute the embedding alignment loss component
        alignment = torch.exp(kappa_1 * F.cosine_similarity(mean_1, mean_2, dim=1)+ \
        kappa_2 * F.cosine_similarity(mean_1, mean_2, dim=1))
        align_loss = align_strength * (-torch.log(alignment + epsilon).mean())

        # Compute the regularization loss for kappa (L2 norm)
        kappa_reg_loss = kappa_reg_strength * (torch.mean(kappa_1 ** 2) + \
        torch.mean(kappa_2 ** 2))


        # Compute SimCLR contrastive loss
        out = torch.cat([mean_1, mean_2], dim=0)
        sim_matrix = torch.exp(torch.mm(out, out.t().contiguous()) / temperature)
        mask = (torch.ones_like(sim_matrix) - torch.eye(2 * batch_size, \
        device=sim_matrix.device)).bool()
        sim_matrix = sim_matrix.masked_select(mask).view(2 * batch_size, -1)
        pos_sim = torch.exp(torch.sum(mean_1 * mean_2, dim=-1) / temperature)
        pos_sim = torch.cat([pos_sim, pos_sim], dim=0)
        contrastive_loss = simclr_strength * \
        (-torch.log(pos_sim / sim_matrix.sum(dim=-1))).mean()

        # Compute the final loss
        loss = align_loss + contrastive_loss + kappa_reg_loss

        # Backward and optimize
        train_optimizer.zero_grad()
        loss.backward()
        train_optimizer.step()

```