Representation Uncertainty in Self-Supervised Learning as Variational Inference

Hiroki Nakamura*
Panasonic Holdings Corp.
nakamura.hiroki003@jp.panasonic.com

Masashi Okada*
Panasonic Holdings Corp.
okada.masashi001@jp.panasonic.com

Tadahiro Taniguchi
Ritsumeikan University
taniguchi@em.ci.ritsumei.ac.jp

Abstract

In this study, a novel self-supervised learning (SSL) method is proposed, which considers SSL in terms of variational inference to learn not only representation but also representation uncertainties. SSL is a method of learning representations without labels by maximizing the similarity between image representations of different augmented views of an image. Meanwhile, variational autoencoder (VAE) is an unsupervised representation learning method that trains a probabilistic generative model with variational inference. Both VAE and SSL can learn representations without labels, but their relationship has not been investigated in the past. Herein, the theoretical relationship between SSL and variational inference has been clarified. Furthermore, a novel method, namely variational inference SimSiam (VI-SimSiam), has been proposed. VI-SimSiam can predict the representation uncertainty by interpreting SimSiam with variational inference and defining the latent space distribution. The present experiments qualitatively show that VI-SimSiam could learn uncertainty by comparing input images and predicted uncertainties. Additionally, we described a relationship between estimated uncertainty and classification accuracy.

1. Introduction

Self-supervised learning (SSL) is a framework for learning representations of data [6, 14, 20, 7, 5, 61, 50, 37, 28, 39]. This method enables training of high-performance models in downstream tasks (e.g., image classification and object detection) without substantial manually labeled data through pre-training the network to generate features. It can mitigate the annotation bottleneck, one of the crucial barriers to the practical application of deep learning. Some state-of-the-art SSL methods, such as SimSiam [6], SimCLR [5], and DINO [4], train image encoders by maximizing the similarity between representations of different augmented views of an image.

The probabilistic generative models with variational inference provide another approach for representation learning [27]. This approach learns latent representations in an unsupervised fashion by training inference and generative models (i.e., autoencoders) together. It can naturally incorporate representation uncertainty by formulating them as probabilistic distribution models (e.g., Gaussian). However, the pixel-wise objective for reconstruction is sensitive to rare samples [32] in such methods. Furthermore, this representation learning is recently found to be less competitive than the SSL methods on the benchmarking classification tasks [32, 37]. Although SSL and variational inference seem...
highly related learning representations without supervision, their theoretical connection has not been fully explored.

In this study, we incorporate the variational inference concept to make the SSL uncertainty-aware and conduct a detailed representation uncertainty analysis. The contributions of this study are summarized as follows.

• We clarify the relationship between SSL (i.e., SimSiam, SimCLR, and DINO) and variational inference, generalizing the SSL methods as the variational inference of spherical or categorical latent variables (§4).

• We derive a novel SSL method called variational inference SimSiam (VI-SimSiam) by incorporating the above relationship. It learns to predict not only representations but also their uncertainty (§5).

• We demonstrate that VI-SimSiam successfully estimates uncertainty without labels while achieving competitive classification performance with SimSiam. We qualitatively evaluate the uncertainty estimation capability by comparing input images and the estimated uncertainty parameter $\kappa$, as shown in Fig. 1. In addition, we also describe that the predicted representation uncertainty $\kappa$ is related to the accuracy of the classification task (§6).

A comparison of SimSiam and VI-SimSiam is illustrated in Fig. 2, where VI-SimSiam estimates the uncertainty by predicting latent distributions.

2. Related work

2.1. Self-supervised learning

SSL [5, 20, 6, 14, 4] has been demonstrated to have notable performance in many downstream tasks, such as classification and object detection. Contrastive SSL methods, including SimCLR[5] and MoCo [20], learn to increase the similarity of representation pairs augmented from an image (positive pairs) and to decrease the similarity of representation pairs augmented from different images (negative pairs). Conversely, non-contrastive SSL, including SimSiam [6], BYOL [14], and DINO [4], learn a model using only the positive pairs. Zbontar et al. [61] proposed another non-contrastive method using the redundancy-reduction principle of neuroscience. Furthermore, several studies have theoretically analyzed the SSL methods, such as Tian et al. [50] investigated the reasons behind the superior performance of the non-contrastive methods. Tao et al. [49] claimed that the (non-)contrastive SSL methods can be unified into one form using gradient analysis. Zhang [62] demonstrated a theoretical connection between masked autoencoder [19] and contrastive learning. However, these studies assumed a deterministic formulation without considering uncertainty in the representations.

2.2. Variational inference

In deep learning, variational inference is generally formulated using auto-encoding variational Bayes [27]. The variational inference estimates latent distributions, such as Gaussian [27], Gaussian mixture [51], and von Mises-Fischer (vMF) distribution for hyperspherical latent space [9]. Although Wang et al. [53] pointed out that SSL methods learn representations on the hypersphere, their relevance to the spherical variational inference [9] has yet to be investigated extensively.

A multimodal variational autoencoder [31, 55, 47, 48] is trained to infer latent variables from multiple observations from different modalities. The latent variable distribution of multimodal variational inference is often assumed to be the product of experts or the mixture of experts of unimodal distributions [31, 55, 47]. Sutter et al. [48] also clarified the connection between them and generalized them as mixture-of-products-of-experts (MoPoE) VAE. Multimodal variational inference seems highly related to the SSL utiliz-
ing the multiview inputs. However, their relationship has been unclear.

2.3. Uncertainty-aware methods

Uncertainty-aware methods [12, 25, 26, 36, 54] have been proposed to solve the problem of learning hindered by data with high uncertainty. Kendall and Gal [25] proposed a method that estimated data uncertainty in regression and classification tasks by assuming that outputs follow a normal distribution. Scott et al. [46] proposed a stochastic spherical loss for classification tasks based on the von Mises–Fisher distribution. Additionally, Mohseni et al. [36] and Winkens et al. [54] presented methods for estimating uncertainty distributions. Additionally, Mohseni et al. [36] and Winkens et al. [54] presented methods for estimating uncertainty by combining SSL with a supervised classification task. Uncertainty-aware methods have been proposed for other tasks as well, such as human pose estimation [44, 41, 15], optical flow estimation [23], object detection [22, 17], and reinforcement learning [33, 40, 16].

Several studies have suggested methods to incorporate uncertainty in self-supervised learning of specific tasks [45, 42, 59, 52]. Poggi et al. [45] proposed an uncertainty-aware and self-supervised depth estimation. They considered the variance in depth estimated from multiple models as uncertainty. Wang et al. [52] demonstrated an uncertainty-aware SSL for three-dimensional object tracking, wherein the ratio of the distances between positive and negative pairs was treated as uncertainty. However, these methods hardly discussed representation uncertainty.

3. Preliminary

The formulations of the SSL methods and variational inference are briefly reviewed in this section.

3.1. Self-supervised learning methods

**SimSiam** SimSiam is a non-contrastive SSL method with an objective function that is defined as follows;

\[
\mathcal{J}_{\text{SimSiam}} := g_\theta(x_1)^T f_\phi(x_2) + g_\theta(x_2)^T f_\phi(x_1),
\]

where \(x_1\) and \(x_2\) are two augmented views of a single image. The term \(g_\theta\) and \(f_\phi\) are encoders parameterized with \(\theta\) and \(\phi\), respectively, and they map the image \(x\) to a spherical latent \(z \in \mathbb{S}^{d-1}\). In the literature on non-contrastive SSL, the two encoders \(g_\theta\) and \(f_\phi\) are referred to as online and target networks, respectively. SimSiam defines the online network \(g_\theta\) as \(g_\theta = h_\theta \circ f_\theta\), where \(h_\theta\) and \(f_\theta\) are referred to as a predictor network and projector network [14, 6], respectively.

**SimCLR** The objective function of a contrastive SSL, such as SimCLR is generally described as follows;

\[
\mathcal{J}_{\text{SimCLR}} := \mathcal{J}_{\text{SimCLR}}^{(1,2)} + \mathcal{J}_{\text{SimCLR}}^{(2,1)},
\]

\[
\mathcal{J}_{\text{SimCLR}}^{(i,j)} := \log \frac{\exp(p_\phi(x_i)^T f_\phi(x_j))}{\sum_{x \in B} \exp(g_\theta(x_i)^T f_\phi(x_j))},
\]

where \(g_\theta(x_i), f_\phi(x_j) \in \mathbb{S}^{d-1}\), and \(B\) denotes a minibatch.

**DINO** DINO is another non-contrastive SSL with an objective and a latent space (i.e., categorical latent) different from those of SimSiam. It is described as;

\[
\mathcal{J}_{\text{DINO}} := -\mathcal{H}(P_1, \phi, P_2, \theta) - \mathcal{H}(P_2, \phi, P_1, \theta),
\]

\[
P_i, \phi := \text{softmax} \left( \frac{(f_\phi(x_i) - c)/\tau}{\tau_\phi} \right),
\]

\[
P_j, \theta := \text{softmax} \left( \frac{(g_\theta(x_j) - c)/\tau_\phi}{\tau_\theta} \right),
\]

where \(\mathcal{H}(P, Q) := \sum P \log Q\) denotes cross entropy between two probabilities, \(g_\theta(x), f_\phi(x) \in \mathbb{R}^d\), and \((\tau, c)\) are the parameters for sharpening and centering operations discussed later in §4.3.

3.2. Multimodal generative model and inference

Fig. 3 shows a graphical model for multimodal generative models, where \(D\) indicates a dataset, \(X = \{x_i\}_{i=1}^M\) is a set of multimodal observations \(x_i\), \(z\) is a latent variable of the observations, \(\theta\) is a deterministic parameter of the generative model \(p(X|z, \theta)\), and \(M\) is the number of modalities corresponding to the data augmentation types in this paper. In the SSL context, \(X\) can be regarded as augmented images from stochastic generative processes. The objective is to find a parameter \(\theta^*\) that maximizes marginal observation likelihood;

\[
\theta^* = \arg\max_{\theta} \mathcal{J} = \arg\max_{\theta} \mathbb{E}_{p(z)} [\log p(X|z, \theta)].
\]

Since the marginalization \(\mathbb{E}_{p(z)} [\cdot]\) is intractable, we can instead maximize the evidence lower bound (ELBO);

\[
\mathbb{E}_{p(z)} [\log p(X|z, \theta)] \geq \mathcal{J}_{\text{ELBO}} :=
\]

\[
= \mathbb{E}_{q(z|X, \phi)} [\log p(X|z, \theta)] - D_{\text{KL}}[q(z|X, \phi)p(z)],
\]

where \(q(z|X, \phi)\) is a variational inference model parameterized with \(\phi\). By optimizing \(\mathcal{J}_{\text{ELBO}}\) with respect to both \(\theta\) and \(\phi\), \(q(z|X, \phi)\) approaches the posterior \(p(z|X, \theta)\) as the following relation holds;

\[
\mathcal{J} - \mathcal{J}_{\text{ELBO}} = D_{\text{KL}}[q(z|X, \phi)p(z|X, \theta)] \geq 0,
\]
where \( D_{\text{KL}}[\cdot] \) is the Kullback-Leibler divergence. Notably, the posterior varies during the optimization process since it depends on the parameterized generative model; i.e.,
\[
p(z|x, \theta) \propto p(x|z, \theta)p(z).
\]

### 4. Self-supervised learning as inference

This section shows a connection between SSL and multimodal variational inference. Usually, a generative model \( p(x|z, \theta) \) and variational inference model \( q(z|x, \phi) \) are realized with deep neural networks to solve the problem of Eq. (6), and they are trained via ELBO optimization. Instead, let us consider directly realizing the posterior \( p(z|x, \theta) \) as deep neural networks. For this purpose, we remove the generative model term in Eq. (7) by applying Bayes’ theorem;
\[
p(x|z, \theta) = p(z|x, \theta)p(x|\theta)p(z|x, \theta)] = p(z|x, \theta)p(x|\theta)p(z). \tag{9}
\]

Since \( p(x|\theta) \) is intractable, we approximate it with the empirical data distribution \( p_D(x) \). Substituting Eq. (9) into Eq. (7) and applying the approximation yields a new objective;
\[
J_{\text{SSL}} := J_{\text{align}} + J_{\text{uniform}} + J_{\text{KL}} + p_D(x),
\]
\[
\pm J_{\text{align}} + J_{\text{uniform}} + J_{\text{KL}}, \tag{10}
\]

where,
\[
J_{\text{align}} := \mathbb{E}_{q(z|x, \phi)} [\log p(z|x, \theta)], \tag{11}
\]
\[
J_{\text{uniform}} := \mathbb{E}_{q(z|x, \phi)} [-\log p_D(z|\theta)], \tag{12}
\]
\[
J_{\text{KL}} := D_{\text{KL}}(q(z|x, \phi)||p(z)), \tag{13}
\]
\[
p_D(z|\theta) := \mathbb{E}_{p_D(x)}[p(z|x, \theta)]. \tag{14}
\]

Furthermore, let \( p(z|x, \theta) \) and \( q(z|x, \phi) \) respectively be Product-of-Experts (PoE) and Mixture-of-Experts (MoE) of the single-modal inference models;
\[
p(z|x, \theta) = \eta_\theta \prod_{j=1}^M p(x_j | z_j, \theta), \tag{15}
\]
\[
q(z|x, \phi) := \frac{1}{M} \sum_{i=1}^M q(z|x_i, \phi), \tag{16}
\]

where \( \eta_\theta^{-1} := \int \prod p(x_j | z_j, \theta) dz \) is the renormalization term.

Then, we can rewrite \( J_{\text{align}} \) as a form that encourages aligning latent variables from different models;
\[
J_{\text{align}} = \sum_{i,j} \mathbb{E}_{q(z|x_i, \phi)} [\log p(z|x_j, \theta)] + M \log \eta_\theta, \tag{17}
\]

Eq. (15) is from Prop. 4.1 described below. Eq. (16) is the definition theoretically validated in [47, 48]. Practically, we can ignore unimodal comparisons (i.e., \( i = j \)) since they provide less effective information.

**Proposition 4.1.** Let \( p(z) \) be a non-informative prior. The multi-modal posterior \( p(z|x, \theta) \) takes the form of PoE of the single-modal posteriors \( p(z|x_j, \theta) \).

**Proof.** See Appx. A.1.

We claim that Eq. (12) generalizes the objectives in Eqs. (1), (2) and (3) as summarized in Table 1. In the rest of this section, we describe how to recover the objectives in the table from Eq. (10). In the derivations, the term \( M \log \eta_\theta \) is ignored by approximating it as a constant (denoted as \( \tilde{\eta} \)).

#### 4.1. SimSiam as inference

First, we discuss the relationship between SimSiam and the following definition involving the hyperspherical space \( S^{d-1} \).

**Definition 4.2.**
\[
p(z) := U(S^{d-1}), \tag{18}
\]
\[
q(z|x, \phi) := \frac{1}{M} \sum_{i=1}^M \delta(z - f_\phi(x_i)), \tag{19}
\]
\[
p(z|x, \theta) := \eta_\theta \prod_{j=1}^M \text{vMF}(z; \mu = g_\theta(x_j), \kappa), \tag{20}
\]
\[
\text{vMF}(z; \mu, \kappa) := C_{\text{vMF}}(\kappa) \exp(\kappa\mu^T z), \tag{21}
\]

where \( g_\theta(x), f_\phi(x) \in S^{d-1} \), and \( U(S^{d-1}) \) is the uniform distribution on the hypersphere. The term \( \text{vMF}(z; \mu, \kappa) \) is the von-Mises-Fisher distribution [35] parameterized with the mean direction \( \mu \in S^{d-1} \) and concentration (inversed variance) \( \kappa \in \mathbb{R}^+ \). The term \( C_{\text{vMF}}(\kappa) \) is a normalization constant defined with the modified Bessel function. In §4, \( \kappa \) and \( C_{\text{vMF}}(\kappa) \) are defined to be constants.

**Relation to \( J_{\text{align}} \)**

**Claim 4.3.** Under Def. 4.2, we can recover \( J_{\text{SimSiam}} \) in Eq. (1) from \( J_{\text{align}} \);
\[
J_{\text{align}} \overset{\Delta}{=} \sum_{i,j} q_\theta^T(x_j) f_\phi(x_i), \tag{22}
\]

**Proof.** See Appx. A.2.

**Relation to \( J_{\text{uniform}} \)**

**Claim 4.4.** The presence of the predictor \( h_\theta \) implicitly maximizes \( J_{\text{uniform}} \).

**Proof.** Here, we borrow theoretical findings from the DirectPred literature [50]. In the literature, considering that \( g_\theta = h_\theta \circ f_\phi \), the predictor \( h_\theta \) defined as the following linear model, can lead to successful convergence;
\[
h_\theta(z) := (U \Lambda \frac{1}{2} U^T) z, \tag{23}
\]
Table 1: The summary of SSL methods interpreted as variational inference.

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>$p(z)$</td>
<td>$\mathbb{U}^{(d-1)}$</td>
<td>Deterministic</td>
<td>$\mathbb{U}(\Delta^{d-1})$</td>
<td>Deterministic</td>
</tr>
<tr>
<td>$q(z</td>
<td>x, \phi)$</td>
<td>$\mathbb{U}^{(d-1)}$</td>
<td>Von-Mises-Fisher</td>
<td>Categorical</td>
</tr>
<tr>
<td>$p(z</td>
<td>x, \theta)$</td>
<td>$\mathbb{U}^{(d-1)}$</td>
<td>Categorical</td>
<td>Categorical</td>
</tr>
</tbody>
</table>

| $\mathcal{J}_{\text{align}}$ | $z^T_iz_i$ | DirectPred | Log, $\sum_i \exp(z_i^Tz_i)$ | Centering | Sharpening |
| $\mathcal{J}_{\text{uniform}}$ | $\mathcal{J}_{\text{KL}}$ | Const. | $\sum_i P_i \cdot \log P_i$ | Const. |            |

| Uncertainty aware | (uncertainty parameter $\kappa$ is fixed) | Yes | (but has not been discussed) | Yes | (uncertainty parameter $\kappa$ is estimated) |

where $U$ and $\Lambda$ are the eigenvectors and eigenvalues of the covariance matrix of random variables $z$ from $f_\theta(x)$; i.e.,

$$\mathbb{E}_{P_D}(z|f_\theta(x)f_\theta^T(x)) = U\Lambda U^T.$$  \hspace{1cm} (24)

With Eq. (23), the cosine similarity can be rewritten as\(^2\):

$$g^T_j(x_j)f_\phi(x_i) = (U^Tz_j)^T\Lambda^{\frac{1}{2}}(U^Tz_i),$$  \hspace{1cm} (25)

where $z_j = f_\phi(x_j)$ and $z_i = f_\phi(x_i)$ are projected by $U^T$ (i.e., projection matrix of PCA) so that each dimension is as independent as possible. In addition, the eigenvalues $\Lambda$ converge to the identity matrix, i.e., the variances of $z$ are constant. These behaviors encourage the marginalized $z$-distribution to be isotropic, i.e., $\mathbb{U}(\Delta^{d-1})$. Consequently, $\mathcal{J}_{\text{uniform}}$ is maximized since it takes the maximum iff $p_D(z|\theta) = \mathbb{U}(\Delta^{d-1})$.

Relation to $\mathcal{J}_{\text{KL}}$ Since $q(z|X, \phi)$ is deterministic, $\mathcal{J}_{\text{KL}}$ is constant.

4.2. SimCLR as inference

Further, we derive the contrastive learning objective with Def. 4.2. $\mathcal{J}_{\text{KL}}$ is constant, as in the previous section, and hence, is disregarded in the current section.

Relation to $\mathcal{J}_{\text{align}}$ and $\mathcal{J}_{\text{uniform}}$ Converse to the previous non-contrastive case, we consider explicitly optimizing $\mathcal{J}_{\text{uniform}}$, leading to the following claim:

Claim 4.5. Under Def. 4.2, we can recover $\mathcal{J}_{\text{SimCLR}}$ in Eq. (2) from $\mathcal{J}_{\text{align}}$ and $\mathcal{J}_{\text{uniform}}$:

$$\mathcal{J}_{\text{align}} + \mathcal{J}_{\text{uniform}} \simeq \sum_{i,j} \mathbb{E}_{q(z|x, \phi)} \left[ \log \frac{\exp(g\phi(x_i)^Tf\phi(x_j))}{\sum_{x_j \sim B} \exp(g\phi(x_j)^Tf\phi(x_j))} \right].$$  \hspace{1cm} (26)

Proof. See Appx. A.3.

Note that as per Ref. [53], Eq. (2) can be decomposed into two objectives similar to $\mathcal{J}_{\text{align}}$ and $\mathcal{J}_{\text{uniform}}$. In contrast, our derivation here aimed to derive Eq. (2) in a general form from variational inference.

4.3. DINO as inference

Finally, we derive the objective of DINO with the following definitions, where $z$ is defined in the simplex $\Delta^{d-1}$.

Definition 4.6.

$$p(z) := \mathbb{U}(\Delta^{d-1}),$$  \hspace{1cm} (27)

$$q(z|X, \phi) := \frac{1}{M} \sum_{i=1}^M \text{Cat} \left( z; P_i, \phi \right),$$  \hspace{1cm} (28)

$$p(z|x, \theta) := \eta_0 \prod_{j=1}^M \text{Cat} \left( z; P_j, \theta \right),$$  \hspace{1cm} (29)

where the definitions of $P_i, \phi$ and $P_j, \theta$ follow Eqs. (4) and (5), and Cat means categorical distribution.

Relation to $\mathcal{J}_{\text{align}}$

Claim 4.7. Under Def. 4.6, we can recover $\mathcal{J}_{\text{DINO}}$ in Eq. (3) from $\mathcal{J}_{\text{align}}$:

$$\mathcal{J}_{\text{align}} \simeq -\sum_{i,j} \mathcal{H}(P_i, \phi, P_j, \theta),$$  \hspace{1cm} (30)

Proof. Substituting Eqs. (28) and (29) into Eq. (11) derives Eq. (30).

Relation to $\mathcal{J}_{\text{uniform}}$

Claim 4.8. Centering in Eq. (4) optimizes $\mathcal{J}_{\text{uniform}}$.

Proof. The centering parameter $c$ in Eq. (4) is determined and updated with an exponential moving average of batch mean of $f_\phi; c \leftarrow mc + (1-m)\mathbb{E}_{B}[f_\phi(x)]$, where $m$ is a rate parameter. This batch-norm-like centering encourages the marginalized distribution $q(z|\phi) = \mathbb{E}_{P_D}(z)$. Therefore, the centering parameter $c$ is fixed at $0$.
to be $\mathcal{U}(\Delta^{d-1})^3$ As a result, $p_D(z|\theta)$ will also be uniform while optimizing $\mathcal{J}_{\text{align}}$, thus maximizing $\mathcal{J}_{\text{uniform}}$.

**Relation to $\mathcal{J}_{\text{KL}}$**

**Claim 4.9. Sharpening in Eq. (4) regularizes $\mathcal{J}_{\text{KL}}$.**

**Proof.** Since $p(z)$ is defined to be uniform, $\mathcal{J}_{\text{KL}}$ acts as an entropy regularizer to the variational distribution $q(z|X, \phi)$. With a carefully designed temperature $\tau_\phi$, sharpening directly regularizes the entropy for learning success; e.g., $\tau_\phi$ must be within 0.02 and 0.06 [4].

### 4.4. Discussion

**Relation to another SSL method** Moreover, we can integrate Barlow Twins [61] into this SSL as inference framework. Barlow Twins considers the cross-correlation matrix $E_{[x]}[g_\theta(x_i)f_\theta(x_j)^T]$ and optimizes its diagonal and non-diagonal elements to be ones and zeros, corresponding to the optimization of $\mathcal{J}_{\text{align}}$ and $\mathcal{J}_{\text{uniform}}$, respectively.

**Limitations** In the derivation, we made some assumptions: $p(X|\theta) \simeq p_D(X)$ in Eq. (9), $q(z|X, \phi)$ is MoE (Eq. (16)), and the renormalization term $\eta_\phi$ can be ignored in Eq. (17). The $\mathcal{J}_{\text{SSL}}$ derived under these assumptions is not guaranteed to be the lower bound of $\mathcal{J}$ in Eq. (6). However, the previous success of the SSL method empirically indicates that the assumptions are valid and the optimization progressed while keeping $\mathcal{J} \geq \mathcal{J}_{\text{SSL}}$. We hypothesize that to update the target network $f_\theta$ such as stop gradient and exponential moving average (EMA) [14, 6, 4, 18], the heuristics may contribute to the above optimization behavior. However, further theoretical analysis is necessary for future work. Notably, the stop gradient and EMA can be interpreted as the EM algorithm in variational inference [1]. It is also an attractive research direction to extend SSL methods by relaxing the assumption of $q(z|X, \phi)$; e.g., changing its form from MoE to MoPoE [48], and replacing the deterministic $\delta$ in Eq. (19) with probability distributions.

**Representation Uncertainty** As shown in Table 1. DINO inherently can estimate uncertainty because it fully estimates categorical parameters. However, its ability has hardly been discussed. Conversely, although SimSiam and SimCLR assumed hyperspherical distributions, the uncertainty parameter $\kappa$ is fixed, thus missing the ability for uncertainty estimation. To bridge the gap of uncertainty awareness, we propose a new uncertainty-aware method VI-SimSiam by extending SimSiam in §5. The capability of uncertainty estimation of VI-SimSiam and DINO is subsequently evaluated in §6.

---

5. Variational inference SimSiam

This section introduces a novel uncertainty-aware SSL method by extending SimSiam based on the SSL as inference principle. The previous derivation assumes that $\kappa$ is constant. We relax this assumption and allowed $\kappa$ to be estimated by a newly introduced encoder $u_\theta : x \rightarrow \kappa$. Additionally, we replace the vMF distribution with another spherical distribution, i.e., Power Spherical (PS) distribution [10]. This is because estimating gradients through the modified Bessel function in $C_{vMF}(\kappa)$ is computationally expensive and unstable [10]. The modified posterior is defined as:

$$p(z|X, \theta) := \eta_\theta \prod_{j=1}^M PS(z; \mu = g_\theta(x_j), \kappa = u_\theta(x_j)), \quad \text{(31)}$$

$$PS(z; \mu, \kappa) := C_{PS}(\kappa)(1 + \mu^T z)^\kappa. \quad \text{(32)}$$

The normalization constant of the PS distribution $C_{PS}(\kappa)$ is defined with the beta distribution. It can be efficiently computed using general deep-learning frameworks. Substituting Eqs. (19) and (31) into Eq. (11) yields:

$$\mathcal{J}_{\text{align}} \doteq \sum_{i,j} \log C_{PS}(u_\theta(x_j)) + u_\theta(x_j) \log p(y_\theta(x_j)^T f_\phi(x_i)), \quad \text{(33)}$$

where $\log p(\cdot) := \log(1 + \cdot)$. Similar to Eqs. (1) and (22), this loss maximizes the cosine similarity of features from different models, but the similarity term is weighted by $\kappa$.

A novel method referred to as variational inference SimSiam (VI-SimSiam) that optimizes Eq. (33) is proposed. The architecture and pseudo code are shown in Fig. 2 and Alg. 1, respectively, where the number of modalities is $M = 2$ and the single-modal comparisons are ignored.

### 6. Experiments

We evaluate VI-SimSiam for two aspects: performance and method of uncertainty prediction. First, we compare the performance of VI-SimSiam and SimSiam. We perform a linear evaluation of these methods on ImageNet100 [11].

Second, we investigate representation uncertainty. We qualitatively evaluate representation uncertainty by comparing input images and the predicted uncertainty parameter $\kappa$. Then, we examine the relationship between uncertainty and classification accuracy. We also study how DINO predicts representation uncertainty.

#### 6.1. Linear evaluation

We conduct self-supervised pretraining with ImageNet100 dataset without labels to learn image representations using SimSiam and VI-SimSiam at 50, 100, 200, and
Algorithm 1: Pseudocode of VI-SimSiam, PyTorch-like

```python
# backbone: ResNet-backbone
# f: Projector
# h, u: Predictor for mu and kappa

# Definition of power spherical dist.
class PSd(Distribution):
    def __init__(self, mu, kappa):
        ...

for x in loader:  # Load a minibatch x
    x1, x2 = aug(x), aug(x)  # Apply augmentation
    y1, y2 = backbone(x1), backbone(x2)
    z1, z2 = f(y1), f(y2)  # Project
    mu1, mu2 = h(z1), h(z2)  # Predict mu
    kappa1, kappa2 = u(y1), u(y2)  # Predict kappa
    p1, p2 = PSd(mu1, kappa1), PSd(mu2, kappa2)

    # Stop gradient & compute loss
    z1, z2 = z1.detach(), z2.detach()
    L = -p1.log_prob(z2) - p2.log_prob(z1)
    L.mean().backward()  # Back-prop.
    update(backbone, f, h, u)  # SGD update
```

Table 2: Top-1 accuracies of linear evaluation on ImageNet100. Each setting is repeated in triplicate to compute mean and standard derivation. We report the result as "mean ± std."

<table>
<thead>
<tr>
<th>Method</th>
<th>50 epochs</th>
<th>100 epochs</th>
<th>200 epochs</th>
<th>500 epochs</th>
</tr>
</thead>
<tbody>
<tr>
<td>SimSiam</td>
<td>41.07 ± 0.99</td>
<td>63.86 ± 1.14</td>
<td>78.49 ± 0.89</td>
<td>81.61 ± 0.23</td>
</tr>
<tr>
<td>VI-SimSiam</td>
<td><strong>63.89 ± 2.74</strong></td>
<td><strong>73.78 ± 0.59</strong></td>
<td>76.31 ± 0.32</td>
<td><strong>77.49 ± 0.42</strong></td>
</tr>
</tbody>
</table>

500 epochs. Then, we trained a linear classifier on frozen representations on the training set of the dataset with the labels. Finally, we evaluated it in the test set of the dataset. The implementation details are reported in Appx. B. We used Top-1 accuracy as an evaluation metric.

Table 2 shows Top-1 accuracy on the validation split of ImageNet100. VI-SimSiam achieves a competitive result to SimSiam in several epochs. VI-SimSiam significantly outperforms SimSiam, especially when the number of epochs is less, e.g., 50 and 100.

6.2. Qualitative analysis of uncertainty

We evaluate uncertainty estimation qualitatively by comparing an input image to the predicted concentration $\kappa$, a parameter related to the uncertainty. We use images from the validation split of ImageNet100. We use the model pre-trained with ImageNet100 on 100 epochs. Fig. 1 shows the images for which $\kappa$ is in the top 1% and those for which $\kappa$ is in the bottom 1%. When $\kappa$ is low, i.e., the uncertainty of the latent variable is high, there are few noticeable features in the input images. This result shows that our method can estimate high uncertainty for semantically uncertain images. Additional image examples estimated to have high or low uncertainty are shown in Appx. L.

6.3. Quantitative analysis of uncertainty

We investigate the effects of uncertainty on a classification task with ImageNet100 dataset. We evaluate the relationship between uncertainty and Top-1 accuracy using the AUROC (Area Under the Receiver Operating Characteristics Curve) metrics following as per Ref. [13]. Fig. 4 shows ROC and AUROC when the $\kappa$ predicted by VI-SimSiam is used as the confidence score. We use the VI-SimSiam-trained 500 epochs on ImageNet100. We use two classifiers – a linear classifier and a k-nearest-neighbor (KNN) classifier [56]. The AUROC score of 0.72 by the linear classifier is greater than the chance rate of 0.5 ($p < 0.01$). Furthermore, the AUROC score of KNN is 0.76. The score of the linear classifier is lower than that of the KNN, possibly due to the presence of epistemic uncertainty in the linear classifier itself. These results suggest we can estimate the difficulty of classifying an image based on uncertainty-related parameters without training the classification model or having the correct labels.

6.4. Relationship between DINO and uncertainty

We consider that DINO can estimate uncertainty because it fully estimates distributions, similar to VI-SimSiam. In this section, we discuss how DINO expresses uncertainty. We assume that the entropy of the latent variable, related to the variability of the distribution such as $\kappa$, has a relationship with uncertainty.

To discuss this, we evaluate the relationship between its entropy and Top-1 accuracy using the AUROC metrics. We use the negative entropy of representation as the confidence score. The implementation details are reported in Appx. B. We use ImageNet100 dataset, and the number of epochs is set to 200. The results are shown in Fig 5. The AUROC score of 0.67 by linear classification is greater than the chance rate of 0.5 ($p < 0.01$).

Fig. 6 shows the scatter plot of $\kappa$ by VI-SimSiam and the entropy of representation by DINO. Their correlation

5How to calculate the p-value is mentioned in Appx. J.
Figure 5: ROC and AUROC when the negative entropy predicted by DINO is used as the confidence score.

Figure 6: Scatter plot of $\kappa$ of VI-SimSiam and entropy of DINO. The solid red line is the linear regression line. Their correlation coefficient is 0.5459.

Table 3: Mean and standard derivation of $\kappa$ for each augmentation. The term "normal" means without augmentations.

<table>
<thead>
<tr>
<th>Augmentation</th>
<th>Mean $\kappa$</th>
<th>Std Dev $\kappa$</th>
</tr>
</thead>
<tbody>
<tr>
<td>normal</td>
<td>852.63 ± 143.69</td>
<td>836.05 ± 147.96</td>
</tr>
<tr>
<td>blur</td>
<td>843.94 ± 144.59</td>
<td>846.14 ± 140.33</td>
</tr>
<tr>
<td>color jitter</td>
<td>812.26 ± 159.05</td>
<td></td>
</tr>
<tr>
<td>grayscale</td>
<td>843.94 ± 144.59</td>
<td>846.14 ± 140.33</td>
</tr>
<tr>
<td>random crop</td>
<td>812.26 ± 159.05</td>
<td></td>
</tr>
</tbody>
</table>

The results show that entropy is highly related to $\kappa$ ($p < 0.01$). Therefore, the entropy of representation by DINO seems to be related to uncertainty.

6.5. Relation to image augmentation

We investigate how image augmentations affect uncertainty. We use the validation dataset of ImageNet100. We also prepare image augmentations for blur, color jitter, grayscale, and random crop. The random crop scale for random crop is set from 0.05 to 1.0. We perform five augmentations for each image and calculate the average concentration $\kappa$ estimated by the pre-trained model for each augmentation. The results are shown in Table 3. The variance of the $\kappa$ of the random crop is greater, and its mean value is less than those of other augmentations. Fig. 7 shows examples of images that applied random crop. This result shows $\kappa$ of images excluding important objects by cropping is low. Therefore, random crop is considered to have a more significant impact on $\kappa$ than the other augmentations.

6.6. Limitation

Performance of Linear evaluation. In Sec 6.1, VI-SimSiam underperforms at 200 and 500 epochs, when compared with SimSiam. When the cosine similarity is low during training, SimSiam only learns to increase the cosine similarity, while VI-SimSiam learns to also decrease $\kappa$. It is assumed that VI-SimSiam underperforms at greater epochs because it is not trained to increase cosine similarity for inputs with representations that are difficult to predict. We compare the cosine similarity of SimSiam and VI-SimSiam in Appx. K. Our method would be more effective if an easy solution to this issue is proposed.

Utilization of Representation Uncertainty. Our future work involves proposing novel applications using representation uncertainty. One example of such use of uncertainty is the selection of data, the prototype results of which are reported in Appx. E.

7. Conclusion

In this work, we clarify the theoretical relationship between variational inference and SSL. Additionally, we propose variational inference SimSiam (VI-SimSiam), which could model the latent variable’s uncertainty. We investigate the estimated uncertainty parameter $\kappa$ from various perspectives. We derive the relationship between $\kappa$ and input images and between $\kappa$ and classification accuracy. We also experimentally demonstrate that uncertainty could be estimated even when the latent variable follows the categorical distribution.
References


[19] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. In CVPR, pages 16000–16009, 2022.


