Enhancing Contrastive Learning with Variable Similarity

Haowen Cui[1], **Shuo Chen**[2,3*], **Jun Li**[1], **Jian Yang**[1*]
[1]PCA Lab,[†] Nanjing University of Science and Technology, China
[2]School of Intelligence Science and Technology, Nanjing University, China
[3]Center for Advanced Intelligence Project, RIKEN National Science Institute, Japan

Abstract

Contrastive learning has achieved remarkable success in self-supervised learning by pretraining a generalizable feature representation based on the augmentation invariance. Most existing approaches assume that different augmented views of the same instance (i.e., the *positive pairs*) remain semantically invariant. However, the augmentation results with *varying extent* may introduce semantic discrepancies or even content distortion, and thus the conventional (pseudo) supervision from augmentation invariance may lead to misguided learning objectives. In this paper, we propose a novel method called Contrastive Learning with Variable Similarity (CLVS) to accurately characterize the intrinsic similarity relationships between different augmented views. Our method dynamically adjusts the similarity based on the augmentation extent, and it ensures that strongly augmented views are always assigned lower similarity scores than weakly augmented ones. We provide a theoretical analysis to guarantee the effectiveness of the variable similarity in improving model generalizability. Extensive experiments demonstrate the superiority of our approach, achieving gains of 2.1% on ImageNet-100 and 1.4% on ImageNet-1k compared with the state-of-the-art methods.

1 Introduction

Learning effective feature representations [1, 2, 3] is a fundamental task in machine learning, with profound implications for various applications, including image classification [4, 5], object detection [6, 7], and segmentation [8, 9]. In recent years, self-supervised learning has emerged as a leading paradigm for unsupervised visual representation learning [10, 11, 12, 13]. Among various pretext tasks in self-supervised learning, contrastive learning constructs self-supervisory signals by treating different augmented views of the same image as positive pairs, enabling the extraction of high-quality feature representations without extensive labeled data. This paradigm has achieved remarkable success, demonstrating its potential to bridge the gap between supervised and unsupervised learning.

*Corresponding Authors
†PCA Lab, Key Lab of Intelligent Perception and Systems for High-Dimensional Information of Ministry of Education, School of Computer Science and Engineering, Nanjing University of Science and Technology

39th Conference on Neural Information Processing Systems (NeurIPS 2025).

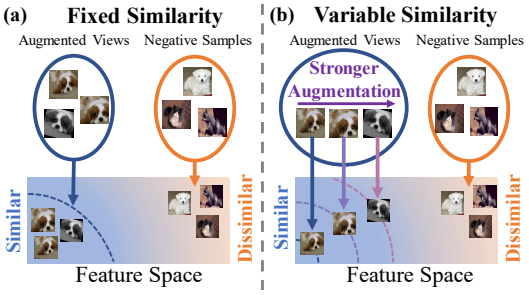


Figure 1: Illustration of different similarity measure in contrastive learning. (a) Fixed maximum similarity among augmented views of the same instance. (b) Variable similarity (ours) among augmented views of the same instance.

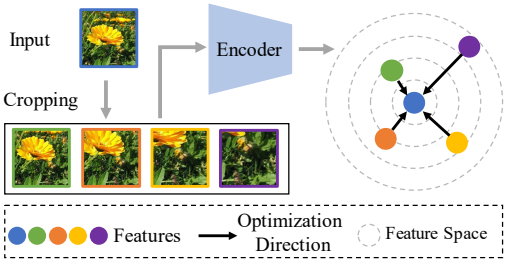


Figure 2: Example of learning framework with the cropping augmentation in contrastive learning. Despite the semantic inconsistency between different augmented views, current contrastive learning method equally optimize them towards the direction with the maximum similarity.

As illustrated in Fig. 1(a), the core mechanism of contrastive learning is to maximize the similarity of embeddings between different views generated by data augmentations. These augmented views share the same underlying semantic concepts, enabling the model to learn invariant representations and thereby improving its generalization ability [10, 14, 15]. However, the assumption that all augmented views are equally valid representations of the original data is still flawed, as the semantic consistency between these views cannot always be guaranteed. As shown in Fig. 2, over-cropping augmentation can produce views containing only background, leading to significant semantic discrepancies between these views and others. This inconsistency may force the model to optimize in the wrong direction, compromising its ability to learn meaningful representations.

In fact, data augmentations inherently induce a spectrum of semantic consistency between views, rather than a binary state of similar or dissimilar. As shown in Fig. 2, the progressive loss of critical features becomes more pronounced as the extent of cropping augmentation increases. This observation suggests that the similarity usually varies dynamically, rather than remaining fixed. Inspired by the above observation, we argue that the similarity between augmented views in contrastive learning should be *variable* according to the extent of applied data augmentations, as illustrated in Fig. 1(b). This variability allows the model to reflect the semantic variation introduced by data augmentations. In particular, the stronger data augmentation leads to more significant information degradation, resulting in a lower similarity compared with weakly augmented views.

In this paper, we propose Contrastive Learning with Variable Similarity (CLVS), which models the similarity between views as a variable determined by their augmentation parameters. Specifically, we first build an augmentation-aware module to predict the variable similarity between two augmented views based on their augmentation parameters. Then, we introduce an alignment objective that constrains the similarity between augmented views to align with the predicted variable similarity, thereby guiding the representations to reflect varying semantic relations. Additionally, the loss function will penalize cases when the similarity of strongly augmented views exceeds that of weakly augmented views. To further validate our approach, we provide a theoretical analysis demonstrating that the generalization error bound of our method can be effectively shrunk through the use of variable similarity. Our method is a general-purpose technique that can be easily integrated into many existing contrastive learning frameworks to enhance their performance. When applied to MoCo [16] and SimSiam [12], our approach achieves significant improvements of 6.1% and 5.9% on the ImageNet-100 dataset, demonstrating its strong generalizability and practical effectiveness.

In summary, our main contributions are as follows:

- We enhance contrastive learning by introducing the variable similarity to capture variations between augmented views, supported by an augmentation-aware module that estimates accurate semantic similarity between these views based on their augmentation parameters.
- We theoretically prove that the statistical variance of similarity decreases with the utilization of variable similarity, thereby resulting in a reduction in the error bound of the generalization.
- Experimental results conducted on standard benchmarks demonstrate the superiority of our method, surpassing the state-of-the-art methods by 2.1% on ImageNet-100 and 1.4% on ImageNet-1k, respectively.

2 Related Works

2.1 Contrastive Learning

Contrastive learning has recently emerged as a leading paradigm in self-supervised learning [10, 11, 12, 13, 17, 18, 19], gradually closing the performance gap with supervised learning. The core principle of contrastive learning is to pull together the embedding features of positive pairs, which are different augmented views of the same instance. This aligns with the broader objective of metric learning [20, 21, 22, 23, 24, 25], where the goal is to learn representations that preserve semantic similarity in the embedding space. A crucial challenge in contrastive learning is to prevent the representation collapse, where all samples are mapped to the same representation, resulting in a loss of discriminative capability. Several negative-used contrastive methods [10, 11, 26, 16, 27] tackle this issue using the InfoNCE criterion [28]. This criterion pulls positive pairs together while pushing negative pairs apart. Negative-free methods [13, 12, 29] mitigate the collapse problem by leveraging only positive pairs and introducing an asymmetric design, i.e., the stop-gradient mechanism [30]. Additionally, some works [17, 18, 31, 32] aim to alleviate the collapse problem by maximizing the information content.

2.2 Augmentation Technique

Data augmentation plays a crucial role in many representation learning tasks. It improves the quality and diversity of positive sample pairs, enhances the robustness of learned representations [14, 33], and can help mitigate collapse [34]. Recent studies have explored various strategies to leverage data augmentation more effectively. For instance, AugSelf [35] preserves augmentation-aware information through an auxiliary model that predicts the difference between augmentation parameters of augmented views. HAIEV [36] argues that different augmentations should be treated unequally and proposes computing contrastive loss at different network layers to achieve a broader distribution of augmentation invariance. LoGo [37] simultaneously incorporates global and local views, encouraging local views to have distinct representations. EquiMod [38] introduces equivariance into contrastive learning, aiming to better capture the augmentation information and improve the robustness of learned representations. Other works focus on leveraging stronger augmentations: CLSA [39] aims to minimize the distribution divergence between weak and strong augmented views, while RényiCL [40] generalizes representation learning by utilizing Rényi divergence as the learning objective. CoCor [41] focuses on exploiting the strength of the composite data augmentation quantified by frequency, but it does not explicitly model the augmentation parameters. As a result, it cannot ensure the variation within the same augmentation type or ensure that the similarity between augmented views aligns with the extent of augmentation.

While these methods typically treat all positive pairs as equally similar, they usually ignore the fact that different augmented views may preserve varying levels of semantic content. To address this limitation, we propose the concept of variable similarity, which better captures the nuanced relationships between views with diverse augmentations.

3 Method

3.1 Preliminaries of Contrastive Learning

Contrastive learning aims to train a generalizable feature encoder $\boldsymbol{f} : \mathbb{R}^n \to \mathbb{R}^d$ that maps input data from the original n-dimensional space to a d-dimensional feature space. Given an input instance $\boldsymbol{x} \in \mathbb{R}^n$ sampled from the dataset $\mathbb{D}$ and a distribution of data augmentations $\mathcal{T}$, the training objective is to maximize the similarity between the embeddings $\boldsymbol{z}_1 = \boldsymbol{f}(t_1(\boldsymbol{x}))$ and $\boldsymbol{z}_2 = \boldsymbol{f}(t_2(\boldsymbol{x}))$ of two augmented views $t_1(\boldsymbol{x})$ and $t_2(\boldsymbol{x})$, where $t_1, t_2 \sim \mathcal{T}$. This can be formulated as minimizing the following general objective:

$$\mathcal{L}_{con}(\boldsymbol{f}) = \mathbb{E}_{x\sim\mathbb{D},t_{1,2}\sim\mathcal{T}}[\ell(\boldsymbol{z}_1, \boldsymbol{z}_2)], \tag{1}$$

where $\ell(\cdot, \cdot)$ is an empirical loss to evaluate the inconsistency between $\boldsymbol{z}_1$ and $\boldsymbol{z}_2$. For the traditional contrastive methods [10, 11], the InfoNCE criterion [28] is typically used as the loss function:

$$\ell(\boldsymbol{z}_1, \boldsymbol{z}_2) = -log\frac{\mathrm{e}^{sim(\boldsymbol{z}_1\cdot\boldsymbol{z}_2)/\tau}}{\sum_i^N \mathrm{e}^{sim(\boldsymbol{z}_1\cdot\boldsymbol{z}_i)/\tau}}, \tag{2}$$

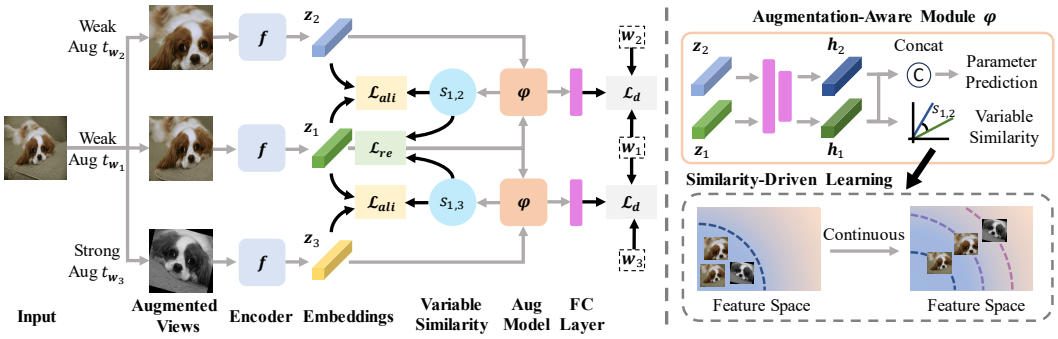


Figure 3: Overview of the CLVS. The encoder $\boldsymbol{f}$ extracts the embedding $\boldsymbol{z}_i$ of augmented views. The augmentation-aware module $\boldsymbol{\varphi}$ predicts the variable similarity between two augmented views. The similarity between the embeddings of two augmented views is constrained by $\mathcal{L}_{ali}$ to approximate the predicted variable similarity during training. $\mathcal{L}_{re}$ enforces the similarity to the weakly augmented view $s_{1,2}$ larger than that to the strongly augmented view $s_{1,3}$. By incorporating $\mathcal{L}_d$, the augmentation-aware module $\boldsymbol{\varphi}$ gains the ability to perceive augmentations.

where $sim(\cdot,\cdot)$ is the similarity measurement function, such as cosine similarity. τ is a temperature hyperparameter and N is the number of negative samples in a mini-batch. This encourages the model to pull positive pairs closer while pushing negative pairs apart in the embedding space.

3.2 Variable Similarity in Contrastive Learning

To address the limitation of enforcing a fixed similarity for positive pairs in contrastive learning, we propose CLVS. Unlike traditional methods, CLVS dynamically adjusts the similarity between augmented views based on the parameters of applied augmentations. This dynamic adjustment enables the model to better capture semantic relationships and produce more robust representations.

Augmentation-Aware Learning. Our key idea is to make the similarity between augmented views of an input instance $\boldsymbol{x}$ consistent with the extent of their augmentations. Let $\boldsymbol{w}_i = (\boldsymbol{w}_i^1, ..., \boldsymbol{w}_i^c)$ denote the augmentation parameters for transformation $t_{\boldsymbol{w}_i}$, where the vector $\boldsymbol{w}_i^j$ represents the parameter of the j-th augmentation type (e.g., cropping, color jittering), and c is the number of augmentation types. Inspired by [35], we introduce an auxiliary task to predict the differences in augmentation parameters. Specifically, for the j-th augmentation type, the difference between $t_{\boldsymbol{w}_1}$ and $t_{\boldsymbol{w}_2}$ is defined as $\boldsymbol{d}_{1,2}^j = \boldsymbol{w}_1^j - \boldsymbol{w}_2^j$. To make the learned representations sensitive to these differences, we build an augmentation-aware module $\boldsymbol{\varphi} = (\boldsymbol{\varphi}^1, ..., \boldsymbol{\varphi}^c)$, where each $\boldsymbol{\varphi}^j : \mathbb{R}^d \to \mathbb{R}^m$ is implemented as a Multi-Layer Perceptron (MLP) for generating augmentation-aware embeddings. Here, m is the embedding dimensionality. Each $\boldsymbol{\varphi}^j$ projects the feature embeddings $\boldsymbol{z}_1$ and $\boldsymbol{z}_2$ (produced by the encoder $\boldsymbol{f}$) into an augmentation-aware latent space, yielding $\boldsymbol{h}_1^j = \boldsymbol{\varphi}^j(\boldsymbol{z}_1)$ and $\boldsymbol{h}_2^j = \boldsymbol{\varphi}^j(\boldsymbol{z}_2)$. The concatenated embedding of $\boldsymbol{h}_1^j$ and $\boldsymbol{h}_2^j$ is then fed into a predictor $\boldsymbol{\psi}^j$ to estimate the parameter difference $\hat{\boldsymbol{d}}_{1,2}^j$. This prediction task encourages the features to retain augmentation-aware information, allowing them to better reflect the extent of augmentations, and thereby laying the foundation for variable similarity estimation. The learning process for this auxiliary task is summarized by:

$$\mathcal{L}_d(\boldsymbol{f}, \boldsymbol{\varphi}, \boldsymbol{\psi}) = \sum\nolimits_{j=1}^{c} \ell_j(\hat{\boldsymbol{d}}_{1,2}^j, \boldsymbol{d}_{1,2}^j) = \sum\nolimits_{j=1}^{c} \ell_j(\boldsymbol{\psi}^j(\text{concat}(\boldsymbol{\varphi}^j(\boldsymbol{z}_1), \boldsymbol{\varphi}^j(\boldsymbol{z}_2))), \boldsymbol{d}_{1,2}^j). \quad (3)$$

The loss function ℓ_j dynamically adjusts depending on the type of data augmentation. [3]

Variable Similarity Alignment. Building on the above auxiliary task, we calculate the variable similarity through an adaptive selection mechanism governed by $\boldsymbol{\varphi}$. In this paper, we use $\hat{s}_{i,j}$ to denote the predicted variable similarity, and $s_{i,j}$ for the similarity between encoder outputs of two augmented views. For each augmentation type j, we calculate the similarity between $\boldsymbol{h}_1^j$ and $\boldsymbol{h}_2^j$ as:

[3]For details, $\ell_j = ||\hat{\boldsymbol{d}}_{1,2}^j - \boldsymbol{d}_{1,2}^j||_2$ for augmentations with parameters, e.g., cropping, color jittering. $\ell_j = \boldsymbol{d}_{1,2}^j \log(\hat{\boldsymbol{d}}_{1,2}^j) + (1 - \boldsymbol{d}_{1,2}^j) \log(1 - \hat{\boldsymbol{d}}_{1,2}^j)$ for augmentations without parameters, e.g., flipping, grayscaling.

$\hat{s}^j_{1,2} = sim(\boldsymbol{h}^j_1, \boldsymbol{h}^j_2) = \boldsymbol{\varphi}^j(\boldsymbol{z}_1) \cdot \boldsymbol{\varphi}^j(\boldsymbol{z}_2)/(||\boldsymbol{\varphi}^j(\boldsymbol{z}_1)||_2||\boldsymbol{\varphi}^j(\boldsymbol{z}_2)||_2)$, i.e., $sim(\cdot,\cdot)$ represents cosine similarity, where $\|\cdot\|_2$ denotes the ℓ_2 normalization. This is motivated by the intuition that smaller augmentation parameter differences indicate more consistent transformations, and thus a higher semantic similarity. Since different types of augmentation have varying impacts on the similarity, we aggregate the similarities across all augmentation types. To ensure that the variable similarity captures the most significant discrepancies between views, we calculate the minimum similarity across all augmentation types. The variable similarity between two augmented views is defined as:

$$\hat{s}_{1,2} = \min_{j\in\{1,\ldots,c\}} sim(\boldsymbol{\varphi}^j(\boldsymbol{z}_1), \boldsymbol{\varphi}^j(\boldsymbol{z}_2)). \tag{4}$$

To integrate the above predicted variable similarity into the contrastive learning framework, we align the cosine similarity between the embeddings of two augmented views with the predicted variable similarity via an ℓ_2 loss:

$$\mathcal{L}_{ali}(\boldsymbol{f},\boldsymbol{\varphi}) = \mathbb{E}_{\boldsymbol{x},t_{\boldsymbol{w}_1},\boldsymbol{w}_2}\left[\ell_{ali}(\hat{s}_{1,2}, s_{1,2})\right] = \mathbb{E}_{\boldsymbol{x},\,t_{\boldsymbol{w}_1},\boldsymbol{w}_2}\left[\left\|\min_{j\in\{1,\ldots,c\}} sim(\boldsymbol{\varphi}^j(\boldsymbol{z}_1),\boldsymbol{\varphi}^j(\boldsymbol{z}_2)) - sim(\boldsymbol{z}_1,\boldsymbol{z}_2)\right\|_2^2\right]. \tag{5}$$

By minimizing $\mathcal{L}_{\text{ali}}$, the model learns to adaptively adjust the similarity objective for each positive pair based on the predicted variable similarity, addressing semantic inconsistencies caused by data augmentations.

Relative Similarity Constraint. To ensure the validity of the predicted variable similarity, we introduce a constraint based on the observation that stronger augmentations cause greater information degradation, thereby reducing the semantic consistency between augmented views. Specifically, the similarity between a view and its strongly augmented version should be lower than its similarity to a weakly augmented view. Given a stronger augmentation t_{w_3}, the relative similarity constraint can be formalized as follows:

$$\begin{aligned}\mathcal{L}_{re}(\boldsymbol{f},\boldsymbol{\varphi}) &= \mathbb{E}_{x,t_{w_1,w_2,w_3}}\left[\ell_{con}(\hat{s}_{1,2},\hat{s}_{1,3})\right] \\ &= \mathbb{E}_{x,t_{w_1,w_2,w_3}}\{\max[0, \min_{j\in 1,\ldots,c}(sim(\boldsymbol{\varphi}^j(\boldsymbol{z}_1),\boldsymbol{\varphi}^j(\boldsymbol{z}_3))) - \min_{j\in 1,\ldots,c}(sim(\boldsymbol{\varphi}^j(\boldsymbol{z}_1),\boldsymbol{\varphi}^j(\boldsymbol{z}_2)))]\}.\end{aligned} \tag{6}$$

Here, $\hat{s}_{1,3}$ represents the predicted variable similarity between the weakly augmented view $t_{w_1}(x)$ and the strongly augmented view $t_{w_3}(x)$. The max operator in Eq. (6) ensures that the loss is only activated when the similarity to a strongly augmented view exceeds that to a weakly augmented view, thus enforcing the desired relationship between augmentation strength and semantic consistency.

The overview of CLVS is shown in Fig. 3. The total training loss consists of four components: the base contrastive loss, the parameter prediction loss, the alignment loss for variable similarity, and the constraint loss for augmentation consistency. Formally, the total loss is defined as:

$$\mathcal{L}_{total} = \mathcal{L}_{con}(\boldsymbol{f}) + \omega\mathcal{L}_d(\boldsymbol{f},\boldsymbol{\varphi},\boldsymbol{\psi}) + \lambda\mathcal{L}_{ali}(\boldsymbol{f},\boldsymbol{\varphi}) + \gamma\mathcal{L}_{re}(\boldsymbol{f},\boldsymbol{\varphi}), \tag{7}$$

where ω, λ, and γ are positive weight coefficients that balance the contributions of different loss terms. By combining these losses, our method preserves the core objective of contrastive learning while addressing the limitation caused by information degradation during data augmentation, resulting in more robust and semantically meaningful representations.

3.3 Theoretical Analysis of Variable Similarity

In this section, we aim to demonstrate that our proposed learning algorithm enhances the Generalization Error Bound (GEB) [42] compared to traditional contrastive learning methods. The GEB typically characterizes how well a model trained on empirical data performs on unseen samples. For self-supervised learning, although the learned encoders are used in different recognition tasks, such an error bound of the learning objective can still provide a quantitative result to evaluate the reliability of the encoder on unseen test data. This is because that the lower generalization error is expected to bring about the smaller InfoNCE loss on the test data, and thus the corresponding feature discriminability during test phase can be ensured. Here, we establish that the proposed new loss terms $\mathcal{L}_d$, $\mathcal{L}_{ali}$, and $\mathcal{L}_{re}$ contribute to tightening the GEB, thereby validating the efficacy of our method.

To provide a more detailed understanding, we consider the underlying data distribution $\mathscr{D}$, and introduce the expected risk, defined as $\widehat{\mathcal{L}}_{total}(\boldsymbol{f},\boldsymbol{\varphi};\mathscr{D}) = \mathbb{E}_{\{\boldsymbol{t}_i|\boldsymbol{t}_i\sim\mathscr{D}\}_{i=1}^N}[\mathcal{L}_{total}(\boldsymbol{f},\boldsymbol{\varphi};\{\boldsymbol{t}_i\}_{i=1}^N)]$. The expected risk represents the true objective we aim to minimize, while the empirical risk $\mathcal{L}_{total}(\boldsymbol{f},\boldsymbol{\varphi};\mathscr{D})$

Table 1: Top-1 accuracies of linear evaluation (%). All compared methods with a ResNet-50 encoder are pretrained for 200 epochs on the ImageNet-100 dataset.

Method	Batch Size	Top-1
MoCo [16]	256	78.8
SimSiam [12]	256	79.1
AugSelf [35]	256	80.5
RényiCL [40]	256	82.1
EquiMod [38]	256	83.9
CoCor [41]	256	83.7
GCA [44]	256	72.4
INTL [32]	256	83.9
MoCo+CLVS	256	84.9 (+6.1)
SimSiam+CLVS	256	**86.0** (+5.9)

Table 2: Top-1 accuracies of linear evaluation (%). All compared methods with a ResNet-50 encoder are pretrained for 200 epochs on the ImageNet-1k dataset.

Method	Batch Size	Top-1
SimCLR [10]	4096	68.3
MoCo [16]	256	67.5
SimSiam [12]	256	70.0
AugSelf [35]	256	69.4
SwAV [26]	4096	69.1
EquiMod [38]	256	67.6
GCA [44]	256	56.1
INTL [32]	256	69.9
MoCo+CLVS	256	69.9 (+2.4)
SimSiam+CLVS	256	**71.4** (+1.4)

Table 3: Linear evaluation accuracies (%) on various datasets. All compared methods with a ResNet-50 encoder are pretrained for 200 epochs on the ImageNet-100 dataset.

Method	CIFAR10	CIFAR100	Caltech101	SUN397	Food	Flowers	Pets
MoCo [16]	85.81	64.96	85.78	48.73	62.37	84.01	69.15
SimSiam [12]	86.52	65.98	87.21	49.68	62.67	84.09	69.28
AugSelf [35]	88.10	68.48	88.95	50.65	65.56	88.60	71.93
RényiCL [40]	86.60	64.92	87.21	48.45	61.89	85.33	74.87
EquiMod [38]	87.86	69.59	90.37	52.85	65.71	89.88	75.88
CoCor [41]	86.89	66.36	87.85	49.92	62.48	87.95	76.04
GCA [44]	76.69	51.87	72.24	37.20	44.96	68.21	49.55
INTL [32]	87.55	67.79	88.77	51.62	63.05	87.75	75.22
MoCo+CLVS	87.29 (+1.48)	67.22 (+2.26)	88.93 (+3.15)	52.55 (+3.82)	65.07 (+2.70)	87.48 (+3.47)	**77.65** (+8.50)
SimSiam+CLVS	**89.92** (+3.40)	**71.37** (+5.42)	**91.73** (+4.52)	**54.80** (+5.12)	**68.97** (+6.30)	**91.12** (+7.03)	77.16 (+7.88)

is computed from a finite dataset. The gap between these two quantities, known as the generalization error, reflects the model's ability to generalize beyond the training set.

Theorem 1. *For any $\{\boldsymbol{f}, \boldsymbol{\varphi}\}$ learned from the objective $\mathcal{L}_{total}(\boldsymbol{f}, \boldsymbol{\varphi})$ and any given constant $\delta \in (0, 1)$, we have that with probability at least $1 - \delta$,*

$$|\mathcal{L}_{total}(\boldsymbol{f}, \boldsymbol{\varphi}; \mathscr{D}) - \widehat{\mathcal{L}}_{total}(\boldsymbol{f}, \boldsymbol{\varphi}; \mathscr{D})| \leq C_1/(C_2 + \omega)\log(1 + \rho(\lambda, \gamma))\sqrt{[\ln(2/\delta)]/(2N)}, \tag{8}$$

where $\rho(\lambda, \gamma) = \sum_{i<j}(s_{ij} - \overline{s})/C_N^2 > 0$ is monotonically decreasing w.r.t. λ and γ. Here $C_1/(C_2 + \omega)$ is monotonically decreasing w.r.t. ω, where $C_1, C_2 > 0$ are constants that independent of $\boldsymbol{f}$ and $\boldsymbol{\varphi}$.

From the result presented in the existing work [43], which highlights the relationship between larger data volumes and improved model generalization. This decrease reflects the statistical principle that larger datasets provide more reliable approximations of the underlying data distribution, thereby reducing the generalization gap. Meanwhile, we observe that the error bound becomes tighter as the parameter ω increases. This is because that the augmentation-specific loss effectively constrains the consistency between similarities, thereby enhancing the model generalization ability. More importantly, we can find that such an error bound becomes *tighter* as λ and γ increase. Actually, this is due to the further decrease of the variance $\rho(\lambda, \gamma) = \sum_{i<j}(s_{ij} - \overline{s})/\mathrm{C}_N^2$ induced by the parameters λ and γ. Intuitively, since the similarity values are constrained to be floats between 0 and 1 rather than those extreme values of either 0 or 1, they are more likely located in the central area of $[0, 1]$, and thus the variance is naturally reduced. Notably, our theoretical analysis is task-agnostic and focuses on the generalization behavior of the training objective itself, rather than any specific downstream application.

In summary, the above theorem demonstrates that our method successfully improves the generalization performance of conventional contrastive learning algorithms by leveraging both increased data and the new loss terms.

Table 4: Transfer learning results on VOC and COCO object detection tasks. All compared methods with a ResNet-50 encoder are pretrained for 200 epochs on the ImageNet-100 dataset.

Method	VOC07+12			COCO					
	AP	AP_{50}	AP_{75}	AP	AP_{50}	AP_{75}	AP_s	AP_m	AP_l
MoCo [16]	40.87	69.78	41.41	37.29	56.36	40.14	20.63	41.63	51.54
SimSiam [12]	39.83	67.27	40.09	37.24	56.61	40.25	21.21	41.78	50.26
Augself [35]	46.96	73.88	50.27	37.91	57.10	41.03	21.58	42.44	51.36
RényiCL [40]	41.59	72.76	41.63	37.82	57.16	40.46	<u>21.90</u>	42.37	51.31
EquiMod [38]	32.24	63.17	28.05	35.30	54.95	37.82	18.11	39.65	49.03
CoCor [41]	**54.31**	**80.60**	**59.58**	34.80	56.29	37.05	17.53	39.88	51.74
GCA [44]	30.80	60.80	27.39	34.93	54.04	37.17	20.09	39.05	48.34
INTL [32]	50.47	78.64	54.95	38.14	57.05	41.07	20.46	42.00	52.01
MoCo+CLVS	52.22 (+11.35)	<u>79.43</u> (+9.65)	56.87 (+15.46)	**39.35** (+2.06)	**58.90** (+2.54)	**42.91** (+2.77)	**22.06** (+1.43)	**44.02** (+2.39)	**54.75** (+3.21)
SimSiam+CLVS	<u>53.20</u> (+13.37)	79.13 (+11.86)	<u>58.39</u> (+18.30)	<u>38.24</u> (+1.00)	<u>57.51</u> (+0.90)	<u>41.37</u> (+1.12)	21.71 (+0.50)	<u>42.58</u> (+0.80)	<u>52.52</u> (+2.26)

Table 5: Few-shot classification accuracy with 95% confidenceinterval averaged over 2000 episodes on FC100, CUB200 and Plant Disease. (N, K) denotes N-way K-shot task. All compared methods with a ResNet-50 encoder are pretrained for 200 epochs on the ImageNet-100 dataset.

Method	FC100		CUB200		Plant	
	$(5, 1)$	$(5, 5)$	$(5, 1)$	$(5, 5)$	$(5, 1)$	$(5, 5)$
MoCo [16]	38.84±0.39	54.53±0.38	41.63±0.46	55.96±0.46	68.42±0.49	86.22±0.34
SimSiam [12]	36.04±0.37	53.83±0.39	39.31±0.43	55.14±0.47	71.00±0.50	88.39±0.33
Augself [35]	40.73±0.41	58.41±0.39	42.50±0.44	59.97±0.45	73.66±0.49	90.23±0.31
RényiCL [40]	37.79±0.36	55.75±0.39	42.30±0.47	58.79±0.46	73.51±0.46	89.67±0.31
EquiMod [38]	44.83±0.41	<u>62.76±0.40</u>	44.81±0.46	<u>62.92±0.46</u>	<u>77.21±0.45</u>	91.16±0.25
GCA [44]	30.34±0.33	44.68±0.37	37.61±0.41	52.68±0.45	64.63±0.51	86.03±0.33
INTL [32]	<u>44.96±0.40</u>	62.49±0.40	<u>44.79±0.48</u>	62.27±0.48	77.05±0.46	<u>91.92±0.28</u>
MoCo+CLVS	40.28±0.40 (+1.44)	56.26±0.39 (+1.73)	42.77±0.47 (+1.14)	57.98±0.46 (+2.02)	71.16±0.49 (+2.74)	88.54±0.32 (+2.32)
SimSiam+CLVS	**45.30±0.42** (+9.26)	**62.97±0.39** (+9.14)	**47.23±0.48** (+7.92)	**63.57±0.47** (+8.43)	**78.70±0.45** (+7.70)	**92.11±0.28** (+3.72)

4 Experiments

In this section, we first present the implementation details of the experiments. Then, we perform extensive experiments on downstream tasks and compare our method with existing state-of-the-art methods. Finally, we provide ablation studies of the proposed method.

4.1 Implementation Details

We pretrain the standard ResNet-50 [4] encoder on the ImageNet-100 and ImageNet-1k [45] datasets. The encoder is pretrained for 200 epochs with the batch size of 256. Each model in $\boldsymbol{\varphi}$ is implemented with 2 fully connected layers, and each predictor in ψ consists of a single fully connected layer. We set $\omega = 0.5$ following [35], $\lambda = 0.5$, and $\gamma = 1$ in the training loss, respectively. To evaluate the effectiveness of CLVS in modeling variable similarity between positive samples, we adopt both the negative-used contrastive method MoCo [16] and the negative-free method SimSiam [12] as baselines.

For weak augmentations, we employ standard augmentation strategies [12], including *random cropping*, *color jittering*, *horizontal flipping*, *grayscale conversion*, and *gaussian blurring*. For strong augmentations, we integrate *RandAugment* [46] into the weak augmentation, which automates the selection of augmentation types and their magnitudes. The parameters of augmentations are set according to [35].

4.2 Main Results

In this subsection, we present experimental results on various downstream tasks. We compare CLVS with state-of-the-art contrastive learning methods. For clarity, the best results are highlighted in **bold**, while the second-best results are <u>underlined</u>.

Comparison on Linear Evaluation. We initially pretrain CLVS and perform linear evaluation on the ImageNet-100 and ImageNet-1k datasets. The linear evaluation protocol follows the approach of [12]. The experimental results are shown in Tab. 1 and Tab. 2, respectively. Our proposed method achieves

Table 6: Comparison with methods using stronger augmentations. Top-1 linear evaluation accuracies (%) and few-shot classification accuracy (%) are reported. All compared methods with a ResNet-50 encoder are pretrained for 200 epochs on the ImageNet-1k dataset.

Method	Linear evaluation		5-way 1-shot		5-way 5-shot	
	ImageNet	Caltech101	FC100	Plant	FC100	Plant
CLSA [39]	72.2	91.21	34.46	67.39	49.16	86.15
RényiCL [40]	<u>72.6</u>	<u>94.04</u>	<u>36.31</u>	<u>80.68</u>	<u>53.39</u>	<u>94.29</u>
CLVS	**72.9**	**94.60**	**53.33**	**81.04**	**73.14**	**95.17**

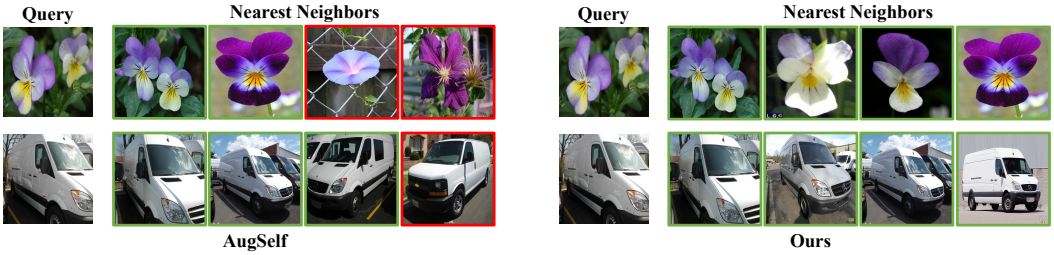


Figure 4: Visualization of retrieval results on the flowers and cars datasets. Red boxes indicate incorrect retrieval results, and green boxes indicate correct retrieval results.

Top-1 accuracy of 86.0% and 71.4% on the ImageNet-100 and ImageNet-1k datasets, significantly surpassing state-of-the-art methods. We also provide a comparison of training time among CLVS and other methods in Appendix C.

We further evaluate the performance of CLVS in transfer learning. We utilize the encoder pretrained on the ImageNet-100 dataset and conduct linear evaluation on 7 datasets: CIFAR10/100 [47], Caltech101 [48], SUN397 [49], Food [50], Flowers [51], and Pets [52]. The experimental results are presented in Tab. 3. CLVS achieves the best performance across all datasets, demonstrating its strong generalization ability and robustness.

Comparison on Object Detection. We also evaluate our method on the object detection task. The encoder pretrained on the ImageNet-100 dataset is converted to a generalized R-CNN detector following [11] and fine-tuned on the PASCAL VOC [53] and COCO [54] datasets. As shown in Tab. 4, CLVS demonstrates competitive performance on both datasets. Although CLVS performs slightly worse than CoCor on the smaller VOC dataset, where CoCor's strict similarity ordering is particularly effective in regularizing feature learning, it demonstrates a clear advantage on the larger and more complex COCO dataset. The performance improvement suggests that our proposed variable similarity can learn more generalizable feature representations.

Comparison on Few-shot Classification. In this experiment, we evaluate the performance of different contrastive learning methods on the few-shot classification task. We conduct experiments on FC100 [55], CUB200 [56], and Plant [57] datasets. Following the experimental setting of [35], we freeze the encoder pretrained on the ImageNet-100 dataset and employ logistic regression for classification. Tab. 5 presents the results of different methods under the 5-way 1-shot and 5-way 5-shot settings. In particular, the CUB and Plant datasets contain fine-grained labels, requiring models to have a strong generalizability to discern subtle differences. CLVS achieves excellent performance across all datasets, which can be attributed to the variable similarity mechanism that enables the model to capture fine-grained features, thereby significantly improving its generalization ability.

Comparison on Retrieval. We validate the effectiveness of our proposed method through visualization experiments. The image features are extracted by the encoder pretrained on the ImageNet-100 dataset, and the top-4 nearest neighbors are retrieved based on the cosine similarity metric. Fig. 4 presents the image retrieval results on the Flowers [51] and Cars [58] datasets. The retrieved images of flowers exhibit a smooth transition in color, starting with light-colored flowers and gradually transitioning to darker shades. Similarly, the retrieved results of cars show a progressive variation in object structure, ranging from close-up views to distant perspectives. These reflect the ability of our variable similarity framework to capture fine-grained visual attributes, ensuring that the retrieval results are not only semantically relevant but also visually coherent.

$\mathcal{L}_{con}$	$\mathcal{L}_d$	$\mathcal{L}_{ali}$	$\mathcal{L}_{re}$	IN-100	CIFAR10	CIFAR100
✓				79.1	86.5	66.0
✓	✓			80.5 (+0.6)	88.1 (+1.6)	68.7 (+2.7)
✓	✓	✓		85.5 (+5.4)	89.8 (+3.3)	71.1 (+5.1)
✓	✓		✓	85.0 (+4.9)	89.4 (+2.9)	70.8 (+4.8)
✓	✓	✓	✓	**86.0** (+5.9)	**89.9** (+3.4)	**71.4** (+5.4)

Table 7: Ablation studies for different loss terms of CLVS. All results are reported on linear evaluation accuracy (%).

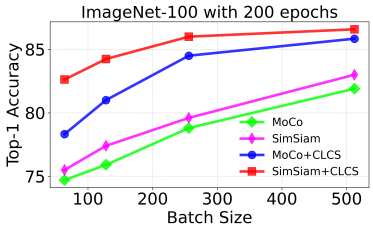


Figure 5: Top-1 accuracies of linear evaluation (%) with different batch sizes.

Table 8: Top-1 accuracies of linear evaluation (%) on various baselines. All methods with a ResNet-50 encoder are pretrained for 200 epochs on the ImageNet-100 dataset.

Method	ImageNet-100	CIFAR10	CIFAR100	Caltech101	SUN397	Food	Flowers	Pets
SimCLR [10]	80.30	85.89	64.31	86.82	48.59	61.95	83.59	68.36
SimCLR + CLVS	**83.48** (+3.18)	**88.15** (+2.26)	**68.45** (+4.14)	**89.18** (+2.36)	**52.63** (+4.04)	**66.13** (+4.18)	**88.03** (+4.44)	**73.15** (+4.79)
BYOL [13]	83.24	87.32	66.95	87.27	49.97	63.89	85.31	75.01
BYOL + CLVS	**85.80** (+2.56)	**89.86** (+2.54)	**71.27** (+4.32)	**89.92** (+2.65)	**53.42** (+3.45)	**67.83** (+3.94)	**89.88** (+4.57)	**77.62** (+2.61)
SwAV [26]	75.48	83.87	60.70	84.52	44.81	58.39	79.83	62.58
SwAV + CLVS	**81.08** (+5.60)	**86.84** (+2.97)	**66.76** (+6.06)	**88.42** (+3.90)	**51.14** (+6.33)	**64.13** (+5.74)	**87.27** (+7.44)	**68.96** (+6.38)

Comparison with Methods Using Stronger Augmentations. To validate the capability of our proposed method in handling stronger augmentations, comparative experiments are conducted with several methods that employ stronger augmentations. Specifically, the proposed method is implemented using RényiCL as the baseline, with the encoder pretrained on the ImageNet-1k dataset. We perform the linear evaluation on the ImageNet-1k and Caltech101 datasets, and few-shot classification on FC100 and Plant datasets. The experimental results shown in Tab. 6 demonstrate that the proposed variable similarity can effectively mitigate the semantic inconsistency caused by stronger augmentations, resulting in a significant boost in linear evaluation performance.

4.3 Ablation Studies

In this subsection, we conduct ablation studies to evaluate the robustness of our method. All encoders are pretrained for 200 epochs on the ImageNet-100 dataset.

Effect of Training Loss. To evaluate the improvement contribution of each loss term, we conduct ablation studies on the ImageNet-100 dataset. The experimental results are shown in Tab. 7. First, we introduce a prediction loss based on the difference in augmentation parameters, which encourages the model to perceive augmentations, which yields a slight improvement. Second, the alignment loss $\mathcal{L}_{ali}$ drives the model to adjust embedding similarities in accordance with the semantic differences induced by data augmentations. This leads to a substantial performance gain, as it directly aligns feature similarities with semantic similarity. Third, the consistency loss $\mathcal{L}_{re}$ constrains the similarity between strongly and weakly augmented views. This further enhances the model's representation consistency under diverse augmentations. Combining all these loss terms leads to the best performance, demonstrating their complementary effects and validating the effectiveness of our proposed method. We also conduct experiments to validate the effectiveness of the augmentation-aware module, which can be found in Appendix C.

Compatibility with Other Baselines. We further integrate our proposed method into additional baseline models, including SimCLR [10], BYOL [13], and SwAV [26]. These methods are pretrained on ImageNet-100 and perform linear evaluation on the datasets mentioned in subsection 4.2. The experimental results in Tab. 8 demonstrate that our method consistently improves performance across all baseline models.

Effect of Batch Size. Given that contrastive learning is typically sensitive to batch size [10], we further investigate the impact of batch size on the performance of our proposed method. The experimental results are shown in Fig. 5. When the batch size increases, the accuracy of linear evaluation also improves. Even with a small batch size, our method maintains strong performance, demonstrating its good robustness.

5 Conclusion

In this paper, we introduced a variable similarity mechanism to extend the contrastive learning framework, which dynamically adjusts the similarity between views based on the extent of data augmentation. The variable similarity is estimated as the similarity between the refined features encoded by the augmentation-aware module, so that the variations induced by augmentations can be fully considered. By introducing an alignment loss for similarity and a relative similarity constraint between strong and weak augmentations, our method successfully addresses the semantic inconsistency caused by data augmentations. We also provided a theoretical analysis demonstrating that the variable similarity enhances the generalization error bound, validating its effectiveness. Experimental results indicated that the variable similarity enables the model to learn more robust feature representations. However, our method is limited to positive samples and does not incorporate negative samples into the variable similarity mechanism. In the future, we plan to extend the variable similarity to all samples to further optimize the contrastive learning framework.

Acknowledgements

This work was supported by National Natural Science Fund of China (Nos. U24A20330, 62361166670 and 62506155), and Provincial Natural Science Fund of Jiangsu (Nos. BK20251985).

References

[1] Jiexi Yan, Lei Luo, Cheng Deng, and Heng Huang. Unsupervised hyperbolic metric learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 12465–12474, 2021.

[2] Jiexi Yan, Erkun Yang, Cheng Deng, and Heng Huang. Metricformer: A unified perspective of correlation exploring in similarity learning. *Advances in Neural Information Processing Systems (NeurIPS)*, 35:33414–33427, 2022.

[3] Jiexi Yan, Lei Luo, Cheng Deng, and Heng Huang. Adaptive hierarchical similarity metric learning with noisy labels. *IEEE Transactions on Image Processing*, 32:1245–1256, 2023.

[4] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2016.

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

[6] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39(6):1137–1149, 2016.

[7] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *European Conference on Computer Vision (ECCV)*, pages 213–229, 2020.

[8] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, pages 234–241, 2015.

[9] Bowen Cheng, Alex Schwing, and Alexander Kirillov. Per-pixel classification is not all you need for semantic segmentation. *Advances in Neural Information Processing Systems (NeurIPS)*, 34:17864–17875, 2021.

[10] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *International Conference on Machine Learning (ICML)*, pages 1597–1607, 2020.

[11] 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 (CVPR)*, pages 9729–9738, 2020.

[12] Xinlei Chen and Kaiming He. Exploring simple siamese representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 15750–15758, 2021.

[13] 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 (NeurIPS)*, 33:21271–21284, 2020.

[14] Yonglong Tian, Chen Sun, Ben Poole, Dilip Krishnan, Cordelia Schmid, and Phillip Isola. What makes for good views for contrastive learning? *Advances in Neural Information Processing Systems (NeurIPS)*, 33:6827–6839, 2020.

[15] Tete Xiao, Xiaolong Wang, Alexei A Efros, and Trevor Darrell. What should not be contrastive in contrastive learning. In *International Conference on Learning Representations (ICLR)*, 2021.

[16] Xinlei Chen, Haoqi Fan, Ross Girshick, and Kaiming He. Improved baselines with momentum contrastive learning. *arXiv preprint arXiv:2003.04297*, 2020.

[17] 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 (ICML)*, pages 12310–12320, 2021.

[18] Adrien Bardes, Jean Ponce, and Yann LeCun. Vicreg: Variance-invariance-covariance regularization for self-supervised learning. In *International Conference on Learning Representations (ICLR)*, 2022.

[19] Xing Wu, Chaochen Gao, Yipeng Su, Jizhong Han, Zhongyuan Wang, and Songlin Hu. Smoothed contrastive learning for unsupervised sentence embedding. In *Proceedings of the International Conference on Computational Linguistics (COLING)*, 2022.

[20] Lei Feng, Senlin Shu, Nan Lu, Bo Han, Miao Xu, Gang Niu, Bo An, and Masashi Sugiyama. Pointwise binary classification with pairwise confidence comparisons. In *International Conference on Machine Learning (ICML)*, pages 3252–3262, 2021.

[21] Songhua Wu, Tongliang Liu, Bo Han, Jun Yu, Gang Niu, and Masashi Sugiyama. Learning from noisy pairwise similarity and unlabeled data. *Journal of Machine Learning Research*, 23(307):1–34, 2022.

[22] Jiexi Yan, Lei Luo, Chenghao Xu, Cheng Deng, and Heng Huang. Noise is also useful: Negative correlation-steered latent contrastive learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 31–40, 2022.

[23] Lei Feng, Senlin Shu, Yuzhou Cao, Lue Tao, Hongxin Wei, Tao Xiang, Bo An, and Gang Niu. Multiple-instance learning from unlabeled bags with pairwise similarity. *IEEE Transactions on Knowledge and Data Engineering*, 35(11):11599–11609, 2023.

[24] Jiexi Yan, Zhihui Yin, Erkun Yang, Yanhua Yang, and Heng Huang. Learning with diversity: Self-expanded equalization for better generalized deep metric learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 19365–19374, 2023.

[25] Jiexi Yan, Cheng Deng, Heng Huang, and Wei Liu. Causality-invariant interactive mining for cross-modal similarity learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 46(9):6216–6230, 2024.

[26] 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 (NeurIPS)*, 33:9912–9924, 2020.

[27] Chun-Hsiao Yeh, Cheng-Yao Hong, Yen-Chi Hsu, Tyng-Luh Liu, Yubei Chen, and Yann LeCun. Decoupled contrastive learning. In *European Conference on Computer Vision (ECCV)*, pages 668–684, 2022.

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

[29] 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 (ICCV)*, pages 9650–9660, 2021.

[30] Yuandong Tian, Xinlei Chen, and Surya Ganguli. Understanding self-supervised learning dynamics without contrastive pairs. In *International Conference on Machine Learning (ICML)*, pages 10268–10278, 2021.

[31] Aleksandr Ermolov, Aliaksandr Siarohin, Enver Sangineto, and Nicu Sebe. Whitening for self-supervised representation learning. In *International Conference on Machine Learning (ICML)*, pages 3015–3024, 2021.

[32] Xi Weng, Yunhao Ni, Tengwei Song, Jie Luo, Rao Muhammad Anwer, Salman Khan, Fahad Shahbaz Khan, and Lei Huang. Modulate your spectrum in self-supervised learning. *International Conference on Learning Representations (ICLR)*, 2024.

[33] Yuning You, Tianlong Chen, Yang Shen, and Zhangyang Wang. Graph contrastive learning automated. In *International Conference on Machine Learning (ICML)*, pages 12121–12132, 2021.

[34] Li Jing, Pascal Vincent, Yann LeCun, and Yuandong Tian. Understanding dimensional collapse in contrastive self-supervised learning. In *International Conference on Learning Representations (ICLR)*, 2022.

[35] Hankook Lee, Kibok Lee, Kimin Lee, Honglak Lee, and Jinwoo Shin. Improving transferability of representations via augmentation-aware self-supervision. *Advances in Neural Information Processing Systems (NeurIPS)*, 34:17710–17722, 2021.

[36] Junbo Zhang and Kaisheng Ma. Rethinking the augmentation module in contrastive learning: Learning hierarchical augmentation invariance with expanded views. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 16650–16659, 2022.

[37] Tong Zhang, Congpei Qiu, Wei Ke, Sabine Süsstrunk, and Mathieu Salzmann. Leverage your local and global representations: A new self-supervised learning strategy. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 16580–16589, 2022.

[38] Alexandre Devillers and Mathieu Lefort. Equimod: An equivariance module to improve visual instance discrimination. In *International Conference on Learning Representations (ICLR)*, 2023.

[39] Xiao Wang and Guo-Jun Qi. Contrastive learning with stronger augmentations. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(5):5549–5560, 2022.

[40] Kyungmin Lee and Jinwoo Shin. Rényicl: Contrastive representation learning with skew rényi divergence. *Advances in Neural Information Processing Systems (NeurIPS)*, 35:6463–6477, 2022.

[41] Zihu Wang, Yu Wang, Zhuotong Chen, Hanbin Hu, and Peng Li. Contrastive learning with consistent representations. *Transactions on Machine Learning Research*, 2024.

[42] Shuo Chen, Gang Niu, Chen Gong, Jun Li, Jian Yang, and Masashi Sugiyama. Large-margin contrastive learning with distance polarization regularizer. In *International Conference on Machine Learning (ICML)*, pages 1673–1683, 2021.

[43] Nikunj Saunshi, Orestis Plevrakis, Sanjeev Arora, Mikhail Khodak, and Hrishikesh Khandeparkar. A theoretical analysis of contrastive unsupervised representation learning. In *International Conference on Machine Learning (ICML)*, pages 5628–5637, 2019.

[44] Zihao Chen, Chi-Heng Lin, Ran Liu, Jingyun Xiao, and Eva L Dyer. Your contrastive learning problem is secretly a distribution alignment problem. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2024.

[45] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 248–255, 2009.

[46] Ekin D Cubuk, Barret Zoph, Jonathon Shlens, and Quoc V Le. Randaugment: Practical automated data augmentation with a reduced search space. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshop*, pages 702–703, 2020.

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

[48] Li Fei-Fei, Rob Fergus, and Pietro Perona. Learning generative visual models from few training examples: An incremental bayesian approach tested on 101 object categories. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshop*, pages 178–178, 2004.

[49] Jianxiong Xiao, James Hays, Krista A Ehinger, Aude Oliva, and Antonio Torralba. Sun database: Large-scale scene recognition from abbey to zoo. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3485–3492, 2010.

[50] Lukas Bossard, Matthieu Guillaumin, and Luc Van Gool. Food-101–mining discriminative components with random forests. In *European Conference on Computer Vision (ECCV)*, pages 446–461, 2014.

[51] Maria-Elena Nilsback and Andrew Zisserman. Automated flower classification over a large number of classes. In *Proceedings of the Indian Conference on Computer Vision, Graphics & Image Processing (ICVGIP)*, pages 722–729, 2008.

[52] Omkar M Parkhi, Andrea Vedaldi, Andrew Zisserman, and CV Jawahar. Cats and dogs. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3498–3505, 2012.

[53] Mark Everingham, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. The pascal visual object classes (voc) challenge. *International Journal of Computer Vision*, 88:303–338, 2010.

[54] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European Conference on Computer Vision (ECCV)*, pages 740–755, 2014.

[55] Boris Oreshkin, Pau Rodríguez López, and Alexandre Lacoste. Tadam: Task dependent adaptive metric for improved few-shot learning. *Advances in Neural Information Processing Systems (NeurIPS)*, 31, 2018.

[56] Catherine Wah, Steve Branson, Peter Welinder, Pietro Perona, and Serge Belongie. The caltech-ucsd birds-200-2011 dataset. 2011.

[57] Sharada P Mohanty, David P Hughes, and Marcel Salathé. Using deep learning for image-based plant disease detection. *Frontiers in plant science*, 7:1419, 2016.

[58] Jonathan Krause, Michael Stark, Jia Deng, and Li Fei-Fei. 3d object representations for fine-grained categorization. In *Proceedings of the IEEE/CVF International Conference on Computer Vision Workshop*, pages 554–561, 2013.

[59] Yang You, Igor Gitman, and Boris Ginsburg. Large batch training of convolutional networks. *arXiv preprint arXiv:1708.03888*, 2017.

[60] Dong C Liu and Jorge Nocedal. On the limited memory bfgs method for large scale optimization. *Mathematical Programming*, 45(1):503–528, 1989.

[61] Ross Girshick. Fast r-cnn. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 1440–1448, 2015.

[62] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 2961–2969, 2017.

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

[64] Xinlei Chen, Saining Xie, and Kaiming He. An empirical study of training self-supervised vision transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 9640–9649, 2021.

[65] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations (ICLR)*, 2019.

[66] Laurens Van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of Machine Learning Research*, 9(11), 2008.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The contributions and scope of this paper are accurately reflected in the abstract and introduction.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: In section 5, we discuss the limitations of our work and identify them as valuable future work.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [Yes]

Justification: The full set of assumptions and complete proofs can be found in the Appendix.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: We claim the details of methods and the experiments settings in our paper.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: Code [Yes] DATA [No]

Justification: We will soon open-source the code, and the datasets used are publicly available for download by anyone.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: All detailed specifications are presented in the section 4 and Appendix B.1.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [No]

   Justification: We lack sufficient computational resources to calculate error bars for every experiment.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).

- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: See Appendix C.7.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: We confirm that our research conducted in the paper conforms with the NeurIPS Code of Ethics.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [NA]

    Justification: Our paper don't have obvious societal impacts currently.

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.
    - If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.

- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [NA]

    Justification: The paper poses no such risks.

    Guidelines:

    - The answer NA means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: We cite and comply with the licenses of the public datasets we use.

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.

- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [NA]

    Justification: The paper does not release new assets.

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigorousness, or originality of the research, declaration is not required.

Answer: [NA]

Justification: The paper does not involve the usage of LLMs.

Guidelines:

- The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy (`https://neurips.cc/Conferences/2025/LLM`) for what should or should not be described.

A Proof

In this section, we present the proof for theory in the paper. We first introduce the following lemma for proving Theorem 1.

Lemma 1. *For independent random variables $t_1, t_2, \ldots, t_n \in \mathcal{T}$ and a given function $\omega : \mathcal{T}^n \to \mathbb{R}$, if $\forall v_i' \in \mathcal{T}$ $(i = 1, 2, \ldots, n)$, the function satisfies*

$$|\omega(t_1, \ldots, t_i, \ldots, t_n) - \omega(t_1, \ldots, t_i', \ldots, t_n)| \leq \rho_i, \tag{9}$$

then for any given $\mu > 0$, it holds that $P\{|\omega(t_1, \ldots, t_n) - \mathbb{E}[\omega(t_1, \ldots, t_n)]| > \mu\} \leq 2\mathrm{e}^{-2\mu^2 / \sum_{i=1}^n \rho_i^2}$.

Proof. We prove Theorem 1 by analyzing the perturbation (i.e., ρ_i in the above Eq. (9)) of the loss function $\mathcal{L}_{\text{emp}}$.

We denote that

$$\omega = \mathcal{L}_{\text{emp}}(\boldsymbol{f}, \boldsymbol{\varphi}; \mathscr{X}) = \frac{1}{N} \sum_{i=1}^{N} -\log \frac{\mathrm{e}^{sim(\boldsymbol{z}_1, \boldsymbol{z}_2)/\gamma}}{\sum_{j=1}^n \mathrm{e}^{sim(\boldsymbol{z}_1, \boldsymbol{z}_j)/\gamma}}, \tag{10}$$

and

$$\widetilde{\omega_r} = \frac{1}{N} \left[\left(\sum_{i \neq r}^{N} -\log \frac{\mathrm{e}^{sim(\boldsymbol{z}_1, \boldsymbol{z}_2)/\gamma}}{\sum_{j=1}^n \mathrm{e}^{sim(\boldsymbol{z}_1, \boldsymbol{z}_j)/\gamma}}, \right) - \log \frac{\mathrm{e}^{sim(\widehat{\boldsymbol{z}}_1, \widehat{\boldsymbol{z}}_2)/\gamma}}{\sum_{j=1}^n \mathrm{e}^{sim(\widehat{\boldsymbol{z}}_1, \widehat{\boldsymbol{z}}_j)/\gamma}} \right], \tag{11}$$

where $(\widehat{\boldsymbol{x}}, \{\widehat{\boldsymbol{x}}_{b_j}\}_{j=1}^n)$ is an arbitrary mini-batch from the sample space. Then we have that

$$\begin{aligned}
&|\omega - \widetilde{\omega_r}| \\
&= \frac{1}{N} \left| \log \frac{\mathrm{e}^{sim(\widehat{\boldsymbol{z}}_1, \widehat{\boldsymbol{z}}_2)/\gamma}}{\sum_{j=1}^n \mathrm{e}^{sim(\widehat{\boldsymbol{z}}_1, \widehat{\boldsymbol{z}}_j)/\gamma}} - \log \frac{\mathrm{e}^{sim(\boldsymbol{z}_1, \boldsymbol{z}_2)/\gamma}}{\sum_{j=1}^n \mathrm{e}^{sim(\boldsymbol{z}_1, \boldsymbol{z}_j)/\gamma}} \right| \\
&\leq \frac{1}{N} \log \left[\frac{\mathrm{e}^{sim(\widehat{\boldsymbol{z}}_1, \widehat{\boldsymbol{z}}_2)/\gamma} (\mathrm{e}^{sim(\boldsymbol{z}_1, \boldsymbol{z}_2)/\gamma} + \sum_{j=1}^n \mathrm{e}^{sim(\widehat{\boldsymbol{z}}_1, \widehat{\boldsymbol{z}}_j)/\gamma})}{\mathrm{e}^{sim(\boldsymbol{z}_1, \boldsymbol{z}_2)/\gamma} (\sum_{j=1}^n \mathrm{e}^{sim(\widehat{\boldsymbol{z}}_1, \widehat{\boldsymbol{z}}_j)/\gamma} + \mathrm{e}^{sim(\widehat{\boldsymbol{z}}_1, \widehat{\boldsymbol{z}}_2)/\gamma})} \right] \\
&\leq \frac{(C_1/(C_2 + \omega)) \log(1 + \sum_{i<j} (s_{ij} - \overline{s})/\mathrm{C}_N^2)}{2N},
\end{aligned} \tag{12}$$

where $C_1, C_2 > 0$ are constants. Meanwhile, we have

$$\begin{aligned}
&\frac{1}{N} \sum_{i=1}^{N} -\log \frac{\mathrm{e}^{sim(\boldsymbol{z}_1, \boldsymbol{z}_2)/\gamma}}{\sum_{j=1}^n \mathrm{e}^{sim(\boldsymbol{z}_1, \boldsymbol{z}_j)/\gamma}} - \mathbb{E} \left(-\log \frac{\mathrm{e}^{sim(\boldsymbol{z}_1, \boldsymbol{z}_2)/\gamma}}{\sum_{j=1}^n \mathrm{e}^{sim(\boldsymbol{z}_1, \boldsymbol{z}_j)/\gamma}} \right) \\
&= \mathcal{L}_{\text{emp}}(\boldsymbol{f}, \boldsymbol{\varphi}; \mathscr{X}) - \widetilde{\mathcal{L}}_{\text{emp}}(\boldsymbol{f}, \boldsymbol{\varphi}; \mathscr{D}).
\end{aligned} \tag{13}$$

By Lemma 1, we let that for all $i = 1, 2, \ldots, N$

$$\rho_i = \frac{(C_1/(C_2 + \omega)) \log(1 + \sum_{i<j} (s_{ij} - \overline{s})/\mathrm{C}_N^2)}{2N}, \tag{14}$$

so that we have

$$\begin{aligned}
&P \left\{ \left| \mathcal{L}_{\text{emp}}(\boldsymbol{f}, \boldsymbol{\varphi}; \mathscr{X}) - \widetilde{\mathcal{L}}_{\text{emp}}(\boldsymbol{f}, \boldsymbol{\varphi}; \mathscr{D}) \right| < (C_1/(C_2 + \omega)) \log(1 + \sum_{i<j} (s_{ij} - \overline{s})/\mathrm{C}_N^2) \sqrt{\frac{\ln(2/\delta)}{2N}} \right\} \\
&= 1 - 2\mathrm{e}^{-2\mu^2 / \sum_{i=1}^N \rho_i^2} \\
&\geq 1 - 2\mathrm{e}^{\frac{-2N(\eta \sqrt{[\ln(2/\delta)]/(2C\lambda N)})^2}{(C_1/(C_2+\omega)) \log(1 + \sum_{i<j} (s_{ij} - \overline{s})/\mathrm{C}_N^2)}} \\
&= 1 - 2\mathrm{e}^{-2N \left(\sqrt{[\ln(2/\delta)]/(2C\lambda N)} \right)^2} \\
&= 1 - 2\mathrm{e}^{-\ln(2/\delta)} \\
&= 1 - \delta,
\end{aligned} \tag{15}$$

where $\eta = \frac{(C_1/(C_2+\omega)) \log(1 + \sum_{i<j} (s_{ij} - \overline{s})/\mathrm{C}_N^2)}{2N}$ and $\mu = \sqrt{[\ln(2/\delta)]/(2N)}$. The proof is completed. □

B Pretraining Setup and Evaluation Protocols

B.1 Pretraining Setup

In the experiments, we integrate CLVS into MoCo and SimSiam. Here, we provide the detailed pretraining setup used for each method:

- MoCo. The learning rate is set to 0.03 and the temperature parameter for contrastive loss is 0.2. The projector consists of 2 MLP layers with an output dimension of 128. The memory queue size is 65536 and the exponential moving average (EMA) parameter is 0.999.
- SimSiam. The learning rate is set to 0.05. The projector and predictor consist of 3 MLP layers and 2 MLP layers with an output dimension of 2048, respectively.

B.2 Linear Evaluation

The linear evaluation protocol on ImageNet-100 and ImageNet-1k datasets follows [12]. Specifically, we freeze the backbone and train a classifier by minimizing the cross-entropy loss using the LARS [59] optimizer.

The linear evaluation protocol for transfer learning follows [35]. The training datasets are split into a train set and a validation set, with 90% for training and the remaining 10% for validation. The representation of 224×224 center-cropped images are extracted by the frozen backbone. The classifier is trained by minimizing the L_2-regularized cross-entropy loss using a L-BFGS [60] optimizer.

B.3 Object Detection

The object detection protocol follows [11]. We train the Faster R-CNN [61] detector with a backbone of ResNet-50-C4 [62]. The detector is fine-tuned end-to-end on all layers for 24000 iterations on the VOC dataset and 180000 iterations on the COCO dataset. During training, the image scale is resized to $[480, 800]$ pixels for VOC and $[640, 800]$ pixels for COCO, while a fixed scale of 800 pixels is used during inference.

B.4 Few-Shot Classification

The few-shot classification protocol follows [35]. We conduct logistic regression using representation extracted by the frozen backbone from 224×224 images in an N-way K-shot episode.

C Additional Experimental Results

C.1 Effect of Augmentation-Aware Module

We conduct experiments with the augmentation-aware module φ. The experimental results in Tab. 9 demonstrate the effectiveness of φ, which benefits the representation robustness via reasonably predicting the similarity between views caused by different augmentations.

C.2 Effect of Minimum Similarity Strategy

We conduct experiments with different strategies of similarity selection in variable similarity. The experimental results in Tab. 10 demonstrate the effectiveness of the minimum similarity strategy. These results support our hypothesis that focusing on the most challenging cases enhances representation robustness.

C.3 More Epochs

In the aforementioned experiments, we adopt the setup of 200 training epochs from prior work [41]. To comprehensively evaluate model performance at convergence, we conduct experiments with more training epochs. Specifically, we pretrain both CLVS and comparative methods on the ImageNet-100 dataset for 800 epochs and perform linear evaluation. As shown in Tab. 11, our method consistently

Table 9: Effect of the augmentation-aware module in CLVS to the linear evaluation accuracy (%). All methods are pretrained for 200 epochs on the ImageNet-100 dataset.

Method	Baseline	w/o φ	w φ
MoCo+CLVS	78.8	83.1	**84.5**
SimSiam+CLVS	79.1	84.5	**86.0**

Table 10: Top-1 accuracies of linear evaluation (%) with different strategies. All methods with a ResNet-50 encoder are pretrained for 200 epochs on the ImageNet-100 dataset.

Method	Min	Max	Mean
MoCo+CLVS	**84.5**	84.0	83.8
SimSiam+CLVS	**86.0**	85.4	85.7

outperforms comparative methods. These extended results further corroborate and align with the findings presented in Table 3, reinforcing the robustness and superiority of our proposed method.

Table 11: Top-1 accuracies of linear evaluation (%). All methods with a ResNet-50 encoder are pretrained for 800 epochs on the ImageNet-100 dataset.

Method	ImageNet-100	CIFAR10	CIFAR100	Caltech101	SUN397	Food	Flowers	Pets
MoCo [16]	85.7	87.5	67.0	89.1	51.3	62.8	84.9	76.6
SimSiam [12]	81.6	88.1	68.7	89.7	52.1	64.6	88.0	75.3
AugSelf [35]	83.3	89.7	71.7	91.4	53.2	68.3	90.5	77.0
RényiCL [40]	86.6	90.0	69.8	90.5	55.4	65.6	89.9	77.9
EquiMod [38]	86.2	88.6	70.0	89.6	54.0	66.7	89.3	**79.7**
GCA [44]	74.0	75.2	49.7	78.0	37.3	43.6	70.6	52.2
INTL [32]	86.5	88.3	69.5	89.9	53.6	64.5	89.0	79.0
MoCo+CLVS	**88.7**	88.4	68.3	90.6	54.9	66.7	88.9	78.7
SimSiam+CLVS	87.6	**91.0**	**73.4**	**92.1**	**56.6**	**69.6**	**91.3**	79.2

C.4 ViT Backbone

To further validate the effectiveness of our proposed method, we conduct experiments using the ViT [63] backbone. Recently, MoCo-v3 [64] explores training ViT backbone in self-supervised learning framework. Therefore, we implement our method into MoCo-v3. We pretrain the ViT-Small encoder on the Imagenet-100 dataset for 200 epochs with the batch size of 1024. We follow the experimental settings in MoCo-v3, including AdamW optimizer [65] with a linear learning rate warm-up for the first 40 epochs, a momentum of 0.9, and a weight decay of 0.1. A cosine learning rate schedule is applied to the encoder and predictor. The learning rate is set to 1.5e-4 and temperature is 0.2. The experimental results are shown in Tab. 12. The proposed method with the ViT backbone exhibits better performance across all datasets. This reveals the effectiveness of the variable similarity, which is applicable of the ViT backbone.

Table 12: Top-1 accuracies of linear evaluation (%). All methods with a ViT-Small encoder are pretrained for 200 epochs on the ImageNet-100 dataset.

Method	ImageNet-100	CIFAR10	CIFAR100	Caltech101	SUN397	Food	Flowers	Pets
MoCo-v3 [16]	78.4	86.2	66.4	82.2	47.1	60.5	83.7	66.9
MoCo-v3 + CLVS	**80.7**	**86.9**	**68.5**	**85.3**	**49.5**	**63.9**	**86.8**	**69.7**

C.5 Impact of Individual Data Augmentations

To better understand how different data augmentations contribute to variable similarity estimation, we perform an ablation where each augmentation type is applied in isolation within the SimSiam framework. Specifically, we consider Random Cropping, Color Jittering, Gaussian Blurring, and Horizontal Flipping, and evaluate the similarity prediction using only the parameter difference from the chosen augmentation. As shown in Table 13, cropping and color jittering yield the strongest performance, indicating that view differences along spatial and color dimensions provide more informative cues for similarity estimation. In contrast, blurring and flipping lead to slightly lower accuracy, likely because they have limited impact on semantic content: flipping only changes orientation and blurring mainly suppresses low-level details. This suggests that augmentations that alter semantically relevant aspects of the image are more effective for guiding similarity estimation.

Table 13: Linear evaluation accuracy (%) with individual augmentations. The method uses a ResNet-50 encoder pretrained for 200 epochs on ImageNet-100, with only one augmentation type applied during similarity estimation.

Method	Random Cropping	Color Jittering	Gaussian Blurring	Horizontal Flipping
SimSiam + CLVS	85.1	84.9	84.4	84.2

C.6 Parametric Sensitivity

We systematically analyze the sensitivity of the loss hyperparameters ω, λ and γ by varying each in $0.1, 0.5, 1$. As shown in Table 14, ω exhibits the clearest trend, i.e., intermediate values consistently lead to the best results, while both small and large values reduce accuracy. The effect of λ is similar, where performance peaks at 0.5 and slightly declines at 1. In contrast, γ shows relatively stable behavior across different values, with a marginal gain at $\gamma = 1$, indicating lower sensitivity. Overall, these findings suggest that CLVS is robust to hyperparameter variations within a reasonable range, with optimal performance typically achieved at moderate settings.

Table 14: Parametric sensitivity of different hyperparameters in the training loss. Linear evaluation accuracy (%) of ResNet-50 backbone encoder pretrained for 200 epochs on the ImageNet-100 dataset.

Parameter	Value	MoCo+CLVS	SimSiam+CLVS
ω	0.1	84.1	85.4
	0.5	**84.9**	**86.0**
	1	83.9	85.1
λ	0.1	84.4	85.8
	0.5	**84.9**	**86.0**
	1	84.5	85.2
γ	0.1	84.1	85.8
	0.5	84.7	85.8
	1	**84.9**	**86.0**

C.7 Training Time Comparison

We provide experiments to record the training time of our method and the compared methods. Specifically, we use 4 NVIDIA A100-SXM4-40GB GPUs to train these methods, where the batch size is set to 256. As shown in Tab. 15, although our method introduces an extra module φ, it only increases the time by around 5% compared to the baseline. This demonstrates competitive efficiency in terms of training time.

Table 15: Time comparison of our proposed method and compared methods on the ImageNet-1k dataset (in minutes).

Method	Time / 1 epoch
SimSiam [12]	24.22
RényiCL [40]	24.72
EquiMod [38]	38.83
INTL [32]	29.17
CLVS	25.78

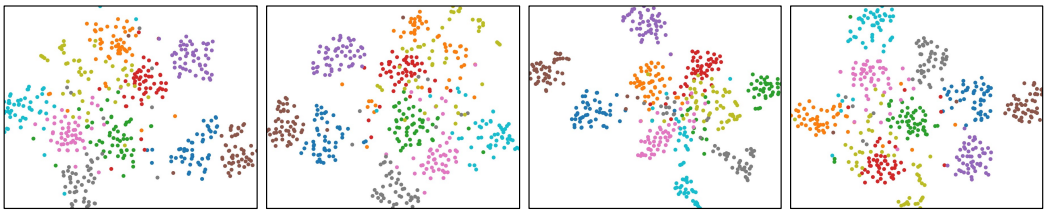
Table 16: NMI score of our methods and compared method to quantify the class separation on the ImageNet-100 dataset.

Method	NMI
SimSiam	0.69
Augself	0.74
RényiCL	0.68
CLVS	**0.76**

C.8 Visualization of Learned Representations

We show the t-SNE [66] visualizations of the representations learned by our proposed method and several methods on the ImageNet-100 dataset. As shown in Fig. 6, our proposed method leads to better class separation. This indicates that variable similarity facilitates the model to learn more

discriminative feature representations. Furthermore, we also provide the NMI (Normalized Mutual Information) score to quantify the class separation in Tab. 16. Our method achieves the highest NMI score, aligning with the trends observed in t-SNE visualizations.



(a) SimSiam (b) AugSelf (c) RényiCL (d) CLVS

Figure 6: t-SNE visualizations of image representations from randomly selected 10 classes in the ImageNet-100 validation set.