A Unified Mixture-View Framework for Unsupervised Representation Learning

Xiangxiang Chu[1]
chuxiangxiang@meituan.com
Xiaohang Zhan[2]
xiaohangzhan@outlook.com
Bo Zhang[1]
zhangbo97@meituan.com

[1] Meituan
Beijing, China
[2] The Chinese University of Hong Kong
HongKong, China

Abstract

Recent unsupervised contrastive representation learning follows a Single Instance Multi-view (SIM) paradigm where positive pairs are usually constructed with intra-image data augmentation. In this paper, we propose an effective approach called Beyond Single Instance Multi-view (BSIM). Specifically, we impose more accurate instance discrimination capability by measuring the joint similarity between two randomly sampled instances and their mixture, namely spurious-positive pairs. We believe that learning joint similarity helps to improve the performance when encoded features are distributed more evenly in the latent space. We apply it as an orthogonal improvement for unsupervised contrastive representation learning, including current outstanding methods SimCLR [6], MoCo [19], BYOL [18] and SimSiam [7]. We evaluate our learned representations on many downstream benchmarks like linear classification on ImageNet-1k and PASCAL VOC 2007, object detection on MS COCO 2017 and VOC, etc. We obtain substantial gains with a large margin almost on all these tasks compared with prior arts.

1 Introduction

Unsupervised representational learning is now on the very rim to take over supervised representation learning. It is supposed to be a perfect solver for real-world scenarios full of unlabeled data. Among them, self-supervised learning has drawn the most attention for its good data efficiency and generalizability.

Self-supervised learning typically involves a proxy task to learn discriminative representations from self-derived labels. Among all manners of these proxy tasks [12, 16, 22, 25, 27], instance discrimination [23, 36], known as contrastive representation learning, has emerged as the most effective paradigm. Its subsequent methods [6, 18, 19, 30, 43] have greatly reduced the gap between unsupervised and supervised learning. Specifically, instance discrimination features a Single Instance Multi-view (SIM) paradigm to separate different instances. It seeks to narrow the distance among multiple views of the same instance (e.g. an image), which are typically yielded from vanilla data augmentation policies like color jittering, cropping, resizing, applying Gaussian noise. Consequently, the invariance of the

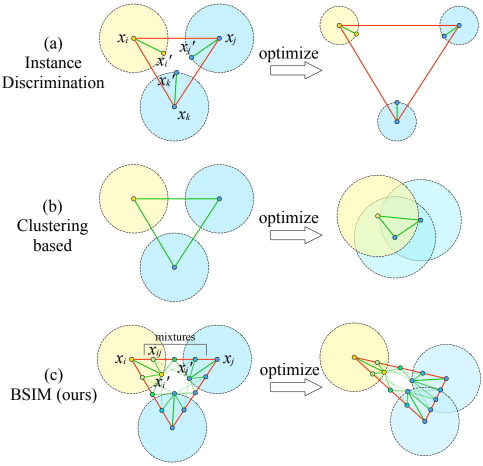


Figure 1: A schematic view of three self-supervised paradigms. Note x_i, x_j and x_k are different instances. Green lines link the positive pairs while red for negative. Circles show decision boundaries (same color for the same class). Instance discrimination narrows the boundary and pushes away all instances. Clustering-based methods might cluster wrong instances (*e.g.* yellow and blue) due to a shortcut defect. BSIM mixes instances (where hue indicates the ratio) to construct spurious positive pairs (*e.g.* (x_{ij}, x_i') and (x_{ij}, x_j'). BSIM encourages contrastive competition among instances, thus being better at learning interclass and intraclass representation.

network is easily bounded by these limited augmentations. Since different instances are continuously driven apart, SIM prevents itself from characterizing the relations among different instances from the same class, as opposed to supervised classification.

Meanwhile, clustering-based self-supervised methods [3, 39] alternate feature clustering with learning to capture similarities among different instances. These methods avoid the intrinsic weakness of instance discrimination, but suffer from a so-called 'shortcut' problem, i.e., when two instances are occasionally grouped into a cluster, their similarity will be further enhanced. As a result, the training easily drifts into trivial solutions, *e.g.*, merely grouping images in similar color or texture.

In view that instance discrimination pushes apart different instances indistinguishably as shown in Fig. 1(a), and clustering-based methods are easily trapped in shortcut issues as shown in Fig. 1(b), we are motivated to explore a new paradigm to distinguish both intraclass and interclass instances. In this work, we propose BSIM to learn better representations that capture high-level inter-image relations, which also potentially avoid the above-mentioned shortcut issue. To make the minimal modification from previous works, BSIM shares similar pipelines to SimCLR [6], MoCo [19], BYOL [18] and SimSiam [7], while focusing on a new way to construct positive pairs.

Specifically, as shown in Fig. 1(c), BSIM first creates mixtures using CutMix [38] among instances by proportion, *e.g.*, x_{ij} that mixes λ of x_i and $(1-\lambda)$ of x_j, where $\lambda \in (0,1)$ obeys a Beta distribution. Different from instance discrimination that constructs positive pairs between two views from the same image, BSIM makes use of mixed views x_{ij} to create what we call *spurious-positive* pairs (x_{ij}, x_i') and $\left(x_{ij}, x_j'\right)$. The optimization also proportionally takes λ into account for computing the losses. The interaction of spurious-positive pairs compete for an equilibrium state when grouping intra-instance and inter-instance views, modulated by the distribution of λ[1]. Meantime, negative pairs keep pushing away different instances. As a result, BSIM encourages contrastive competition among instances, leaning towards exploring higher-level inter-image relations. Since BSIM does not maintain dynamically changing pseudo labels as clustering-based methods, the shortcut issue is naturally avoided. The contribution of this paper is twofold,

- We propose a novel paradigm, namely BSIM, to encourage contrastive competition among instances for higher-level representation learning. Specifically, we generate *spurious positive examples* using CutMix mixture, and we quantitatively score the distance between any image pairs by formulating a new contrastive loss .

- BSIM is a general-purpose enhancement to existing methods that rely on instance discrimination (e.g. SimCLR, MoCo, BYOL, SimSiam). BSIM boosts performance for prior arts by clear margins and the gain from BSIM (e.g. BYOL-BSIM) is even comparable to the latest elaborately designed methods such as SimSiam [7]. Moreover, it requires minimal modification to current self-supervised learning frameworks while adding neglectable cost. In general, BSIM-powered networks achieve state-of-the-art performance in a large body of standard benchmarks.

2 Related Work

Self-supervised learning based on contrastive loss. Early methods focus on devising proxy tasks to either reconstruct the image after transformations [22, 27, 42], or predict the configurations of applied transformation on a single image [11, 13, 16, 25]. Till recently contrastive loss based approaches [2, 6, 18, 19] emerge as the mainstream paradigm, which features two components: the selection of positive or negative examples and the contrastive loss design. This routine leverages different augmented views of an image to construct positive pairs, while deeming other images as negative samples.

Particularly, SimCLR [6] produces positive and negative pairs within a mini-batch of training data and chooses InfoNCE [26] loss to train the feature extraction backbone. It requires a large batch-size to effectively balance the positive and negative ones. MoCo [19] makes use of a feature queue to store negative samples, which greatly reduces high memory cost in [6]. Moreover, it proposes a momentum network to boost the consistency of features. BYOL [18] challenges the indispensability of negative examples and achieves impressive performance by only using positive ones. A mean square error loss is applied to make sure that positive pairs can predict each other. SimSiam [7] utilizes stop-gradient as an alternative method to avoid mode collapse, simplifying the design compared to prior arts.

Besides, carefully designed augmentations to build positive pairs are proven to be critical for good performance [1, 6, 8, 17, 24, 29, 30, 34], because appropriate augmentations modulate the distribution of positive examples in the feature space. SwAV [2] obtains the state-of-the-art unsupervised performance by using a mixture of views in different resolutions in place of two full-resolution ones. In the meantime, some researches study the role of hard negative examples [9, 20, 21, 35, 37]. However, all the above approaches try to push each image instance away from each other by regarding them as its negative samples. Is it possible to model the vicinity relation by measuring that distance quantitatively? To our best knowledge, this problem is rarely studied in the field of self-supervised learning and BSIM is aimed to bridge this gap.

[1] It is worth noting that $\lambda \sim Bernoulli(0.5)$ degenerates the problem into instance discrimination exactly.

Mixture as a regularization technique in supervised learning. Mixture of training samples like Mixup [41], CutMix [38], and Manifold Mixup [33] has been proved to be a strong regularization for supervised learning, based on the principle of vicinal risk minimization [5]. It is designed to model vicinity relation across different classes other than vanilla data augmentation tricks that only considers the same class. CutMix [38] debates that Mixup introduces unnatural artifacts by mixing the whole image region while Cutout [10] might pay attention to less discriminative parts. CutMix claims to effectively localize the two classes instead. [31] argues that mixing images is akin to mixing sounds [32] for CNNs, although not easily perceptible for humans. Other than deeming it as vanilla data augmentation that adds data variation, they consider it as an enlargement of Fisher's criterion [15], *i.e.*, the ratio of the between-class scatter to the within-class scatter, and it regularizes positional relationship among latent feature distributions. Furthermore, [28] notices that label smoothing during mixup training has a calibration effect which regularizes over-confident predictions.

3 Methodology

Different from the wide use of augmentation as a useful regularization in supervised learning, how to use it in unsupervised learning remains to be an open problem. Using the mixture solely as one of data augmentation techniques to produce positive pairs *significantly* weakens the performance of MoCoV2 [8] (Table 1). This preferred regularization in supervised learning seems inherently incompatible with contrastive learning: drawing near multiple augmented views of an image while pushing away from the others, which we call Single Instance Multi-view (SIM) for simplicity. Instead, a mixture view shall be drawn close to both source images. This motivates us to scheme an alternative strategy for contrastive learning. There are two basic and coupled problems to be answered: how to address the degradation and how to design feasible mixtures.

Method	Epoch	SVM @VOC2007	LC @ImageNet
MoCoV2 (2020)	200	83.81%	67.5%
MoCoV2+MixAug	200	80.10% (-3.71%↓)	64.6% (-2.9%↓)

Table 1: Regarding mixture as an extra data augmentation (MoCoV2+MixAug) weakens its performance severely. LC: linear classification on ImageNet.

The central principle of contrastive learning is to encode semantically similar views (positive pairs) into latent representations that are close to each other while driving dissimilar ones (negative pairs) apart. A major question is how to effectively *synthesize positive and negative pairs* given a dataset of i.i.d. samples as raised by [30]. Another question engages the *design of contrastive loss*. Next, we discuss BSIM in details to address both.

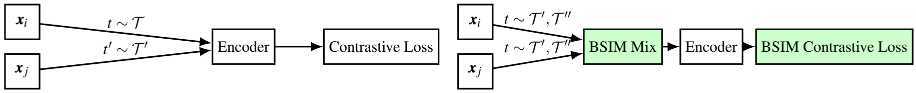
3.1 Spurious-Positive Views From Multiple Images

Given a set of images $\mathcal{D}$, two images x_1, x_2 sampled uniformly from $\mathcal{D}$, and two image augmentation distributions of $\mathcal{T}'$ and $\mathcal{T}''$ (whether $\mathcal{T}'$ and $\mathcal{T}''$ are the same depends on different methods), we first generate a new example $x'_{1,2}$ by mixing $t'(x_1)$ and $t'(x_2)$, where $t' \sim \mathcal{T}'$. Specifically, $x'_{1,2}$ borrows λ region from $t'(x_1)$ and remaining $(1-\lambda)$ part from $t'(x_2)$. Thus, $x'_{1,2}$ has two spurious-positive examples, i.e., $t''(x_1)$ and $t''(x_2)$, where $t'' \sim \mathcal{T}''$. These images are encoded by a neural network f to extract high-level features, followed by a projection head g that maps the representation to a space ready to apply contrastive loss. The projection function is often implemented as a simple MLP network. For a better understanding, we give a schematic construction under SimCLR framework in Fig. 6 (supp.).

The distance between the newly mixed example and its parents is controlled by λ. BSIM uses a popular and handy option to generation λ from Beta distributions, i.e., $\lambda \sim \beta(\alpha, \alpha)$, where α is a hyper-parameter. It is evident that it would degrade to the single-instance multi-view case if λ is always 0 or 1 when $\alpha \to 0$. That being said, BSIM is a generalization of SIM so that previous SIM methods reside within our larger framework.

3.2 Loss Functions of BSIM

It's intuitive to change loss functions since spurious-positive examples are introduced. For example, it's unreasonable to assign a new instance generated by half mixing two images (a dog and a cat, $\lambda = 0.5$) to a dog or cat. Human would easily tell this image is half cat and half dog. This means its projected feature in the high-dimensional latent space should be nearby a dog, as well as a cat, but much far from an orangutan. This motivates us to design a particular loss for BSIM on top of the spurious-positive views, shown in Fig. 2. Specifically, we adapt our method to four popular frameworks SimCLR [6], MoCo [19], BYOL [18] and SimSiam [7]. In order to make our paper more readable, we roughly follow the same naming conventions as these papers and list the symbol notations in Table 1 (supp.).



(a) (b)

Figure 2: Our generic BSIM framework (b) serves as a plug-and-play adds-on for current contrastive learning paradigm (a). Note $\mathcal{T}$ and $\mathcal{T}'$ are augmentation policy distributions.

SimCLR-BSIM. SimCLR uses a single augmentation distribution, i.e. $\mathcal{T}'$ and $\mathcal{T}''$ are identical herein. The encoder network f encodes $x'_{1,2}$ as $f(x'_{1,2})$. Note $x'_{1,2}$ should show similarities with x''_1 as well as x''_2, which is measured by the sim function in the projected z space. We follow the definition in [6] for the similarity function as $\mathrm{sim}(\boldsymbol{z}_i, \boldsymbol{z}_j) = \boldsymbol{z}_i^\top \boldsymbol{z}_j / (\|\boldsymbol{z}_i\| \|\boldsymbol{z}_j\|)$. We use λ to regularize these similarities and the matching loss can be formulated as,

$$\begin{aligned} \ell'_i(\lambda) = & -\lambda \log \frac{e^{\mathrm{sim}(\boldsymbol{z}'_{i,j}, \boldsymbol{z}''_i)/\tau}}{\sum_{k=1}^{N} [e^{\mathrm{sim}(\boldsymbol{z}'_{i,j}, \boldsymbol{z}''_k)/\tau} + \mathbb{1} \cdot e^{\mathrm{sim}(\boldsymbol{z}'_{i,j}, \boldsymbol{z}'_{i,k})/\tau}]} \\ & -(1-\lambda) \log \frac{e^{\mathrm{sim}(\boldsymbol{z}'_{i,j}, \boldsymbol{z}''_j)/\tau}}{\sum_{k=1}^{N} [e^{\mathrm{sim}(\boldsymbol{z}'_{i,j}, \boldsymbol{z}''_k)/\tau} + \mathbb{1} \cdot e^{\mathrm{sim}(\boldsymbol{z}'_{i,j}, \boldsymbol{z}'_{i,k})/\tau}]}. \\ & \text{where} \quad \mathbb{1} = \begin{cases} 1 & k \notin \{i, j\} \\ 0 & \text{otherwise} \end{cases} \end{aligned} \tag{1}$$

Similarly, we can formulate ℓ''_i if we use $x''_{1,2}$ as the anchor. Hence, the NT-Xent [6] loss is defined by the summation of each individual loss within the mini-batch data of size N as,

$$L_{\texttt{NT-Xent}}(\lambda) = \frac{1}{2N} \sum_{k=1}^{N} \ell'_i(\lambda) + \ell''_i(\lambda), \;\; \lambda \sim \beta(\alpha, \alpha). \tag{2}$$

SimCLR [6] has $2N$ positive pairs and $2N(N-1)$ negative ones in total at each iteration. Whereas, our method includes $4N$ spurious-positive pairs, i.e., $(x'_{i,j}, x''_i)$, $(x'_{i,j}, x''_j)$, $(x''_{i,j}, x'_i)$, $(x''_{i,j}, x'_j)$, and $2N(N-2)$ negative ones. The proposed method is depicted in Figure 6 (supp.).

MoCo-BSIM. We produce the query q of MoCo by forwarding the mixed image controlled by λ. We illustrate the procedure in Fig. 7 (supp.).

$$\mathcal{L}_q = -\lambda \log \frac{\exp(q \cdot k_+^{\lambda}/\tau)}{\sum_{i=1}^{N} \exp(q \cdot k_i/\tau)} - (1-\lambda) \log \frac{\exp(q \cdot k_+^{1-\lambda}/\tau)}{\sum_{i=1}^{N} \exp(q \cdot k_i/\tau)} \tag{3}$$

where k_+^{λ} and $k_+^{1-\lambda}$ represent the corresponding key of images that produced the mixture respectively, and k_i are the keys in the current queue. τ is the softmax temperature.

BYOL-BSIM. BYOL-BSIM generates two augmented views $x'_1 \triangleq t'(x_1)$ and $x''_1 \triangleq t''(x_1)$ from x_1 by applying respectively image augmentations $t' \sim \mathcal{T}'$ and $t'' \sim \mathcal{T}''$. Following the same procedure, we produce x'_2 and x''_2. Then we produce a new image $x'_{1,2}$ by λ-based mixture x'_1 and x'_2 through cutmix. The online network outputs $y'_\theta \triangleq f_\theta(x'_{1,2})$ and the projection $z'_\theta \triangleq g_\theta(y')$. The target network yields two ℓ_2-normalized projections $\bar{z}''_1$, $\bar{z}''_2$ from x''_1 and x''_2.

We sum up the MSE loss between the projection of the mixed image and its parents by the mixture coefficient λ. This process is shown in Fig. 8 (supp.). Formally, the loss is:

$$\mathcal{L}'_{\theta,\xi} = -2[\lambda \frac{\langle q'_\theta(z'_\theta), z''_{i,\xi} \rangle}{\|q'_\theta(z'_\theta)\|_2 \cdot \|z''_{i,\xi}\|_2} + (1-\lambda) \frac{\langle q'_\theta(z'_\theta), z''_{j,\xi} \rangle}{\|q'_\theta(z'_\theta)\|_2 \cdot \|z''_{j,\xi}\|_2}] \tag{4}$$

Note $z''_{i,\xi}$ and $z''_{j,\xi}$ mean the projection of the representation of x''_i and x''_j generated by the target network. We obtain $\mathcal{L}''_{\theta,\xi}$ by using x''_1 and x''_2 as the input of online network. Note that BYOL doesn't rely on negative samples. The normalized projection is on the sphere of a unit ball in the high dimensional space, see Fig. 5 (supp.).

SimSiam-BSIM. Since SimSiam utilizes the same loss as BYOL, we use exactly the same loss form as Eq 4 with scale 0.5 to match the loss in SimSiam [7].

WBSIM. Alternatively, we offer BSIM as a general adds-on by adding a weighted BSIM (WBSIM) loss to the usual SIM losses, see Sec 2 (supp.) for details.

3.3 Mixture Strategy Design

CutMix and Mixup [41] are two popular strategies of generating mixtures at image level. Whereas we don't utilize Mixup because it is less natural, even humans cannot easily tell the mixture coefficient λ simply by checking the mixed image. We compare Mixup with CutMix via carefully controlled experiments (both with BSIM loss and the same hyper-parameter settings) under the framework of MoCoV2. Results from Table 2 disapprove of the use of Mixup in producing spurious-positive examples. The observation differs from supervised learning, where both boost the performance.

We also compare their performance using VOC object detection under the same metrics in Sec 3.3 (supp.). The result is shown in Table 3. Mixup fails to improve the performance of its baseline without mixtures. In contrast, CutMix can improve the detection performance. Therefore, we utilize CutMix as our default mixture strategy.

Method	SVM	SVM Low-Shot (96)
MoCoV2 [19]	83.81%	82.01±0.13%
MoCoV2 (w/ Mixup, α=1.0)	82.50%	80.54±0.26%
MoCoV2 (w/ Mixup, α=0.5)	82.80%	80.58±0.31%
MoCoV2-BSIM (w/ CutMix, α=1.0)	**84.55%**	**82.65±0.34%**

Table 2: SVM evaluations on PASCAL VOC2007. Mixup deteriorates the performance of the baseline.

Method	AP_{50}	AP_{75}	AP
No Mix	82.4%	63.6%	57.0%
Mixup (α=1.0)	82.2%(-0.2↓)	63.4%(-0.2↓)	56.9%(-0.1↓)
CutMix (α=1.0)	**82.7%**(+0.3↑)	**64.0%**(+0.4↑)	**57.3%**(+0.3↑)

Table 3: Object detection results under the MoCoV2 framework on PASCAL VOC `trainval07+12`.

4 Experiments

Setup. We generally follow the compared methods. Details and cost are in Sec 3 (supp.). Object detection and instance segmentation experiments are given in Sec 3.3 (supp.).

4.1 Evaluation on Linear Classification

Linear SVM classification on VOC2007. The results are shown in Table 4. In most cases, BSIM consistently boosts the baselines by about 1% mAP. Particularly, BYOL-BSIM is boosted by a clear margin: **1.4%** mAP. BYOL-BSIM300 outperforms the supervised pre-trained baseline with **0.4%** higher mAP. BYOL-BSIM (200 epochs' training) is comparable to BYOL300 (300 epochs). Noticeably, WBSIM further boosts the performance. MoCoV2 benefits **1%** mAP from BSIM, and an extra **0.6%** higher mAP from WBSIM, indicating that BSIM is complementary to SIM.

Method	SVM	SVM Low-Shot (%mAP)							
	%mAP	1	2	4	8	16	32	64	96
Supervised	87.2	53.0	63.6	73.7	78.8	81.8	83.8	85.2	86.0
SimCLR (2020)	79.0	32.5	40.8	50.4	59.1	65.5	70.1	73.6	75.4
SimCLR-BSIM	80.0	33.9	44.7	50.9	60.5	67.8	72.0	75.4	77.2
MoCo (2020)	79.2	30.0	37.7	47.6	58.8	66.0	70.6	74.6	76.1
MoCoV2 (2020)	83.8	43.7	55.2	63.2	71.5	75.4	79.1	81.2	82.0
MoCoV2-BSIM	84.8	**50.0**	53.9	**65.3**	72.4	76.3	79.3	81.7	82.8
MoCoV2-WBSIM	85.4	46.5	56.9	64.6	**74.7**	78.2	80.6	82.8	83.7
BYOL (2020)	85.1	44.5	52.1	62.9	70.9	76.2	79.5	81.9	83.1
BYOL-BSIM	**86.5**	42.6	**55.9**	64.6	72.7	**78.8**	**81.9**	**83.6**	**84.6**
BYOL300 (2020)	86.6	42.5	56.1	64.7	73.0	77.7	82.2	83.7	84.7
BYOL-BSIM300	87.6	**45.7**	54.5	66.4	75.0	79.8	83.2	**85.2**	86.0
BYOL-WBSIM300	**87.7**	44.1	**60.7**	**68.1**	**76.0**	**81.0**	**83.6**	**85.2**	**86.3**
SwAV (2020)*	85.4	-	-	-	-	-	-	-	-

Table 4: ResNet-50 linear SVMs mAP on VOC07 [14] classification using two 224× 224 views. BYOL variants with "300" are trained for 300 epochs as [18]. *: SwAV is trained for 400 epochs.

Low-shot classification on VOC2007. The results are shown in Table 4. BSIM helps all baselines to achieve better performance by substantial margins. It's interesting to see that BYOL-BSIM300 gradually bridge its gap from the supervised baseline. When the number of the training set is more than 64, it's comparable to the supervised version.

Linear classification on ImageNet. The results are shown in Table 5, where the performance of the competing methods are extracted from [40].

Method	Epoch	Backbone	Top-1 Accuracy
InfoMin Aug (2020)	200	R50	70.1
MoCo (2020)	200	R50	61.0
SimCLR(2020)	200	R50	61.6
SimCLR-BSIM	200	R50	62.3 (+0.7↑)
MoCoV2 (2020)	200	R50	67.5
MoCoV2-BSIM	200	R50	68.0 (+0.5↑)
MoCoV2-WBSIM	200	R50	68.4 (+0.9↑)
BYOL (2020)	200	R50	69.1
BYOL-BSIM	200	R50	69.8 (+0.7↑)
BYOL (2020)†	300	R50	72.3
BYOL-BSIM	300	R50	72.7 (+0.4↑)
BYOL-WBSIM	300	R50	**73.0 (+0.7↑)**
SimSiam (2021)	200	R50	70.0
SimSiam-BSIM (2021)	200	R50	70.4(+0.4↑)
SimSiam-WBSIM (2021)	200	R50	70.8(+0.8↑)
SwAV (2020)	200	R50	69.1
SwAV (2020)	400	R50	70.7

Table 5: Linear classification on ImageNet (top-1 center-crop accuracy on the validation set). All models are trained with two 224×224 views. †: reproduced. SwAV result is from SimSiam [7].

4.2 Evaluation on Semi-supervised Classification

Results are shown in Table 6. BSIM improves the baselines by significant margins, especially when the amount of available labels is small. MoCoV2-BSIM obtains 44.3% top-1 accuracy, which is 5.2% higher than MoCoV2. Although BYOL-BSIM is only trained for 200 epochs, it achieves comparable results as BYOL1000. WBSIM300 can further boost the performance to the state-of-the-art 57.2%. Specifically, it obtains 57.2% top-1 accuracy using 1% labeled data, which is about 4% higher than BYOL1000. When we collect more data (10%), BYOL-WBSIM outperforms BYOL1000 with about 2%. Combining SIM and BSIM seems to learn better representations.

Method	Epoch	1% labelled data		10% labelled data	
		top-1(%)	top-5(%)	top-1(%)	top-5(%)
Supervised	-	68.7	88.9	74.5	92.2
SimCLR (2020)	200	36.1	64.5	58.5	82.6
SimCLR-BSIM	200	$38.2_{2.1\uparrow}$	$67.5_{3.0\uparrow}$	$61.2_{2.7\uparrow}$	$84.5_{2.9\uparrow}$
MoCo (2020)	200	33.2	61.3	60.1	84.0
MoCoV2 (2020)	200	39.1	68.3	61.8	85.1
MoCoV2-BSIM	200	$40.8_{1.7\uparrow}$	$70.3_{2.0\uparrow}$	$62.6_{0.8\uparrow}$	$85.8_{0.7\uparrow}$
MoCoV2-WBSIM	200	$44.3_{5.2\uparrow}$	$72.9_{4.6\uparrow}$	$63.9_{2.1\uparrow}$	$86.6_{1.5\uparrow}$
BYOL (2020)	200	49.4	76.8	65.9	87.8
BYOL-BSIM	200	$\mathbf{53.0}_{3.6\uparrow}$	$\mathbf{79.9}_{3.1\uparrow}$	$\mathbf{68.2}_{2.3\uparrow}$	$\mathbf{89.0}_{2.2\uparrow}$
SwAV (2020)	800	53.9	78.5	70.2	89.9
SimCLR (2020)	1000	48.3	75.5	65.6	87.8
BYOL (2020)	1000	53.2	78.4	68.8	89.0
BYOL-WBSIM	300	**57.2**	**81.8**	**70.7**	**90.5**

Table 6: Semi-supervised classification on ImageNet. We report center-crop accuracy on the val set.

5 Ablation and Discussions

Sensitivity on α. We further analyze the performance sensitivity of α. Regarding the intensive resource cost, we report the SVM and low-shot SVM results in Table 7 using MoCo-V2. We keep the same pre-training setting. The distribution from group $\alpha = 0.75$ performs best. The performance keeps stable when $\alpha > 0.5$. However, it drops severely once $\alpha \to 0$ when it degenerates to MoCo-V2.

Method	α	SVM	SVM Low-Shot (96)
MoCoV2-BSIM	1.0	84.55	82.65±0.34
MoCoV2-BSIM	0.75	**84.56**	**82.67±0.26**
MoCoV2-BSIM	0.5	84.23	82.50±0.29
MoCoV2-BSIM	0.25	84.02	82.18±0.31

Table 7: Performance sensitivity on α using MoCoV2-BSIM. The classification result is averaged across 5 independent experiments. When $\alpha < 0.01$, MoCoV2-BSIM can be regarded as MoCoV2 approximately which achieves 83.8% mAP on SVM.

Why does mixture as data augmentation fail? As mentioned in Table 2, simply adopting mixture methods as a data augmentation option severely degrades the performance. Regarding the mixed image as the same instance as the original one forces the network to expand the decision boundaries blindly. Consequently, the network might be trapped in shortcut solutions to group images in different classes indiscriminately.

Why BSIM improves discrimination? A mixture is close to the decision boundaries in instance discrimination task where neural networks are normally less certain about. In instance discrimination, the decision boundaries keep being separated, leaving some area crowded while others sparse, which is unfavorable for learning high-level inter-image relations. In BSIM, when learning λ-balanced similarities between competing spurious-positive pairs, the network encourages contrastive competition among instances to occupy the area near decision boundaries (Fig. 3 supp.). We sample three times to demonstrate that BSIM in general gives cleaner inter-class representation, while SIM has a more intertwined one. This proves that learning the distance to the mixture helps improve classification by generating a better latent representation. In this way, we hypothesize that the network has to encode the latent representations more accurately like a ruler. As a result, the features have to scatter more evenly, especially for between-class areas that are harder to predict, which is shown in Fig. 2 (supp.). We hypothesize this is a major factor that BSIM offers stronger discrimination capability. In Sec 4 (supp.), we give a comparison with other mixture-based approaches and draw a latent representation via TSNE to manifest the working mechanism of BSIM.

6 Conclusion

In this paper, we propose BSIM, a novel self-supervised representation learning approach beyond the current instance discrimination paradigm. It makes minimal modification to the existing instance discrimination methods such as SimCLR, MoCo, BOYL, and SimSiam, while significantly improving the performance on many downstream tasks. We justify the superiority of BSIM via analyzing the optimization behaviors when combined with different paradigms, which provides a new perspective in the field of contrastive representation learning. Being a simple and lightweight plugin, it substantially enhances the SSL performance.

References

[1] YM Asano, C Rupprecht, and A Vedaldi. A critical analysis of self-supervision, or what we can learn from a single image. In *International Conference on Learning Representations*, 2019.

[2] M. Caron, I. Misra, J. Mairal, Priya Goyal, P. Bojanowski, and Armand Joulin. Unsupervised learning of visual features by contrasting cluster assignments. In *Advances in Neural Information Processing Systems*, 2020.

[3] Mathilde Caron, Piotr Bojanowski, Armand Joulin, and Matthijs Douze. Deep clustering for unsupervised learning of visual features. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 132–149, 2018.

[4] Mathilde Caron, Ishan Misra, Julien Mairal, Priya Goyal, Piotr Bojanowski, and Armand Joulin. Unsupervised learning of visual features by contrasting cluster assignments. *Advances in Neural Information Processing Systems*, 33, 2020.

[5] Olivier Chapelle, Jason Weston, Léon Bottou, and Vladimir Vapnik. Vicinal risk minimization. In *Advances in neural information processing systems*, pages 416–422, 2001.

[6] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. *arXiv preprint arXiv:2002.05709*, 2020.

[7] Xinlei Chen and Kaiming He. Exploring simple siamese representation learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2021.

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

[9] Ching-Yao Chuang, Joshua Robinson, Lin Yen-Chen, Antonio Torralba, and Stefanie Jegelka. Debiased contrastive learning. *arXiv preprint arXiv:2007.00224*, 2020.

[10] Terrance DeVries and Graham W Taylor. Improved regularization of convolutional neural networks with cutout. *arXiv preprint arXiv:1708.04552*, 2017.

[11] Carl Doersch, Abhinav Gupta, and Alexei A Efros. Unsupervised visual representation learning by context prediction. In *Proceedings of the IEEE international conference on computer vision*, pages 1422–1430, 2015.

[12] Jeff Donahue, Philipp Krähenbühl, and Trevor Darrell. Adversarial feature learning. In *International Conference on Learning Representations*, 2017.

[13] Alexey Dosovitskiy, Jost Tobias Springenberg, Martin Riedmiller, and Thomas Brox. Discriminative unsupervised feature learning with convolutional neural networks. In *Advances in neural information processing systems*, pages 766–774, 2014.

[14] 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(2):303–338, 2010.

[15] Ronald A Fisher. The use of multiple measurements in taxonomic problems. *Annals of eugenics*, 7(2):179–188, 1936.

[16] Spyros Gidaris, Praveer Singh, and Nikos Komodakis. Unsupervised representation learning by predicting image rotations. In *International Conference on Learning Representations*, 2018. URL https://openreview.net/forum?id=S1v4N2l0-.

[17] Raphael Gontijo-Lopes, Sylvia J Smullin, Ekin D Cubuk, and Ethan Dyer. Affinity and diversity: Quantifying mechanisms of data augmentation. *arXiv preprint arXiv:2002.08973*, 2020.

[18] Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre H Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Daniel Guo, Mohammad Gheshlaghi Azar, et al. Bootstrap your own latent: A new approach to self-supervised learning. *arXiv preprint arXiv:2006.07733*, 2020.

[19] 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*, pages 9729–9738, 2020.

[20] Ahmet Iscen, Giorgos Tolias, Yannis Avrithis, and Ondřej Chum. Mining on manifolds: Metric learning without labels. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

[21] Yannis Kalantidis, Mert Bulent Sariyildiz, Noe Pion, Philippe Weinzaepfel, and Diane Larlus. Hard negative mixing for contrastive learning. In *Neural Information Processing Systems (NeurIPS)*, 2020.

[22] Gustav Larsson, Michael Maire, and Gregory Shakhnarovich. Colorization as a proxy task for visual understanding. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6874–6883, 2017.

[23] Xiao Liu, Fanjin Zhang, Zhenyu Hou, Zhaoyu Wang, Li Mian, Jing Zhang, and Jie Tang. Self-supervised learning: Generative or contrastive. *arXiv preprint arXiv:2006.08218*, 2020.

[24] Ishan Misra and Laurens van der Maaten. Self-supervised learning of pretext-invariant representations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6707–6717, 2020.

[25] Mehdi Noroozi and Paolo Favaro. Unsupervised learning of visual representations by solving jigsaw puzzles. In *European Conference on Computer Vision*, pages 69–84. Springer, 2016.

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

[27] Deepak Pathak, Philipp Krahenbuhl, Jeff Donahue, Trevor Darrell, and Alexei A Efros. Context encoders: Feature learning by inpainting. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2536–2544, 2016.

[28] Sunil Thulasidasan, Gopinath Chennupati, Jeff A Bilmes, Tanmoy Bhattacharya, and Sarah Michalak. On mixup training: Improved calibration and predictive uncertainty for deep neural networks. In *Advances in Neural Information Processing Systems*, pages 13888–13899, 2019.

[29] Yonglong Tian, Dilip Krishnan, and Phillip Isola. Contrastive multiview coding. *arXiv preprint arXiv:1906.05849*, 2019.

[30] Yonglong Tian, Chen Sun, Ben Poole, Dilip Krishnan, Cordelia Schmid, and Phillip Isola. What makes for good views for contrastive learning. *arXiv preprint arXiv:2005.10243*, 2020.

[31] Yuji Tokozume, Yoshitaka Ushiku, and Tatsuya Harada. Between-class learning for image classification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5486–5494, 2018.

[32] Yuji Tokozume, Yoshitaka Ushiku, and Tatsuya Harada. Learning from between-class examples for deep sound recognition. In *International Conference on Learning Representations*, 2018.

[33] Vikas Verma, Alex Lamb, Christopher Beckham, Amir Najafi, Ioannis Mitliagkas, David Lopez-Paz, and Yoshua Bengio. Manifold mixup: Better representations by interpolating hidden states. In *International Conference on Machine Learning*, pages 6438–6447. PMLR, 2019.

[34] Tongzhou Wang and Phillip Isola. Understanding contrastive representation learning through alignment and uniformity on the hypersphere. In *International Conference on Machine Learning*, 2020.

[35] M. Wu, Chengxu Zhuang, M. Mosse, D. Yamins, and Noah D. Goodman. On mutual information in contrastive learning for visual representations. *ArXiv*, abs/2005.13149, 2020.

[36] Zhirong Wu, Yuanjun Xiong, X Yu Stella, and Dahua Lin. Unsupervised feature learning via non-parametric instance discrimination. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018.

[37] Jiahao Xie, Xiaohang Zhan, Z. Liu, Y. S. Ong, and Chen Change Loy. Delving into inter-image invariance for unsupervised visual representations. *ArXiv*, abs/2008.11702, 2020.

[38] Sangdoo Yun, Dongyoon Han, Seong Joon Oh, Sanghyuk Chun, Junsuk Choe, and Youngjoon Yoo. Cutmix: Regularization strategy to train strong classifiers with localizable features. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 6023–6032, 2019.

[39] Xiaohang Zhan, Jiahao Xie, Ziwei Liu, Yew-Soon Ong, and Chen Change Loy. Online deep clustering for unsupervised representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6688–6697, 2020.

[40] Xiaohang Zhan, Jiahao Xie, and Enze Xie. OpenSelfSup. https://github.com/open-mmlab/OpenSelfSup, 2020.

[41] Hongyi Zhang, Moustapha Cisse, Yann N Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. In *International Conference on Learning Representations*, 2018.

[42] Richard Zhang, Phillip Isola, and Alexei A Efros. Split-brain autoencoders: Unsupervised learning by cross-channel prediction. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1058–1067, 2017.

[43] Chengxu Zhuang, Alex Lin Zhai, and Daniel Yamins. Local aggregation for unsupervised learning of visual embeddings. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 6002–6012, 2019.