IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 1 December 2022, accepted 31 December 2022, date of publication 11 January 2023, date of current version 7 March 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3236087

RESEARCH ARTICLE

DimCL: Dimensional Contrastive Learning for Improving Self-Supervised Learning

THANH NGUYEN, (Graduate Student Member, IEEE), TRUNG XUAN PHAM, (Student Member, IEEE), CHAONING ZHANG, TUNG M. LUU, (Graduate Student Member, IEEE), THANG VU, (Graduate Student Member, IEEE), AND CHANG D. YOO, (Senior Member, IEEE)

School of Electrical Engineering, Korea Advanced Institute of Science and Technology, Daejeon 34141, Republic of Korea

Corresponding author: Chang D. Yoo (cd_yoo@kaist.ac.kr)

This work was supported in part by the Institute for Information and Communications Technology Promotion (IITP) Grant funded by the Korea Government [Ministry of Science and ICT (MSIT)] through the Development of Causal Artificial Intelligence (AI) through Video Understanding and Reinforcement Learning, and Its Applications to Real Environments under Grant 2021-0-01381; and in part by the IITP Grant funded by the Korea Government (MSIT) through the Development and Study of AI Technologies to Inexpensively Conform to Evolving Policy on Ethics under Grant 2022-0-00184.

ABSTRACT Self-supervised learning (SSL) has gained remarkable success, for which contrastive learning (CL) plays a key role. However, the recent development of new non-CL frameworks has achieved comparable or better performance with high improvement potential, prompting researchers to enhance these frameworks further. Assimilating CL into non-CL frameworks has been thought to be beneficial, but empirical evidence indicates no visible improvements. In view of that, this paper proposes a strategy of performing CL along the dimensional direction instead of along the batch direction as done in conventional contrastive learning, named Dimensional Contrastive Learning (DimCL). DimCL aims to enhance the feature diversity, and it can serve as a regularizer to prior SSL frameworks. DimCL has been found to be effective, and the hardness-aware property is identified as a critical reason for its success. Extensive experimental results reveal that assimilating DimCL into SSL frameworks leads to performance improvement by a non-trivial margin on various datasets and backbone architectures.

INDEX TERMS Self-supervise learning, computer vision, contrastive learning, deep learning, transfer learning.

I. INTRODUCTION

The success of self-supervised learning (SSL) has been demonstrated in a wide range of applications, ranging from early attempts in natural language processing [19], [40], [48], [51], [55] to more recent computer vision tasks [13], [21], [42]. To be more specific, in contrast to supervised learning which requires a huge amount of labeled data [18], [63], [63], [73], SSL learns the representations without the need for labeled ones. Thus, it significantly reduces the human-label cost and enables machine learning to learn from a massive amount of available unlabeled data leading to benefits for many real-world applications in various fields: teaching robots to work from raw pixel images [41], [44], [45], [47], training medical diagnosis systems from un-labels checkups results [8], [37], enhance 3D face reconstruction using images in the wild [60], . . .

Without using human annotation labels, SSL methods seek to learn an encoder with augmentation-invariant representation [3], [7], [9], [25], [29]. A common approach is to minimize the distance between two representations of positive samples, *i.e.* two augmented views of the same image. Based on this simple approach, the past few years have witnessed the development of various SSL frameworks, which can be roughly divided into two categories: CL-based and non-CL frameworks. The CL-based frameworks [3], [9], [29], [31], [33], [50], [58], [65], [68], [70], [75] have achieved remarkable developments and greatly

The associate editor coordinating the review of this manuscript and approving it for publication was Mingbo Zhao.

contributed to the progress of SSL. Recently, multiple works [5], [12], [22], [25], [71] have also demonstrated successful attempts with the non-CL frameworks, among which BYOL [12] and SimSiam [9] are the two representatives.

Compared with the CL-based frameworks, the non-CL ones [9], [12] have a unique advantage: they propose simpler frameworks without using the negative samples, yet achieve comparable or even superior performance on benchmark datasets (like ImageNet-1K and CIFAR-10/100). Thus, there is a trend to shift from CL to non-CL frameworks. Recognizing the significance of CL in the development of SSL, this work attempts to distill beneficial properties of CL to push the frontiers of non-CL frameworks further. However, naively assimilating CL to non-CL does not show visible improvement, as pointed out in BYOL [25]. This can be attributed to the fact that the frameworks mentioned above focus on the same inter-instance level of constraints and mainly pursue the same objective (augmentation invariant). In essence, existing CL encourages representation diversity among the instances in the batch. In this paper, CL is utilized to encourage diversity among the representation elements in obtaining "*feature diversity*", referred to as Dimensional Contrastive Learning. To avoid any confusion between batch contrastive learning and dimensional contrastive learning, we denote each as BCL and DimCL, respectively. The difference between BCL and DimCL is depicted in Fig. 2.

A prudent variation in BCL led to a separate SSL framework, while the proposed DimCL (as illustrated in Fig. 1) is designed as a regularizer for feature diversity enhancement to support other frameworks. Even though DimCL is originally motivated to boost non-CL frameworks, empirically, DimCL is found to also enhance the performance of existing CL-based frameworks and can be generalized to other domains (e.g,. supervised learning). This implies that feature diversity is necessary for good representations.

Our contributions are as follows:

- Recognizing the significance of CL in the development of self-supervised learning, we are the first to apply DimCL to push the frontiers of non-CL frameworks. In contrast to existing BCL, our proposed DimCL performs CL along the dimensional direction and can be used as a regularizer for boosting the performance of non-CL (and CL-based) frameworks.
- We perform extensive experiments on various frameworks with different backbone architectures on diverse datasets to validate the effectiveness of our proposed DimCL. We also investigate the reason for the benefit brought by DimCL and identify the hardness-aware property as an essential factor.

The rest of this paper is organized as follows. Section II summarizes the related works. Section III describes the background of Batch Contrastive Learning. Section IV presents the proposed method DimCL. Section V provides the experiment setup and results. Section VI shows the ablation study on important hyper-parameters. Section VII provides some discussions about DimCL. Finally, Section VII concludes this work.

II. RELATED WORK

A. CONTRASTIVE LEARNING

Contrastive learning (CL) is one of the prominent keystones of self-supervised learning. It fosters discriminability in the representation [23], [46], [53], [54], [67]. Early works have studied margin-based contrastive losses [27], [32], [67]. After the advent of [50], [70], NCE-based loss has become the standard loss in CL. Inspired by this success, CL has been extensively studied for SSL pretext training [3], [9], [31], [33], [50], [57], [70], [75]. SimCLR [9] proposes a simple yet effective method to train the unsupervised model. They show that more negative samples (4096, for instance) are beneficial for performance. However, such a massive number of negative samples require a huge batch size for training to achieve the desired performance.

MoCo v1 [29] has attracted significant attention by demonstrating superior performance over supervised pre-training counterparts in downstream tasks while making use of large negative samples, decoupling the need for batch size by introducing a dynamic dictionary. Inspired by [9], MoCo v2 [10] applies stronger augmentations and an additional MLP projector, which shows significant performance improvement over the first version of MoCo. [14] has empirically shown that the predictor from the non-CL frameworks [12], [25] helps to gain performance boost for MoCo variants with ViT structures [20].

Several works explain the key properties that lead to the success of CL. It is noticeable that momentum update [9] and large negative samples play an important role in preventing collapse. InfoNCE loss was identified to have the hardness-aware property, which is critical for optimization [64] and preventing collapse by instance de-correlation [1]. [15], [34], [36], [49], [62], [66], [69] have demonstrated that hard negative samples mining strategies can be beneficial for better performance over the baselines. Notably, [65] identified CL form alignment and uniformity of feature space which benefits downstream tasks.

Most of the contrastive learning frameworks adopt the instance discrimination task which inevitably causes class collision problems [74] where the representations of the same class images are forced to be different. The problem can hurt the quality of the learned representation. Different from the above methods, which perform CL along the batch direction, DimCL performs the CL along the dimensional direction in order to encourage diversity among representation elements instead of representation vectors. This approach never faces class collision problems.

B. NON-CONTRASTIVE LEARNING

Non-contrastive learning focuses on making augmentation invariant without using negative samples. With the absence of negative samples, training the simple siamese network using the cosine similarity loss leads to complete collapse

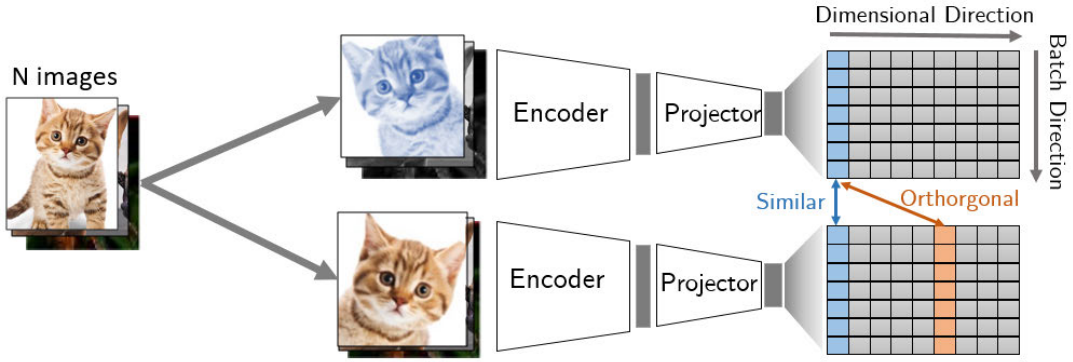


FIGURE 1. Dimensional contrastive learning (DimCL). As the term suggests, existing BCL performs CL along the batch direction to encourage diversity of representations, while our proposed DimCL performs CL along the dimensional direction to encourage diversity among elements within a representation (termed feature diversity). Our DimCL can be used as a plug-and-play regularization method to improve non-CL (and CL-based) SSL frameworks.

[1], [25]. BYOL [25] and Simsiam [12] demonstrated that using a careful architecture design to break the architecture symmetry can avoid collapse. Specifically, a special 'predictor' network is added in conjunction with the exponential moving average update (BYOL) or with a stop gradient in one branch (Simsiam). Besides, several works have attempted to demystify the success of BYOL [25]. A recent work [24] has suggested that batch normalization (BN) plays a critical role in the success of BYOL; however, another work [52] refutes that claim by showing BYOL works without the need for BN.

Recognizing the strong points of CL in the development process, this work tries to distill beneficial properties of CL in a novel manner and use it as a regularizer to boost the performance of non-CL (and CL) based frameworks. Moreover, most non-CL frameworks aim to learn augmentation invariant representation which training often leads to trivial constant solutions (i.e., collapse) [6]. DimCL naturally avoids collapse as it encourages diversity in the solution which is a great complement to non-CL.

III. BACKGROUND

Conventional contrastive learning, *i.e.* BCL, aims to make the representations similar if they are from different augmented versions of the same image and dissimilar if they are from different images. Or shortly, it aims to make meaningful discriminative representations. To be more specific, in BCL, there are query, positive, and negative samples. The considered image is called the query sample. The augmentation views of the query image are called positive samples. The other images in the sample batch and their augmentation views are called negative samples. The loss of CL-based frameworks basically makes the query representation to be near the positive sample presentation and far apart from the negative sample representation. Mathematically, given an encoder f, an input image is augmented and encoded as a query $q \in \mathbb{R}^D$ or positive key $k^+ \in \mathbb{R}^D$, which are often l_2-normalized to avoid scale ambiguity [9], [28]. Processing a mini-batch of N images will form a set of queries $\mathbb{Q} = \{q_1, q_2, \ldots, q_N\}$ and positive keys $\mathbb{K}^+ = \{k_1^+, k_2^+, \ldots, k_N^+\}$. Consider a query q_i, the corresponding negative keys are defined $\mathbb{K}_i^- = \mathbb{Q} \cup \mathbb{K}^+ \setminus \{q_i, k_i^+\} = \{k_1^-, k_2^-, \ldots, k_{2N-2}^-\}$ [9]. With similarity measured by dot product, BCL can be achieved by the simple CL loss below [64]:

$$\mathcal{L} = \frac{1}{N}\sum_{i=1}^{N}\mathcal{L}_i$$
$$\mathcal{L}_i = -q_i{\cdot}k_i^+ + \frac{1}{2N-2}\sum_{j=1}^{2N-2} q_i{\cdot}k_j^-. \quad (1)$$

The gradient of $\mathcal{L}_i$ w.r.t q_i is derived as:

$$\frac{\partial \mathcal{L}_i}{\partial q_i} = -k_i^+ + \frac{1}{2N-2}\sum_{j=1}^{2N-2} k_j^-. \quad (2)$$

The above equation treats all negative keys equally. Based on this, [64] proved that the superficial loss in Eq. 1 performs poorly in practice.

The NCE-based loss [26], [50] has been independently developed with various motivations in multiple popular works [54], [70] and it has become the standard loss for BCL. Following [1], [29], [50], we term it InfoNCE for consistency. The InfoNCE is formulated as follows [28]:

$$\mathcal{L}^{BCL} = \frac{1}{N}\sum_{i=1}^{N}\mathcal{L}_i^{BCL}$$
$$\mathcal{L}_i^{BCL} = -\log\frac{\exp(q_i{\cdot}k_i^+/\tau)}{\exp(q_i{\cdot}k_i^+/\tau) + \sum_{j=1}^{2N-2}\exp(q_i{\cdot}k_j^-/\tau)}, \quad (3)$$

with τ denoting the temperature. The InfoNCE has been identified to outperform the above simple loss Eq. 1 due to its hardness-aware property, which puts more weight on optimizing hard negative pairs (where the query is close to negative keys) as shown in [64].

IV. METHODOLOGY

Dimensional Contrastive Learning (DimCL) explores a new way of using InfoNCE compared to BCL. As shown in Fig. 2, BCL aims to make meaningful discriminative representations by applying InfoNCE along the batch direction. The keys

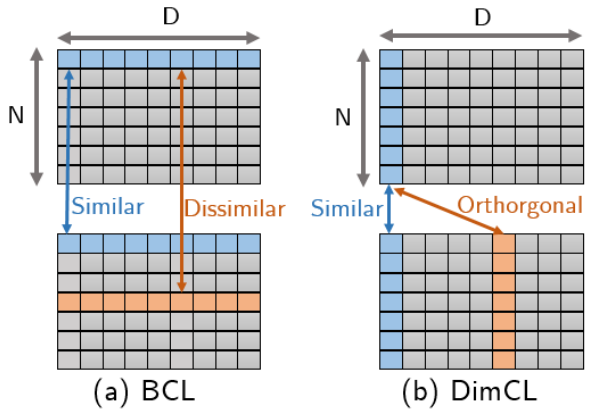


FIGURE 2. The difference between (a) Batch Contrastive Learning (BCL) and (b) Dimensional Contrastive Learning (DimCL). BCL performs along the batch direction to encourage representation diversity whereas DimCL performs along the dimensional direction to encourage feature diversity. N is the batch size, and D is the feature dimension.

and queries are the representation vectors. By contrast, DimCL encourages each representation element to contain a piece of distinct information to maximize the amount of information contained in the overall representation, which is *feature diversity* enhancement.[1] To this end, the DimCL make the elements of the representation vector orthogonal to each other in term of information by minimizing the empirical correlation among column vectors. A novel form of InfoNCE along the dimensional direction is proposed as the loss to achieve this objective. Therein, the corresponding queries and keys are *column vectors*, each of which is formed from the same-index representation elements within a batch, as highlighted in Fig. 2.

Mathematically, similar to BCL, given a mini-batch of N images, we have a set of queries $\mathbb{G} = \{g_1, g_2, \ldots, g_D\}$ and positive keys $\mathbb{H}^+ = \{h_1^+, h_2^+, \ldots, h_D^+\}$. Note that $g, h \in \mathbb{R}^N$ are *column vectors*. Considering a query g_i, the corresponding negative keys are defined as $\mathbb{H}_i^- = \mathbb{G} \cup \mathbb{H}^+ \setminus \{g_i, h_i^+\} = \{h_1^-, h_2^-, \ldots, h_{2D-2}^-\}$. In order to maximize the feature diversity, the considered query g_i should be orthogonal with all negative keys $\mathbb{H}_i^-$. The corresponding objective is:

$$\mathcal{L}^{AbsCL} = \frac{1}{D}\sum_{i=1}^{D}\mathcal{L}_i^{AbsCL}$$
$$\mathcal{L}_i^{AbsCL} = -\log\frac{\exp(g_i{\cdot}h_i^+/\tau)}{\exp(g_i{\cdot}h_i^+/\tau)+\sum_{j=1}^{2D-2}\exp(|g_i{\cdot}h_j^-|/\tau)}. \quad (4)$$

Empirically, we observe that the original InfoNCE is sufficient to achieve the objective without any modification (e.g., adding the absolute) (evidence is provided in the discussion). This can be explained by considering the exp term and the effect of temperature τ. With small τ, the $\exp(x/\tau)$ has high weight on pushing positive value x toward zero with a corresponding high gradient but has almost no consideration on negative value x with the same magnitude due to its much smaller gradient. For simplicity, we adopt the following loss as the DimCL optimization target:

$$\mathcal{L}^{DimCL} = \frac{1}{D}\sum_{i=1}^{D}\mathcal{L}_i^{DimCL}$$
$$\mathcal{L}_i^{DimCL} = -\log\frac{\exp(g_i{\cdot}h_i^+/\tau)}{\exp(g_i{\cdot}h_i^+/\tau)+\sum_{j=1}^{2D-2}\exp(g_i{\cdot}h_j^-/\tau)}. \quad (5)$$

Note that, in DimCL each query g_i has total $2D-2$ negative keys instead of $2N-2$ as in BCL. And each of *collumn vector* g, h are l_2-normalized along the batch direction instead of dimensional direction as in BCL. Furthermore, the proposed DimCL inherits the hardness-aware property of the traditional BCL for which we provide more detail in the discussion part.

Contrary to BCL, which works as an independent SSL framework, DimCL serves as a regularizer to benefit existing SSL frameworks. We denote $\mathcal{L}^{BASE}$ as the loss of the SSL baseline. DimCL can be simply assimilated into the baseline by a linear combination to form a final loss as:

$$\mathcal{L} = \lambda\mathcal{L}^{DimCL} + (1-\lambda)\mathcal{L}^{BASE}, \quad (6)$$

where $\lambda \in [0, 1]$ is a weight factor to balance the two loss components. We perform a grid search and find that $\lambda = 0.1$ works well in most cases and recommend this value as a starting point for more fine-grained tuning. The pseudo algorithm is provided in Algorithm. 1

Algorithm 1 Pytorch-Style Pseudocode for DimCL

```
f: encoder network, f': target encoder
N: batch size, D: dimension
τ: temperature, λ: balance weight
L_base: baseline loss, Optim: optimizer
for x in loader(N) do                          ▷ Load batches with N samples
    y_a, y_b = augment(x)                      ▷ Augmentations of x
                                               ▷ Compute representations
    z_a = f(y_a)                               ▷ N*D
    z_b = f'(y_b)                              ▷ N*D
                                               ▷ Get queries and positives
    G = [z_a[:, i] for i in range(D)]
    H+ = [z_b[:, i] for i in range(D)]
    L_dimCL = 0
    for i in range(D) do
        H- = G ∪ H+ \ {G[i], H[i]}             ▷ 2D-2 elements
        L_dimCL = L_dimCL + L_i^DimCL          ▷ Equation 5
    end for
    L_dimCL = L_dimCL / D
    Loss = λ L_dimCL + (1 − λ) L_base(z_a, z_b)
                                               ▷ Optimization step
    Loss.backward()
    Optim.step()
end for
```

[1]Note that feature diversity is defined as the independence among the elements of a representation. It should not be related to diversity among representation vectors.

V. EXPERIMENTS

A. EXPERIMENT SETUP

To show its effectiveness, we evaluate DimCL by assimilating it to state-of-the-art non-CL and CL-based frameworks. Five widely used benchmark datasets are considered including CIFAR-10 [38], CIFAR-100 [38], STL-10 [16], ImagetNet-100 [57], and ImageNet-1K (1000 classes) [39]. Different encoders (ResNet-18, ResNet-50) are also considered. The performance is bench-marked with linear classification evaluation and transfer learning with object detection following the common evaluation protocol in [12], [25], and [29]. To be more specific, the encoder is pre-trained in an unsupervised manner on the training set of the selected dataset without labels [39]. For the linear classification evaluation, the pre-trained frozen encoder is evaluated by training an additional linear classifier and tested on the corresponding test set. For object detection evaluation, the pre-trained frozen encoder is evaluated by a Faster R-CNN detector (C4-backbone) with the object detection datasets (i.e., VOC object detection). In this paper, the Faster R-CNN detector (C4-backbone) is finetuned on the VOC train-val 07+12 set with standard 2x schedule and tested on the VOC test2007 set [29], [68]. More details regarding the two evaluation methods are provided in Appendix

B. IMPLEMENTATION DETAILS

For a simple implementation, DimCL directly uses the InfoNCE loss [9] but transposes the input. BCL framework implementations are based on the open library solo-learn [61]. Setups of the SSL baseline framework for training are described below.

1) IMAGE AUGMENTATIONS

The paper follows the setting in previous approach [9], [25]. Concretely, a patch of the image is sampled and resized to 224×224. Random horizontal flips and color distortion are applied in turn. The color distortion is a random sequence of saturation, contrast, brightness, hue adjustments, and an optional grayscale conversion. Gaussian blur and solarization are applied to the patches at the final.

2) TRAINING

We use stochastic gradient descent (SGD) as the optimizer. The SGD weight decay is set to 1e-5, and the SGD momentum is 0.9 as BYOL [25]. We use a batch size of 256, and a single GPU for all methods except in benchmark ImageNet-1K for which a mini-batch size of 64×8 to train on an 8-GPUs machine (NVIDIA Titan Xp) is used. As a standard practice, the learning rate is decayed using the cosine scheduler with ten epochs warm-up at the beginning [43]. For baselines, we use the optimal set of hyperparameters tuned by [61]. We re-train all baselines in the same environment for a fair comparison. The balance weight factor λ is set to 0.1. The temperature τ is set to 0.1 for all experiments.

C. EXPERIMENTAL RESULTS

This session reports the results with four settings to prove the efficacy of DimCL: (1) Compatibility and generalization (experiments are performed with 200 epochs across non-CL (and CL) frameworks, datasets, and backbones) (2) Large-scale dataset (experiments are performed with 100 epochs on ImageNet-1K) (3) Longer Training (experiments are conducted with 1000 epochs on CIFAR-100, ImageNet-100), and (4) Transfer Learning on Object Detection.

1) COMPATIBILITY AND GENERALIZATION

To show the compatibility with various SSL frameworks and generalization across datasets and backbones, we provide the extensive result as shown in Tab. 1. The results demonstrate that assimilating DimCL consistently improves the performance by a large margin for all frameworks (MoCo v2, SimCLR, BYOL, SimSiam), datasets (CIFAR-10, CIFAR-100, STL-10, ImageNet-100), and backbones (ResNet-18, ResNet-50).

For example, on CIFAR-100 with Resnet-50, DimCL enhances the baseline MoCo v2 and SimCLR with a performance boost of 1.56% and +4.46%, respectively. A more significant performance boost can be observed for BYOL (+6.63%), and SimSiam (+11.4%). In addition, during the experiment, the BASEs are highly tuned to get the best performance, and BASEs+DimCL does not. With a fine-tuned parameter search, a higher gain might be possible. Overall, the result indicates DimCL is compatible with both CL and non-CL SSL frameworks with a non-trivial performance gain. Furthermore, it also has good generalization across various datasets and backbones.

When evaluating the performance of DimCL under different metrics, the result suggests the same conclusion. To be more specific, an experiment is conducted on CIFAR100 with Resnet-18 backbone. The pre-trained models of the baselines and DimCL are evaluated on the classification task with different performance metrics: Top-1 Accuracy, Top-5 Accuracy, Top-1 KNN, and Top-5 KNN. The result, shown in Tab. 2, suggests that DimCL consistently improves the baseline under various performance metrics.

2) LARGE-SCALE DATASET

For the large-scale dataset, Imagenet-1K is chosen, and BYOL is selected as the baseline. Due to the resource constraint, BYOL and BYOL+DimCL are pre-trained for 100 epochs without labels. The results are reported in Tab. 3. The results show that on the large-scale dataset, DimCL improves the BYOL baseline with a performance boost of +2.0% and outperforms all other frameworks. The performance is consistent with the results in Tab. 1, verifying the generalization and effectiveness of DimCL.

3) LONGER TRAINING

To demonstrate the results are consistent between short training (200 epochs) and long training (1000 epochs), we conduct experiments on CIFAR-100 and ImageNet-100

TABLE 1. The top-1 classification test accuracy (%) of the BASEs (the baseline frameworks) + DimCL (the baseline with DimCL regularization) amongst various datasets, and backbones. All models are trained for 200 epochs Classification is performed with a linear classifier trained on top of the frozen pre-trained encoder (output of the evaluated framework). "*" denotes an improved version of MoCo v2 with symmetric loss.

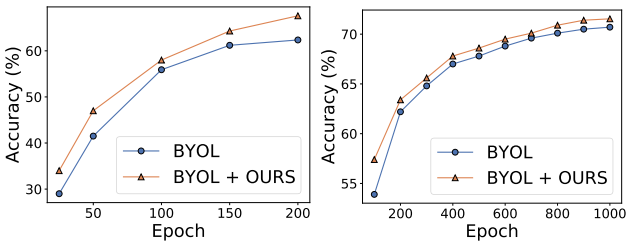
Datasets	Method	Type	ResNet-18			ResNet-50		
			BASE	+ DimCL	Δ_{acc}	BASE	+ DimCL	Δ_{acc}
CIFAR-10	MoCo v2 [28]*	CL	89.55	**89.59**	+ 0.04	90.60	**91.12**	+ 0.60
	SimCLR [9]		86.01	**88.32**	+ 2.31	86.71	**89.67**	+ 2.96
	BYOL [25]	Non-CL	88.51	**90.57**	+ 2.06	88.0	**89.98**	+ 1.98
	SimSiam [12]		83.33	**88.22**	+ 4.89	84.60	**89.33**	+ 4.73
CIFAR-100	MoCo v2 [28]*	CL	62.79	**64.04**	+ 1.25	64.68	**66.24**	+ 1.56
	SimCLR [9]		58.21	**61.75**	+ 3.54	60.81	**65.27**	+ 4.46
	BYOL [25]	Non-CL	62.36	**67.85**	+ 5.49	64.71	**70.94**	+ 6.23
	SimSiam [12]		51.67	**62.49**	+ 10.82	54.00	**65.40**	+ 11.4
STL-10	MoCo v2 [28]*	CL	85.96	**86.34**	+ 0.38	88.16	**88.40**	+ 0.24
	SimCLR [9]		82.35	**82.73**	+ 0.48	84.33	**86.28**	+ 1.95
	BYOL [25]	Non-CL	83.36	**84.94**	+ 1.58	83.83	**87.89**	+ 4.05
	SimSiam [12]		84.24	**84.35**	+ 0.11	86.13	**87.14**	+ 1.01
ImageNet-100	MoCo v2 [28]*	CL	76.02	**78.38**	+ 2.36	82.36	**83.18**	+ 0.82
	SimCLR [9]		75.96	**76.52**	+ 0.56	80.86	**81.78**	+ 1.12
	BYOL [25]	Non-CL	77.30	**80.72**	+ 3.42	81.74	**84.80**	+ 3.06
	SimSiam [12]		70.64	**76.08**	+ 5.42	72.98	**80.20**	+ 7.22

TABLE 2. Performance evaluated with different metrics. The methods are trained on the CIFAR-100 dataset with 200 epochs and use Resnet-18 as the backbone. Classification is performed with a linear classifier trained on top of the frozen pre-trained encoder. The test accuracy is reported with various performance metrics: Top-1 accuracy, Top-5 accuracy, Top-1 KNN, and Top-5 KNN.

Metrics	Top-1 Acc		Top-5 Acc		Top-1 KNN		Top-5 KNN	
Methods	BASE	+DimCL	BASE	+DimCL	BASE	+DimCL	BASE	+DimCL
MoCo V2	62.79	**64.04**	88.75	**89.3**	57.16	**57.82**	80.81	**81.61**
SimCLR	58.21	**61.75**	84.97	**87.72**	51.7	**54.85**	76.61	**77.82**
BYOL	62.36	**67.85**	88.51	**90.87**	56.2	**59.91**	80.27	**81.94**
SimSiam	51.67	**62.49**	81.65	**88.17**	50.11	**55.12**	76.19	**79.65**

TABLE 3. Imagenet-1K classification. All frameworks are trained without labels on the training set for 100 epochs. Evaluation is on a single crop 224 × 224. "†" denotes the results employed from [12].

Method	Top-1 (%)	Top-5 (%)
MoCo v2 [11]†	67.4	-
BYOL [25]†	66.5	-
BYOL (Reproduce)	67.3	88.0
BYOL + DimCL	**69.3**	**89.0**



(a) 200 epochs (b) 1000 epochs

FIGURE 3. Top-1 classification accuracy learning curve on the test set of CIFAR-100 of (a) 200 epochs and (b) 1000 epochs. The figure shows the consistent result between long and short training. Note that at the same epoch, the Top-1 accuracy of two settings is not necessarily the same due to using the cosine learning rate scheduler.

with BYOL as the baseline framework [25]. Top-1 classification accuracies are reported in Tab. 4.

We observe that DimCL also has a consistent performance boost for the long training. Specifically, incorporating DimCL helps to significantly boost the top-1 accuracy of BYOL from 70.54% to **71.94%** (+1.4%) for CIFAR-100, and further improves BYOL from 81.24% to **82.51%** for ImageNet-100. It is reasonable that the performance boost margin can be relatively smaller in the setup of the long training compared to the short training.

Fig. 3 shows the learning curve in two different settings: 200 epochs (a) and 1000 epochs (b). The results demonstrate

TABLE 4. Long training with 1000 epochs. Linear classification accuracy (%) on the test set of CIFAR-100. All models are pre-trained on the training set without labels before evaluation. Note that MoCo v2+ is the improved version of MoCo v2 with symmetric loss [17].

Dataset	Epoch	SimCLR	MoCo v2	BYOL	BYOL+DimCL
CIFAR-100	200	58.21	62.79	62.36	**67.85**
	1000	65.85	69.39	70.54	**71.94**
ImageNet-100	200	75.96	76.02	77.30	**80.72**
	1000	78.76	79.98	81.24	**82.51**

TABLE 5. Transfer learning on detection task VOC07. The † denotes the results from [68].

Method	Epoch	AP	AP50	AP75
SimCLR [9] †	200	51.5	79.4	55.6
BYOL [25] †	200	51.9	81.0	56.5
BYOL [25]	100	50.3	79.8	54.2
BYOL + DimCL	100	**55.6**	**81.9**	**61.4**

that our method does not vanish but further improves BYOL in long training. There is a high correlation in performance improvement between short and long training. It proves that the 200 epochs setting is reasonably adequate to evaluate the performance gain.

4) TRANSFER LEARNING ON OBJECT DETECTION

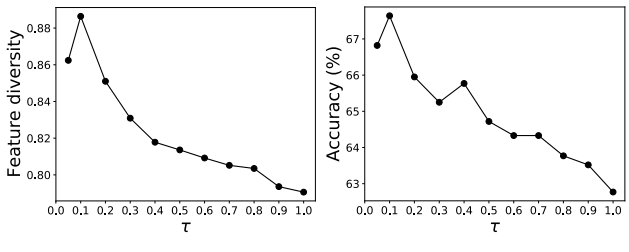
Tab. 5 shows Objection detection evaluation on pre-trained frozen backbone ResNet-50 in Tab. 3. DimCL significantly boosts BYOL in the object detection task by a large margin. Specifically, in 100 epoch pre-training on ImageNet-1K with BYOL [25], the encoder gives 50.3, 79.8, 54.2 in AP, AP50, AP75, respectively. Doubling pre-training epochs with BYOL, *i.e.* 200 epochs, the encoder shows a slight improvement. By contrast, in 100 epoch pre-training with BYOL+DimCL, the encoder can strongly outperform BYOL for the AP, AP50, AP75 with **55.6, 81.9, 61.4**, and even surpass the performance of the baseline 200 epoch pre-training with BYOL for all metrics. It demonstrates the effectiveness of the proposed DimCL. Overall, the results demonstrate DimCL is effective for both downstream tasks: *classification* and *object detection*.

VI. ABLATION STUDY

In this section, we provide ablation for important hyper-parameters of DimCL: the temperature τ, the weight factor λ, and the dimensionality D.

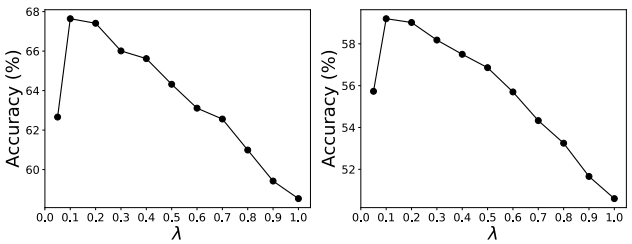
A. THE EFFECT OF THE TEMPERATURE τ

We monitor the changes in feature diversity and performance when assimilating DimCL to BYOL with various τ values. The experiment runs on CIFAR-100 for 200 epochs. The results in Fig. 4 suggest that selecting a reasonable τ leads to high feature diversity (and performance). $\tau = 1$ does not lead to good feature diversity. The value of τ that supports gaining the best performance is around 0.1. This result coincides with



(a) Feature diversity (b) Top-1 linear accuracy

FIGURE 4. Feature diversity (a) and performance (b) with respect to τ on the test set of CIFAR-100. Our hypothesis emphasizes the importance of increasing the feature diversity or decreasing the correlation to remove the residual information of the feature representation.



(a) Top-1 linear accuracy (b) KNN accuracy

FIGURE 5. Top-1 classification accuracy and top-1 KNN with respect to λ on the test set of CIFAR-100. Note that, the performance at $\lambda = 0$, corresponding to the performance of the baseline BYOL, is much lower than the case when incorporating BYOL with our loss.

the τ used in conventional BCL frameworks [9], [29]. There is a drop in performance when using too large or too small τ in DimCL.

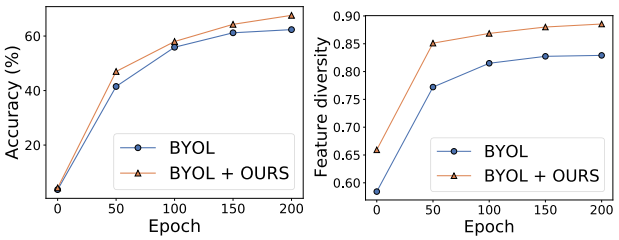
B. THE EFFECT OF THE WEIGHT FACTOR λ

The balance weight factor between DimCL and the baseline plays an important role in gaining performance. We conduct the experiments with a range of [0, 1] for λ in Eq. 6. All other parameters are kept unchanged.

The results in Fig. 5 show that with all λ in the range of (0, 0.7), our method consistently outperforms the baseline BYOL (corresponding to $\lambda = 0$) in both two measures: top-1 classification accuracy and top-1 KNN accuracy. $\lambda = 0.1$ is found to be the optimal value to boost performance when plugging DimCL into BYOL. λ usually depends on the baselines and dataset. However, we empirically find that setting λ to 0.1 often gives the best performance for the most recent SSL frameworks in datasets. It is recommended to use this value at the beginning of the tuning process when using our DimCL regularization.

TABLE 6. The effects of DimCL on dimensionality. The table shows the top 1 accuracy on CIFAR 100 with 200 epochs of BYOL and BYOL+DimCL.

Dimensionality	64	128	256	512	1024	2048	8192
BYOL	59.85	60.72	62.36	62.62	62.44	62.02	62.99
BYOL+DimCL	66.47	66.84	67.85	67.41	67.18	67.41	67.33
Improvement	6.62	6.12	5.49	4.79	4.74	5.39	4.34



(a) Performance (b) Feature diversity

FIGURE 6. Relation between feature diversity and performance during training on CIFAR-100. a) the top-1 test classification accuracy. b) the corresponding feature diversity. The higher feature diversity leads to higher performance.

C. THE EFFECT OF THE DIMENSIONALITY D

As the DimCL targets to address dimension-wise diversity, dimensionality should be a key fact that needs to be considered. We provide ablation studies on the effects of dimensionality. Tab. 6 shows the top 1 accuracy on CIFAR 100 with 200 epochs of BYOL and BYOL+DimCL.

The result shows that for the small dimensionality, DimCL provides a large improvement over the baseline. For bigger dimensionality, the improvement tends to reduce. It is understandable since DimCL aims to maximize the useful information (or in other words, minimize the redundancy) contained in a low dimensionality. For bigger dimensionality, there is plenty of space for storing information which reduces the importance of DimCL. It is also noticeable that for very small dimensionality, the performance starts to drop for both BYOL and BYOL+DimCL (e.g: under 256) since there is not much space for storing information.

VII. DISCUSSION

A. FEATURE DIVERSITY ENHANCEMENT

Our proposed DimCL is motivated to enhance feature diversity which is defined as the independence among the elements of a representation. In other words, good feature diversity means each element of representation should carry a piece of distinct information about the input image. In this view, feature diversity can be evaluated by considering correlation among all pairs of negative *column vectors*. Given a tensor with size $N \times D$, the feature diversity measure is defined as:

$$featurediversity = 1 - \frac{1}{D(D-1)} \sum_{i}^{D} \sum_{j \neq i}^{D} |sim(g_i \cdot h_j)|. \quad (7)$$

Here, $g, h \in \mathbb{R}^N$ are *column vectors*. $sim(.)$ is the cosine similarity measure. The range for the *feature diversity* measure is within $[0, 1]$. The optimum value of *feature diversity* is 1 which means all elements of representation are mutually independent.

To prove the enhancement of feature diversity, we take BYOL [25] and SimSiam [12] into account where the encoders are designed to learn the representation, which is invariant to augmentation without considering feature diversity. We assimilate DimCL to BYOL, and Simsiam then observe changes in feature diversity and accuracy. Results are reported in Tab. 7.

Interestingly, the BASEs generate embedding, which already has high feature diversity. Adding DimCL to BASEs has a strong effect on further increasing the feature diversity, which impacts performance improvement. Specifically, DimCL makes an improvement 0.05 (5% in percentage) feature diversity with corresponding 5.49% accuracy on BYOL and 0.17 (17% in percentage) feature diversity with corresponding 10.82% accuracy on SimSiam. The more feature diversity improvement, the better performance gain. The relation between feature diversity is shown clearly in Fig. 6.

From the perspective of information theory, improving feature diversity can be classified as the Information Bottleneck objective [59] which forces a representation that conserves as much information about the sample as possible. It is mentioned to be beneficial in various research [2], [35], [59]. Our result is one of the empirical pieces of evidence proving the benefit of feature diversity.

B. HARDNESS-AWARE PROPERTY IN DimCL

The hardness-aware property plays a key role in BCL controlling the uniformity-tolerance dilemma [65] leading to its success. In the view of optimization, the hardness-aware property puts more weight into optimizing negative pairs that have high similarities. This way is influenced by hard examples mining and has proven to be effective [4], [36], [49], [62], [66], [72].

The interpretation of Harness-aware in the DimCL can be understood via gradient analysis of loss function. Let's consider loss for query g_i:

$$\mathcal{L}_i^{DimCL} = -\log \frac{\exp(g_i \cdot h_i^+/\tau)}{\exp(g_i \cdot h_i^+/\tau) + \sum_j \exp(g_i \cdot h_j^-/\tau)} \quad (8)$$

where g and h are the l_2-normalized column vectors. The gradient of $\mathcal{L}_i^{DimCL}$ w.r.t query g_i is derived as:

$$\frac{\partial \mathcal{L}_i^{DimCL}}{\partial g_i} = -\frac{1}{\tau}\left(1 - \frac{\exp(g_i \cdot h_i^+/\tau)}{\exp(g_i \cdot h_i^+/\tau) + \sum_j \exp(g_i \cdot h_j^-/\tau)}\right).h_i^+ \quad (9)$$

$$+\frac{1}{\tau} \frac{\sum_j \exp(g_i \cdot h_j^-/\tau).h_j^-}{\exp(g_i \cdot h_i^+/\tau) + \sum_j \exp(g_i \cdot h_j^-/\tau)} \quad (10)$$

$$= -\frac{1}{\tau}(1 - \alpha_i').h_i^+ + \frac{1}{\tau} \sum_j \alpha_j.h_j^-, \quad (11)$$

where $\alpha_i' = \frac{\exp(g_i \cdot h_i^+/\tau)}{\exp(g_i \cdot h_i^+/\tau) + \sum_j \exp(g_i \cdot h_j^-/\tau)}$ can be interpreted as the probability of g_i being recognized as the positive columnm

TABLE 7. Comparison of Feature diversity and performance in CIFAR-100 dataset for both BASE (baseline) and +DimCL (baseline with DimCL regularization). All frameworks are pre-trained with 200 epochs on ResNet-18 backbone.

Method	Feature Diversity		Accuracy Top-1		Accuracy KNN	
	BASE	+ DimCL	BASE	+ DimCL	BASE	+ DimCL
BYOL [25]	0.83	**0.88** (+0.05)	62.36	**67.85** (+5.49)	56.20	**59.91** (+3.71)
SimSiam [12]	0.75	**0.92** (+0.17)	51.67	**62.49** (+10.82)	50.11	**55.12** (+5.01)

TABLE 8. DimCL for improving Barlow Twins (BT) [71]. Frameworks are trained for 200 epochs with ResNet-18 and ResNet-50 backbone on the 4 datasets. We report top-1 linear classification (%) accuracy.

Datasets	ResNet-18		ResNet-50	
	BT	+ DimCL	BT	+ DimCL
CIFAR-10	88.45	**89.21**	88.91	**90.28**
CIFAR-100	65.61	**66.42**	66.35	**66.88**
STL-10	82.26	**82.66**	84.99	**85.34**
IMAGENET-100	78.50	**78.72**	82.44	**82.78**

TABLE 9. Comparison between DimCL and Barlow Twins on top of baseline BYOL. Models are trained for 200 epochs with ResNet-18 on the CIFAR-100. We report top-1 linear classification (%) accuracy.

Dataset	BYOL+DimCL	BYOL+Barlow Twins
CIFAR-100	**67.85**	66.55

vector h_i^+. Similarly, $\alpha_j = \frac{\exp(g_i \cdot h_j^- / \tau)}{\exp(g_i \cdot h_i^+ / \tau) + \sum_j \exp(g_i \cdot h_j^- / \tau)}$ can be interpreted as the probability of g_i being recognized as the negative vector h_j^-. We can easily see that $\alpha_i' + \sum_j \alpha_j = 1$ and all $\alpha > 0$.

The Eq. 11 reveals how DimCL makes the query similar to the positive key and dissimilar from negative keys. Concretely, if g_i and h_i^+ are very close, the gradient of g_i is very small because $1 - \alpha_i' \approx 0$ and $\sum_j \alpha_j \approx 0$ (because $1 - \alpha_i' \approx 0$ and $\alpha_i' + \sum_j \alpha_j = 1$). Thus, the optimizer does not update the query g_i. By contrast, if g_i and h_j^- are very close, the weight α_j is big, encouraging the optimizer to push the query far away from the corresponding negative keys.

Regarding the ability to differently treat negative keys, the gradient weight w.r.t negative keys are proportional to the exponential $\exp(\frac{g_i \cdot h_j^-}{\tau})$. It shows that hard column pairs, where query g_i is far with negative keys, are penalized more with larger α_j. In other words, the optimizer will pay more attention to optimizing hard column pairs, which leads to better optimization results than treating them equally. This phenomenon is the hardness-awareness property of the loss 8. The effect of the hardness-aware property in DimCL in relation to feature diversity can be empirically seen clearly in the ablation study Fig. 4

C. BEYOND CL AND NON-CL

Previous results show that DimCL is most beneficial in boosting the performance of CL and non-CL frameworks with a non-trivial margin. Here, we also investigate the recent work that designed an explicit term for decorrelation, Barlow Twins (BT) [71].

TABLE 10. DimCL for improving supervised learning. Models are trained for 200 epochs with ResNet-18 and ResNet-50 backbone on the 4 datasets. We report top-1 linear classification (%) accuracy.

Dataset	Supervised	+ DimCL (%)
CIFAR-100	70.27	**71.68**
CIFAR-10	93.29	**93.35**

We experiment by adding the correlation-reduction loss of BT to the previous baseline BYOL and comparing it against DimCL. The result in Tab. 9 shows that BYOL+DimCL strongly outperforms BYOL+Barlow. Furthermore, as shown in Tab. 8, When incorporate into BT, DimCL can also improve BT.

This empirical result recommends that DimCL provides better performance than BT.

D. DimCL FOR SUPERVISED LEARNING

Since DimCL works as a regularizer enhancing the feature diversity, it is expected to benefit other fields beyond self-supervised learning (e.g. supervised learning (SL)). This experiment utilizes DimCL to boost SL on CIFAR-100 and CIFAR-10 datasets. We use the solo-learn library [61] to train the supervised model with backbone ResNet-18 [30].

DimCL is assimilated with cross-entropy loss for training the model simultaneously. Tab. 10 shows the top-1 classification accuracy on the test set. For CIFAR-10 DimCL shows slight improvement, while CIFAR-100 shows the DimCL supports to boost the conventional supervised learning from 70.27% to **71.68%** (+1.4%), demonstrating the benefit of DimCL for SL.

E. DimCL VERSUS AbsCL

In order to maximize the feature diversity, the considered query g_i should be orthogonal with all negative keys $\mathbb{H}_i^-$. The corresponding objective is:

$$\mathcal{L}^{AbsCL} = \frac{1}{D}\sum_{i=1}^{D} \mathcal{L}_i^{AbsCL}$$
$$\mathcal{L}_i^{AbsCL} = -\log \frac{\exp(g_i \cdot h_i^+ / \tau)}{\exp(g_i \cdot h_i^+ / \tau) + \sum_{j=1}^{2D-2} \exp(|g_i \cdot h_j^-| / \tau)}. \quad (12)$$

Empirically, Tab. 11 shows that the original InfoNCE is sufficient to achieve the objective without any modification (e.g., adding the absolute). It is important to note that without τ, DimCL and AbsCL can outperform the baseline. However, to achieve the best performance, τ is needed to present.

TABLE 11. DimCL versus AbsCL. We report top-1 linear test accuracy (%) on CIFAR-10 and CIFAR-100. All methods are trained for 200 epochs. For $\tau = 0.1$, all methods DimCL and AbsCL perform best and performance is almost similar.

Datasets	BYOL				SimSiam			
	Baseline	τ	DimCL	AbsCL	Baseline	τ	DimCL	AbsCL
CIFAR-10	88.51	1	88.92	89.81	83.33	1	86.07	86.59
		0.1	**90.57**	**90.83**		0.1	**88.22**	**87.67**
CIFAR-100	62.34	1	62.77	66.02	51.67	1	54.67	57.56
		0.1	**67.85**	**67.87**		0.1	**62.49**	**62.96**

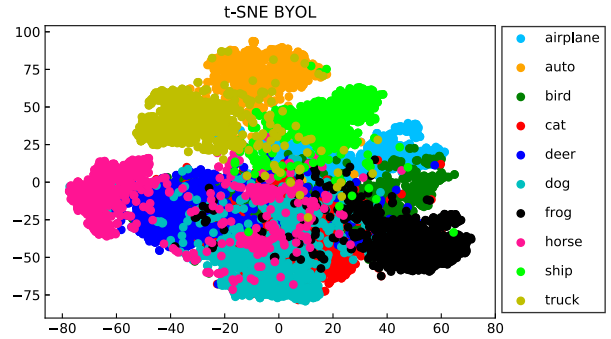


FIGURE 7. t-SNE plot of ten classes for data trained by the BYOL baseline in 200 epochs with accuracy = 88.51% in CIFAR-10 with 10,000 samples of the test set.

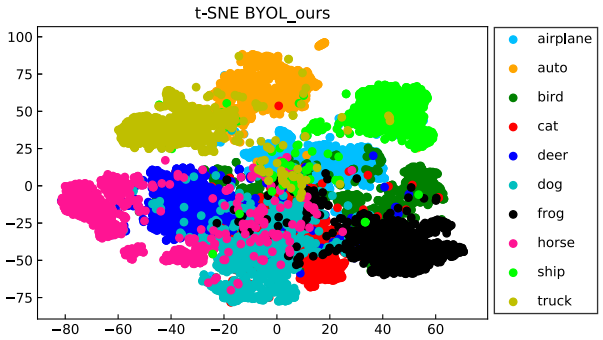


FIGURE 8. t-SNE plot of ten classes for data trained by the BYOL + DimCL in 200 epochs with accuracy = 90.57% in CIFAR-10 with 10,000 samples of the test set.

At the optimal $\tau = 0.1$, the performance of DimCL is nearly the same as AbsCL. This phenomenon can be explained by considering the exp term and the effect of temperature τ. With small τ, the $\exp(x/\tau)$ has a high weight on pushing positive value x toward zero with a corresponding high gradient but has almost no consideration on negative value x with the same magnitude due to its much smaller gradient.

F. VISUALIZATION OF REPRESENTATION.

Visualization of representation via t-SNE is reported to see the effect of DCL on representation space. Fig. 7 and Fig. 8 show the representation of BYOL baseline and our method on the 2D space. The experiment is conducted on CIFAR-10 with 10 classes. The results clearly show that our method in Fig. 8 gives more separable representations. More specifically, airplane, auto, ship, and truck are almost separable among them and also from other animal classes. All classes are scatted in the more compact clusters compared to the baseline in Fig. 7.

TABLE 12. Inter-class distance and Inter-class distance on CIFAR-10 test set.

	Intra-class distance ↓	Inter-class distance ↑
BYOL	27.0	82.3
BYOL+DCL	**24.1**	**85.9**

To show the difference between the two representation spaces quantitatively, intra-class distance and inter-class distance [56] are calculated and provided in Tab. 12. The quantitative result agrees that BYOL+DCL forms the more compact clusters while maintaining a higher separation among different clusters compared to BYOL.

VIII. CONCLUSION

This paper introduces Dimensional Contrastive Learning (DimCL), a new way of applying CL. DimCL works as a regularization that can assimilate with non-CL (and CL) based frameworks to boost performance on downstream tasks such as classification and object detection. DimCL enhances feature diversity among elements within a representation. DimCL has high compatibility and generalization across datasets, frameworks, and backbone architectures. We believe that feature diversity is a key indispensable ingredient for learning representation. This paper focuses on images and provides mostly empirical evidence but DimCL can be generalized to other modalities (e.g. audio, video, text) and proven with theoretical results. We let it for future work.

ACKNOWLEDGMENT

The authors would like to thank the KAIST U-AIM lab and prof. Chang D. Yoo for supporting this work. *(Thanh Nguyen and Trung Xuan Pham contributed equally to this work.)*

REFERENCES

[1] C. Zhang, K. Zhang, C. D. Yoo, and I. S. Kweon, "How does Simsiam avoid collapse without negative samples? Towards a unified understanding of progress in SSL," in *Proc. 10th Int. Conf. Learn. Represent. (ICLR)*, 2022.

[2] B. O. Ayinde, T. Inanc, and J. M. Zurada, "Regularizing deep neural networks by enhancing diversity in feature extraction," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 30, no. 9, pp. 2650–2661, Sep. 2019.

[3] P. Bachman, R. D. Hjelm, and W. Buchwalter, "Learning representations by maximizing mutual information across views," 2019, *arXiv:1906.00910*.

[4] Y. Bai and T. Liu, "Me-momentum: Extracting hard confident examples from noisily labeled data," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 9312–9321.

[5] A. Bardes, J. Ponce, and Y. LeCun, "VICReg: Variance-invariance-covariance regularization for self-supervised learning," 2021, *arXiv:2105.04906*.

[6] H. Barlow, "Redundancy reduction revisited," *Netw., Comput. Neural Syst.*, vol. 12, no. 3, p. 241, 2001.

[7] M. Caron, I. Misra, J. Mairal, P. Goyal, P. Bojanowski, and A. Joulin, "Unsupervised learning of visual features by contrasting cluster assignments," 2020, *arXiv:2006.09882*.

[8] L. Chen, P. Bentley, K. Mori, K. Misawa, M. Fujiwara, and D. Rueckert, "Self-supervised learning for medical image analysis using image context restoration," *Med. Image Anal.*, vol. 58, Dec. 2019, Art. no. 101539.

[9] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton, "A simple framework for contrastive learning of visual representations," in *Proc. Int. Conf. Mach. Learn.* PMLR, 2020, pp. 1597–1607.

[10] X. Chen, H. Fan, R. Girshick, and K. He, "Improved baselines with momentum contrastive learning," 2020, *arXiv:2003.04297*.

[11] X. Chen, H. Fan, R. Girshick, and K. He, "Improved baselines with momentum contrastive learning," 2020, *arXiv:2003.04297*.

[12] X. Chen and K. He, "Exploring simple Siamese representation learning," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 15750–15758.
[13] X. Chen, S. Xie, and K. He, "An empirical study of training self-supervised vision transformers," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 9640–9649.
[14] X. Chen, S. Xie, and K. He, "An empirical study of training self-supervised vision transformers," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 9640–9649.
[15] C. Y. Chuang, J. Robinson, Y. C. Lin, A. Torralba, and S. Jegelka, "Debiased contrastive learning," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 33, 2020, pp. 8765–8775.
[16] A. Coates, A. Ng, and H. Lee, "An analysis of single-layer networks in unsupervised feature learning," in *Proc. 14th Int. Conf. Artif. Intell. Statist.*, 2011, pp. 215–223.
[17] V. G. T. da Costa, E. Fini, M. Nabi, N. Sebe, and E. Ricci, "Solo-learn: A library of self-supervised methods for visual representation learning," *J. Mach. Learn. Res.*, vol. 23, no. 56, pp. 1–6, 2022.
[18] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "ImageNet: A large-scale hierarchical image database," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2009, pp. 248–255.
[19] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," in *Proc. Conf. North Amer. Chapter Assoc. Comput. Linguistics, Hum. Lang. Technol.*, vol. 1, Jun. 2019, pp. 4171–4186.
[20] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby, "An image is worth 16×16 words: Transformers for image recognition at scale," in *Proc. ICLR*, 2021.
[21] A. El-Nouby, H. Touvron, M. Caron, P. Bojanowski, M. Douze, A. Joulin, I. Laptev, N. Neverova, G. Synnaeve, J. Verbeek, and H. Jegou, "XCiT: Cross-covariance image transformers," 2021, *arXiv:2106.09681*.
[22] A. Ermolov, A. Siarohin, E. Sangineto, and N. Sebe, "Whitening for self-supervised representation learning," in *Proc. Int. Conf. Mach. Learn.* PMLR, 2021, pp. 3015–3024.
[23] M. Federici, A. Dutta, P. Forré, N. Kushman, and Z. Akata, "Learning robust representations via multi-view information bottleneck," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2020.
[24] A. Fetterman and J. Albrecht. (2020). *Understanding Self-Supervised and Contrastive Learning With 'Bootstrap Your Own Latent' (Byol)*. [Online]. Available: https://untitled-ai.github.io/understanding-self-supervisedcontrastive-learning.html
[25] J.-B. Grill, F. Strub, F. Altché, C. Tallec, P. H. Richemond, E. Buchatskaya, C. Doersch, B. A. Pires, Z. D. Guo, M. G. Azar, B. Piot, K. Kavukcuoglu, R. Munos, and M. Valko, "Bootstrap your own latent: A new approach to self-supervised learning," 2020, *arXiv:2006.07733*.
[26] M. Gutmann and A. Hyvärinen, "Noise-contrastive estimation: A new estimation principle for unnormalized statistical models," in *Proc. 13th Int. Conf. Artif. Intell. Statist.*, 2010, pp. 297–304.
[27] R. Hadsell, S. Chopra, and Y. LeCun, "Dimensionality reduction by learning an invariant mapping," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, vol. 2, Jul. 2006, pp. 1735–1742.
[28] K. He, H. Fan, Y. Wu, S. Xie, and R. Girshick, "Momentum contrast for unsupervised visual representation learning," 2019, *arXiv:1911.05722*.
[29] K. He, H. Fan, Y. Wu, S. Xie, and R. Girshick, "Momentum contrast for unsupervised visual representation learning," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 9729–9738.
[30] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 770–778.
[31] O. Henaff, "Data-efficient image recognition with contrastive predictive coding," in *Proc. Int. Conf. Mach. Learn.*, 2020, pp. 4182–4192.
[32] A. Hermans, L. Beyer, and B. Leibe, "In defense of the triplet loss for person re-identification," 2017, *arXiv:1703.07737*.
[33] R. D. Hjelm, A. Fedorov, S. Lavoie-Marchildon, K. Grewal, P. Bachman, A. Trischler, and Y. Bengio, "Learning deep representations by mutual information estimation and maximization," 2018, *arXiv:1808.06670*.
[34] C.-H. Ho and N. Vasconcelos, "Contrastive learning with adversarial examples," 2020, *arXiv:2010.12050*.
[35] T. Hua, W. Wang, Z. Xue, S. Ren, Y. Wang, and H. Zhao, "On feature decorrelation in self-supervised learning," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 9598–9608.
[36] A. Iscen, G. Tolias, Y. Avrithis, and O. Chum, "Mining on manifolds: Metric learning without labels," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 7642–7651.
[37] R. Krishnan, P. Rajpurkar, and E. J. Topol, "Self-supervised learning in medicine and healthcare," *Nature Biomed. Eng.*, vol. 6, pp. 1346–1352, Aug. 2022.
[38] A. Krizhevsky, "Learning multiple layers of features from tiny images," Tech. Rep., 2009.
[39] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," *Commun. ACM*, vol. 60, no. 6, pp. 84–90, 2017.
[40] Z. Lan, M. Chen, S. Goodman, K. Gimpel, P. Sharma, and R. Soricut, "Albert: A lite bert for self-supervised learning of language representations," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2020.
[41] M. Laskin, A. Srinivas, and P. Abbeel, "Curl: Contrastive unsupervised representations for reinforcement learning," in *Proc. Int. Conf. Mach. Learn.*, 2020, pp. 5639–5650.
[42] C. Li, J. Yang, P. Zhang, M. Gao, B. Xiao, X. Dai, L. Yuan, and J. Gao, "Efficient self-supervised vision transformers for representation learning," 2021, *arXiv:2106.09785*.
[43] I. Loshchilov and F. Hutter, "SGDR: Stochastic gradient descent with warm restarts," in *Proc. ICLR*, 2017.
[44] T. M. Luu, T. Nguyen, T. Vu, and C. D. Yoo, "Utilizing skipped frames in action repeats for improving sample efficiency in reinforcement learning," *IEEE Access*, vol. 10, pp. 64965–64975, 2022.
[45] T. M. Luu, T. Vu, T. Nguyen, and C. D. Yoo, "Visual pretraining via contrastive predictive model for pixel-based reinforcement learning," *Sensors*, vol. 22, no. 17, p. 6504, Aug. 2022.
[46] I. Misra, C. L. Zitnick, and M. Hebert, "Shuffle and learn: Unsupervised learning using temporal order verification," in *Proc. Eur. Conf. Comput. Vis.* Cham, Switzerland: Springer, 2016, pp. 527–544.
[47] T. Nguyen, T. M. Luu, T. Vu, and C. D. Yoo, "Sample-efficient reinforcement learning representation learning with curiosity contrastive forward dynamics model," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Sep. 2021, pp. 3471–3477.
[48] P. Nie, Y. Zhang, X. Geng, A. Ramamurthy, L. Song, and D. Jiang, "DC-BERT: Decoupling question and document for efficient contextual encoding," in *Proc. 43rd Int. ACM SIGIR Conf. Res. Develop. Inf. Retr.*, Jul. 2020, pp. 1829–1832.
[49] K. Nozawa and I. Sato, "Understanding negative samples in instance discriminative self-supervised representation learning," 2021, *arXiv:2102.06866*.
[50] A. van den Oord, Y. Li, and O. Vinyals, "Representation learning with contrastive predictive coding," 2018, *arXiv:1807.03748*.
[51] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, and I. Sutskever, "Language models are unsupervised multitask learners," *OpenAI Blog*, vol. 1, no. 8, p. 9, 2019.
[52] P. H. Richemond, J.-B. Grill, F. Altché, C. Tallec, F. Strub, A. Brock, S. Smith, S. De, R. Pascanu, B. Piot, and M. Valko, "BYOL works even without batch statistics," 2020, *arXiv:2010.10241*.
[53] F. Schroff, D. Kalenichenko, and J. Philbin, "FaceNet: A unified embedding for face recognition and clustering," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 815–823.
[54] K. Sohn, "Improved deep metric learning with multi-class n-pair loss objective," in *Proc. Adv. Neural Inf. Process. Syst.*, 2016, pp. 1857–1865.
[55] W. Su, X. Zhu, Y. Cao, B. Li, L. Lu, F. Wei, and J. Dai, "VL-bert: Pre-training of generic visual-linguistic representations," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2020.
[56] A. Taufik and S. S. S. Ahmad, "A comparative study of fuzzy c-means and k-means clustering techniques," in *Malays. Tech. Univ. Conf. Eng. Technol. 8th (MUCET)*, vol. 1, 2014, pp. 10–11.
[57] Y. Tian, D. Krishnan, and P. Isola, "Contrastive multiview coding," 2019, *arXiv:1906.05849*.
[58] Y. Tian, D. Krishnan, and P. Isola, "Contrastive multiview coding," in *Proc. ECCV*. Cham, Switzerland: Springer, 2020, pp. 776–794.
[59] N. Tishby and N. Zaslavsky, "Deep learning and the information bottleneck principle," in *Proc. IEEE Inf. Theory Workshop (ITW)*, Apr. 2015, pp. 1–5.
[60] X. Tu, J. Zhao, M. Xie, Z. Jiang, A. Balamurugan, Y. Luo, Y. Zhao, L. He, Z. Ma, and J. Feng, "3D face reconstruction from a single image assisted by 2D face images in the wild," *IEEE Trans. Multimedia*, vol. 23, pp. 1160–1172, 2021.

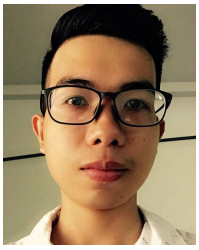
[61] V. G. T. da Costa, E. Fini, M. Nabi, N. Sebe, and E. Ricci, "Solo-learn: A library of self-supervised methods for visual representation learning," 2021, *arXiv:2108.01775*.
[62] B. Vasudeva, P. Deora, S. Bhattacharya, U. Pal, and S. Chanda, "LoOp: Looking for optimal hard negative embeddings for deep metric learning," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 10634–10643.
[63] T. Vu, K. Kim, T. M. Luu, T. Nguyen, and C. D. Yoo, "SoftGroup for 3D instance segmentation on point clouds," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 2708–2717.
[64] F. Wang and H. Liu, "Understanding the behaviour of contrastive loss," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 2495–2504.
[65] T. Wang and P. Isola, "Understanding contrastive representation learning through alignment and uniformity on the hypersphere," in *Proc. Int. Conf. Mach. Learn.* PMLR, 2020, pp. 9929–9939.
[66] W. Wang, W. Zhou, J. Bao, D. Chen, and H. Li, "Instance-wise hard negative example generation for contrastive learning in unpaired image-to-image translation," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 14020–14029.
[67] X. Wang and A. Gupta, "Unsupervised learning of visual representations using videos," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 2794–2802.
[68] X. Wang, R. Zhang, C. Shen, T. Kong, and L. Li, "Dense contrastive learning for self-supervised visual pre-training," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 3024–3033.
[69] M. Wu, M. Mosse, C. Zhuang, D. Yamins, and N. Goodman, "Conditional negative sampling for contrastive learning of visual representations," 2020, *arXiv:2010.02037*.
[70] Z. Wu, Y. Xiong, S. X. Yu, and D. Lin, "Unsupervised feature learning via non-parametric instance discrimination," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 3733–3742.
[71] J. Zbontar, L. Jing, I. Misra, Y. LeCun, and S. Deny, "Barlow twins: Self-supervised learning via redundancy reduction," in *Proc. Int. Conf. Mach. Learn.* PMLR, 2021, pp. 12310–12320.
[72] K. Zeng, M. Ning, Y. Wang, and Y. Guo, "Hierarchical clustering with hard-batch triplet loss for person re-identification," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 13657–13665.
[73] X. Zhai, A. Kolesnikov, N. Houlsby, and L. Beyer, "Scaling vision transformers," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 12104–12113.
[74] M. Zheng, F. Wang, S. You, C. Qian, C. Zhang, X. Wang, and C. Xu, "Weakly supervised contrastive learning," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 10042–10051.
[75] C. Zhuang, A. Zhai, and D. Yamins, "Local aggregation for unsupervised learning of visual embeddings," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 6002–6012.

THANH NGUYEN (Graduate Student Member, IEEE) received the B.Sc. degree in electronic and automation engineering from the Ho Chi Minh City University of Science and Technology, in 2015. He is currently pursuing the M.Sc. and Ph.D. degrees with the Korea Advanced Institute of Science and Technology. His research interests include machine learning, deep learning, and reinforcement learning.

TRUNG XUAN PHAM (Student Member, IEEE) received the B.S. degree from the School of Electronics and Telecommunications (SET), Hanoi University of Science and Technology (HUST), in 2014. He is currently pursuing the Ph.D. degree with the Korea Advanced Institute of Science and Technology (KAIST) under the supervision of Prof. Chang D. Yoo. His research interests include speech processing, self-supervised learning, and computer vision.

CHAONING ZHANG received the dual master's degrees from the Delft University of Technology, The Netherlands, and the Harbin Institute of Technology, China, and the Ph.D. degree from the Korea Advanced Institute of Science and Technology (KAIST), South Korea. His research interests include intersection between computer vision and machine learning, intrigued by interesting ML topics, such as adversarial robustness, deep data hiding, and self-supervised learning.

TUNG M. LUU (Graduate Student Member, IEEE) received the B.Sc. degree in electronic and telecommunication engineering from the Hanoi University of Science and Technology, in 2017, and the M.Sc. degree in electrical engineering from the Korea Advanced Institute of Science and Technology, in 2020, where he is currently pursuing the Ph.D. degree. His research interests include machine learning, deep learning, and reinforcement learning.



THANG VU (Graduate Student Member, IEEE) received the B.Sc. degree in electronic and telecommunication engineering from the Hanoi University of Science and Technology, in 2016, and the M.Sc. degree in electrical engineering from the Korea Advanced Institute of Science and Technology, in 2019, where he is currently pursuing the Ph.D. degree. His research interests include machine learning and deep learning for computer vision.

CHANG D. YOO (Senior Member, IEEE) received the B.S. degree in engineering and applied science from the California Institute of Technology, the M.S. degree in electrical engineering from Cornell University, and the Ph.D. degree in electrical engineering from the Massachusetts Institute of Technology. From January 1997 to March 1999, he was a Senior Researcher at Korea Telecom (KT). He also worked as the Dean of the Office of Special Projects and the Dean of the Office of International Relations. Since 1999, he has been a Faculty Member with the Korea Advanced Institute of Science and Technology (KAIST), where he is currently a Full Professor with tenure with the School of Electrical Engineering and an Adjunct Professor with the Department of Computer Science.