DECOUPLED CONTRASTIVE LEARNING

Anonymous authors
Paper under double-blind review

ABSTRACT

Contrastive learning (CL) is one of the most successful paradigms for self-supervised learning (SSL). In a principled way, it considers two augmented "views" of the same image as *positive* to be pulled closer, and all other images *negative* to be pushed further apart. However, behind the impressive success of CL-based techniques, their formulation often relies on heavy-computation settings, including large sample batches, extensive training epochs, etc. We are thus motivated to tackle these issues and aim at establishing a simple, efficient, and yet competitive baseline of contrastive learning. Specifically, we identify, from theoretical and empirical studies, a noticeable *negative-positive-coupling* (NPC) effect in the widely used cross-entropy (InfoNCE) loss, leading to unsuitable learning efficiency with respect to the batch size. Indeed the phenomenon tends to be neglected in that optimizing infoNCE loss with a small-size batch is effective in solving easier SSL tasks. By properly addressing the NPC effect, we reach a *decoupled contrastive learning* (DCL) objective function, significantly improving SSL efficiency. DCL can achieve competitive performance, requiring neither large batches in SimCLR, momentum encoding in MoCo, or large epochs. We demonstrate the usefulness of DCL in various benchmarks, while manifesting its robustness being much less sensitive to suboptimal hyperparameters. Notably, our approach achieves 66.9% ImageNet top-1 accuracy using batch size 256 within 200 epochs pre-training, outperforming its baseline SimCLR by 5.1%. With further optimized hyperparameters, DCL can improve the accuracy to 68.2%. We believe DCL provides a valuable baseline for future contrastive learning-based SSL studies.

1 INTRODUCTION

As a fundamental task in machine learning, representation learning aims to extract features to reconstruct the raw data fully. It has been regarded as a long-acting goal over the past decades. Recent progress on representation learning has achieved a significant milestone over self-supervised learning (SSL), facilitating feature learning with its competence in exploiting massive raw data without any annotated supervision. In the early stage of SSL, representation learning has focused on exploiting pretext tasks, which are addressed by generating pseudo-labels to the unlabeled data through different transformations, such as solving jigsaw puzzles (Noroozi & Favaro, 2016), colorization (Zhang et al., 2016) and rotation prediction (Gidaris et al., 2018). Though these approaches succeed in computer vision, there is a large gap between these methods and supervised learning. Recently, there has been a significant advancement in using contrastive learning (Wu et al., 2018; van den Oord et al., 2018; Tian et al., 2020a; He et al., 2020; Chen et al., 2020a) for self-supervised pre-training, which significantly closes the gap between the SSL method and supervised learning. Contrastive SSL methods, e.g., SimCLR (Chen et al., 2020a), in general, try to pull different views of the same instance close and push different instances far apart in the representation space.

Despite the evident progress of the state-of-the-art contrastive SSL methods, there have been several challenges in future developing this direction: 1) The SOTA models (He et al., 2020) may require unique structures like the momentum encoder and large memory queues, which may complicate the understanding. 2) The contrastive SSL models (Chen et al., 2020a) may depend on large batch size and huge epoch numbers to achieve competitive performance, posing a computational challenge for academia to explore this direction. 3) They may be sensitive to hyperparameters and optimizers, introducing additional difficulty to reproduce the results on various benchmarks.

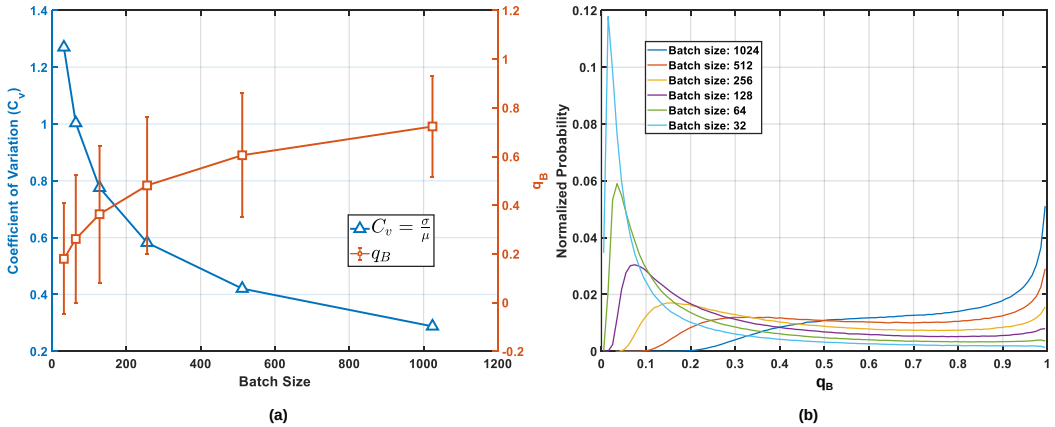


Figure 1: An overview of the batch size issue in the general contrastive approaches: (a) shows the NPC multiplier q_B in different batch sizes. As the large batch size increasing the q_B will approach 1 with a small coefficient of variation ($C_v = \sigma/\mu$). (b) illustrates the distribution of q_B.

Through our analysis of the widely adopted InfoNCE loss in contrastive learning, we identified a negative-positive-coupling (NPC) multiplier q_B in the gradient as shown in Proposition 1. The NPC multiplier modulates the gradient of each sample, and it reduces the learning efficiency when the SSL classification task is easy. A less informative positive view would reduce the gradient from a batch of informative negative samples or vice versa. Such a coupling exacerbates when smaller batch sizes are used. By removing the coupling term, we reach a new formulation, the *decoupled contrastive learning* (DCL). The new objective function significantly improves the training efficiency, requires neither large batches, momentum encoding, or large epochs to achieve competitive performance on various benchmarks. Specifically, DCL reaches 68.2% ImageNet top-1 (linear probing) accuracy with batch size 256, SGD optimizer within 200 epochs. Even if DCL is trained for 100 epochs, it still reaches 64.6% ImageNet top-1 accuracy with batch size 256.

The main contributions of the proposed DCL can be characterized as follows:

1) We provide both theoretical analysis and empirical evidence to show the negative-positive coupling in the gradient of InfoNCE-based contrastive learning;
2) We introduce a new, decoupled contrastive learning (DCL) objective, which casts off the coupling phenomenon between positive and negative samples in contrastive learning, and significantly improves the training efficiency; Additionally, the proposed DCL objective is less sensitive to several important hyperparameters;
3) We demonstrate our approach via extensive experiments and analysis on both large and small-scale vision benchmarks, with an optimal configuration for the standard SimCLR baseline to have a competitive performance within contrastive approaches. This leads to a plug-and-play improvement to the widely adopted InfoNCE contrastive learning methods.

2 Related Work

2.1 Self-supervised representation learning

Self-supervised representation learning (SSL) aims to learn a robust embedding space from data without human annotation. Previous arts can be roughly categorized into generative and discriminative. Generative approaches, such as autoencoders and adversarial learning, focus on reconstructing images from latent representations (Goodfellow et al., 2014; Radford et al., 2016). Conversely, recent discriminative approaches, especially contrastive learning-based approaches, have gained the most ground and achieved state-of-the-art standard large-scale image classification benchmarks with increasingly more compute and data augmentations.

2.2 CONTRASTIVE LEARNING

Contrastive learning (CL) constructs positive and negative sample pairs to extract information from the data itself. In CL, each anchor image in a batch has only one positive sample to construct a positive sample pair (Hadsell et al., 2006; Chen et al., 2020a; He et al., 2020). CPC (van den Oord et al., 2018) predicts the future output of sequential data by using current output as prior knowledge, which can improve the feature representing the ability of the model. Instance discrimination (Wu et al., 2018) proposes a non-parametric cross-entropy loss to optimize the model at the instance level. Inv. spread (Ye et al., 2019) makes use of data augmentation invariants and the spread-out property of instance to learn features. MoCo (He et al., 2020) proposes a dictionary to maintain a negative sample set, thus increasing the number of negative sample pairs. Different from the aforementioned self-supervised CL approaches, Khosla et al. (2020) proposes a supervised CL that considers all the same categories as positive pairs to increase the utility of images.

2.3 COLLAPSING ISSUE VIA BATCH SIZE AND NEGATIVE SAMPLE

In CL, the objective is to maximize the mutual information between the positive pairs. However, to avoid the "*collapsing output*", vast quantities of negative samples are needed so that the learning objectives obtain the maximum similarity and have the minimum similarity with negative samples. For instance, in SimCLR (Chen et al., 2020a), training requires many negative samples, leading to a large batch size (i.e., 4096). Furthermore, to optimize such a huge batch, a specially designed optimizer LARS (You et al., 2017) is used. Similarly, MoCo (He et al., 2020) needs a vast queue (i.e., 65536) to achieve competitive performance. BYOL (Grill et al., 2020) does not collapse output without using any negative samples by considering all the images are positive and to maximize the similarity of "projection" and "prediction " features. On the other hand, SimSiam (Chen & He, 2021) leverages the Siamese network to introduce inductive biases for modeling invariance. With the small batch size (i.e., 256), SimSiam is a rival to BYOL (i.e., 4096). Unlike both approaches that achieved their success through empirical studies, this paper tackles from a theoretical perspective, proving that an intertwined multiplier q_B of positive and negative is the main issue to contrastive learning.

2.4 CONTRASTIVE LEARNING ON BATCH SIZE SENSITIVITY

Recent literature discusses the losses for contrastive learning and focuses on batch size sensitivity. Tsai et al. (2021) start from the contrastive predictive code J_{CPC}, which is equivalent to SimCLR loss (Chen et al., 2020a), and then proposes a new term J_{RPC}. However, J_{RPC} is not the same as SimCLR loss or J_{CPC} in essence. J_{RPC} is more similar to the ranking loss (Chen et al., 2009), which collects and pushes away the positive pairs and negative pairs. Since the ranking loss is not stable enough, Tsai et al. (2021) add additional regularization terms to control the magnitude of the network and gains better results. On the other hand, it brings additional hyperparameters and needs more time to search for the best weight combinations. Hjelm et al. (2019) follow the (Belghazi et al., 2018) and extend the idea between the local and global features. Hence, (Hjelm et al., 2019) is quite different from the contrastive loss. Ozair et al. (2019) follow the approach of and proposes a Wasserstein distance to prevent the encoder from learning any other differences between unpaired samples. The starting point of this paper comes from SimCLR (Chen et al., 2020a) and then provides theoretical analysis to support why decoupling the positive and negative terms in contrastive loss is essential. The target problems are different though the motivations are similar.

3 DECOUPLE NEGATIVE AND POSITIVE SAMPLES IN CONTRASTIVE LEARNING

We choose to start from SimCLR because of its conceptual simplicity. Given a batch of N samples (e.g. images), $\{\mathbf{x}_1, \ldots, \mathbf{x}_N\}$, let $\mathbf{x}_i^{(1)}, \mathbf{x}_i^{(2)}$ be two augmented views of the sample x_i and B be the set of all of the augmented views in the batch, i.e. $B = \{\mathbf{x}_i^{(k)} | k \in \{1, 2\}, i \in [\![1, N]\!]\}$. As shown by Figure 2(a), each of the views $\mathbf{x}_i^{(k)}$ is sent into the same encoder network f and the output $\mathbf{h}_i^{(k)} = f(\mathbf{x}_i^{(k)})$ is then projected by a normalized MLP projector that $\mathbf{z}_i^{(k)} = g(\mathbf{h}_i^{(k)})/\|g(\mathbf{h}_i^{(k)})\|$. For each augmented view $\mathbf{x}_i^{(k)}$, SimCLR solves a classification problem by using the rest of the views in B as targets, and assigns the only positive label to $\mathbf{x}_i^{(u)}$, where $u \neq k$. So SimCLR

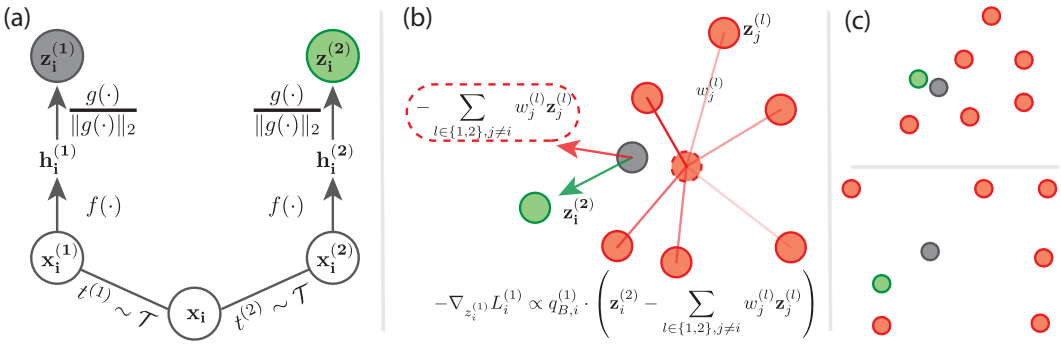


Figure 2: Contrastive learning and negative-positive coupling (NPC). (a) In SimCLR, each sample $\mathbf{x}_i$ has two augmented views $\{\mathbf{x}_i^{(1)}, \mathbf{x}_i^{(2)}\}$. They are encoded by the same encoder f and further projected to $\{\mathbf{z}_i^{(1)}, \mathbf{z}_i^{(2)}\}$ by a normalized MLP. (b) According to Equation 3. For the view $\mathbf{x}_i^{(1)}$, the cross-entropy loss $L_i^{(1)}$ leads to a positive force $\mathbf{z}_i^{(2)}$, which comes from the other view $\mathbf{x}_i^{(2)}$ of $\mathbf{x}$ and a negative force, which is a weighted average of all the negative samples, i.e. $\{\mathbf{z}_j^{(l)} | l \in \{1,2\}, j \neq i\}$. However, the gradient $-\nabla_{\mathbf{z}_i^{(2)}} L_i^{(1)}$ is proportional to the NPC multiplier. (c) We show two cases when the NPC term would affect the learning efficiency. On the top, the positive sample is close to the anchor and less informative. However, the gradient from the negative samples are also reduced. On the bottom, when the negative samples are far away and less informative, the learning rate from the positive sample is mistakenly reduced. In general, the NPC multiplier from the InfoNCE loss tend to make the SSL task simpler to solve, which leads to a reduced learning efficiency.

creates a cross-entropy loss function $L_i^{(k)}$ for each view $\mathbf{x}_i^{(k)}$, and the overall loss function is $L = \sum_{k\in\{1,2\}, i\in[\![1,N]\!]} L_i^{(k)}$.

$$L_i^{(k)} = -\log \frac{\exp(\langle \mathbf{z}_i^{(1)}, \mathbf{z}_i^{(2)} \rangle / \tau)}{\exp(\langle \mathbf{z}_i^{(1)}, \mathbf{z}_i^{(2)} \rangle / \tau) + \sum_{l\in\{1,2\}, j\in[\![1,N]\!], j\neq i} \exp(\langle \mathbf{z}_i^{(k)}, \mathbf{z}_j^{(l)} \rangle / \tau)} \tag{1}$$

Proposition 1. There exists a negative-positive coupling (NPC) multiplier $q_{B,i}^{(1)}$ in the gradient of $L_i^{(1)}$:

$$\begin{cases} -\nabla_{\mathbf{z}_i^{(1)}} L_i^{(1)} = \frac{q_{B,i}^{(1)}}{\tau} \left[\mathbf{z}_i^{(2)} - \sum_{l\in\{1,2\}, j\in[\![1,N]\!], j\neq i} \frac{\exp \langle \mathbf{z}_i^{(1)}, \mathbf{z}_j^{(l)} \rangle / \tau}{\sum_{q\in\{1,2\}, j\in[\![1,N]\!], j\neq i} \exp(\langle \mathbf{z}_i^{(1)}, \mathbf{z}_j^{(q)} \rangle / \tau)} \cdot \mathbf{z}_j^{(l)} \right] \\ -\nabla_{\mathbf{z}_i^{(2)}} L_i^{(1)} = \frac{q_{B,i}^{(1)}}{\tau} \cdot \mathbf{z}_i^{(1)} \\ -\nabla_{\mathbf{z}_j^{(l)}} L_i^{(1)} = -\frac{q_{B,i}^{(1)}}{\tau} \frac{\exp \langle \mathbf{z}_i^{(1)}, \mathbf{z}_j^{(l)} \rangle / \tau}{\sum_{q\in\{1,2\}, j\in[\![1,N]\!], j\neq i} \exp(\langle \mathbf{z}_i^{(1)}, \mathbf{z}_j^{(q)} \rangle / \tau)} \cdot \mathbf{z}_i^{(1)} \end{cases} \tag{2}$$

where the NPC multiplier $q_{B,i}^{(1)}$ is:

$$q_{B,i}^{(1)} = 1 - \frac{\exp(\langle \mathbf{z}_i^{(1)}, \mathbf{z}_i^{(2)} \rangle / \tau)}{\sum_{q\in\{1,2\}, j\in[\![1,N]\!], j\neq i} \exp(\langle \mathbf{z}_i^{(1)}, \mathbf{z}_j^{(q)} \rangle / \tau)} \tag{3}$$

Due to the symmetry, a similar NPC multiplier $q_{B,i}^{(k)}$ exists in the gradient of $L_i^{(k)}, k \in \{1,2\}, i \in [\![1,N]\!]$.

As we can see, all of the partial gradients in Equation 2 are modified by the common NPC multiplier $q_{B,i}^{(k)}$ in Equation 3. Equation 3 makes intuitive sense: when the SSL classification task is easy, the gradient would be reduced by the NPC term. However, the positive samples and negative samples are strongly coupled. When the negative samples are far away and less informative (easy negatives), the gradient from an informative positive sample would be reduced by the NPC multiplier $q_{B,i}^{(1)}$. On the

other hand, when the positive sample is close (easy positive) and less informative, the gradient from a batch of informative negative samples would also be reduced by the NPC multiplier. When the batch size is smaller, the SSL classification problem can be significantly simpler to solve. As a result, the learning efficiency can be significantly reduced with a small batch size setting.

Figure 1(b) shows the NPC multiplier q_B distribution shift w.r.t. different batch sizes for a pre-trained SimCLR baseline model. While all of the shown distributions have prominent fluctuation, the smaller batch size makes q_B cluster towards 0, while the larger batch size pushes the distribution towards $\delta(1)$. Figure 1(a) shows the averaged NPC multiplier $\langle q_B \rangle$ changes w.r.t. the batch size and the relative fluctuation. The small batch sizes introduce significant NPC fluctuation. Based on this observation, we propose to remove the NPC multipliers from the gradients, which corresponds to the case $q_{B,N\to\infty}$. This leads to the decoupled contrastive learning formulation. Wang et al. (2021a) also proposes an important loss which does not have the NPC. However, by a similar analysis that it introduces negative-negative coupling from different positive samples. In Section A.5, we provide a thorough discussion and demonstrate the advantage of DCL loss.

Proposition 2. Removing the positive pair from the denominator of Equation 1 leads to a decoupled contrastive learning loss. If we remove the NPC multiplier $q_{B,i}^{(k)}$ from Equation 2, we reach a decoupled contrastive learning loss $L_{DC} = \sum_{k\in\{1,2\},i\in[\![1,N]\!]} L_{DC,i}^{(k)}$, where $L_{DC,i}^{(k)}$ is:

$$L_{DC,i}^{(k)} = -\log \frac{\exp(\langle \mathbf{z}_i^{(1)}, \mathbf{z}_i^{(2)} \rangle / \tau)}{\cancel{\exp(\langle \mathbf{z}_i^{(1)}, \mathbf{z}_i^{(2)} \rangle / \tau)} + \sum_{l\in\{1,2\},j\in[\![1,N]\!],j\neq i} \exp(\langle \mathbf{z}_i^{(k)}, \mathbf{z}_j^{(l)} \rangle / \tau)} \tag{4}$$

$$= -\langle \mathbf{z}_i^{(1)}, \mathbf{z}_i^{(2)} \rangle / \tau + \log \sum_{l\in\{1,2\},j\in[\![1,N]\!],j\neq i} \exp(\langle \mathbf{z}_i^{(k)}, \mathbf{z}_j^{(l)} \rangle / \tau) \tag{5}$$

The proofs of Proposition 1 and 2 are given in Appendix. Further, we can generalize the loss function L_{DC} to L_{DCW} by introducing a weighting function for the positive pairs i.e. $L_{DCW} = \sum_{k\in\{1,2\},i\in[\![1,N]\!]} L_{DCW,i}^{(k)}$.

$$L_{DCW,i}^{(k)} = -w(\mathbf{z}_i^{(1)}, \mathbf{z}_i^{(2)})(\langle \mathbf{z}_i^{(1)}, \mathbf{z}_i^{(2)} \rangle / \tau) + \log \sum_{l\in\{1,2\},j\in[\![1,N]\!],j\neq i} \exp(\langle \mathbf{z}_i^{(k)}, \mathbf{z}_j^{(l)} \rangle / \tau) \tag{6}$$

where we can intuitively choose w to be a negative von Mises-Fisher weighting function that $w(\mathbf{z}_i^{(1)}, \mathbf{z}_i^{(2)}) = 2 - \frac{\exp(\langle \mathbf{z}_i^{(1)}, \mathbf{z}_i^{(2)} \rangle / \sigma)}{\mathrm{E}_i\left[\exp(\langle \mathbf{z}_i^{(1)}, \mathbf{z}_i^{(2)} \rangle / \sigma)\right]}$ and $\mathrm{E}[w] = 1$. L_{DC} is a special case of L_{DCW} and we can see that $\lim_{\sigma\to\infty} L_{DCW} = L_{DC}$. The intuition behind $w(\mathbf{z}_i^{(1)}, \mathbf{z}_i^{(2)})$ is that there is more learning signal when a positive pair of samples are far from each other.

4 Experiments

This section empirically evaluates our proposed decoupled contrastive learning (DCL) and compares it to general contrastive learning methods. We summarize our experiments and analysis as the following: (1) our proposed work significantly outperforms the general contrastive learning on large and small-scale vision benchmarks; (2) we show the better version of DCL: LDCW could further improve the representation quality. (3) we further analyze our DCL with ablation studies on ImageNet-1K, hyperparameters, and few learning epochs, which shows fast convergence of the proposed DCL. Detailed experimental settings can be found in the Appendix.

4.1 Implementation details

To understand the effect of the sample decoupling, we consider our proposed DCL based on general contrastive learning, where model optimization is irrelevant to the size of batches (i.e., negative samples). Extensive experiments and analysis are demonstrated on large-scale benchmarks: ImageNet-1K (Deng et al., 2009), ImageNet-100 (Tian et al., 2020a), and small-scale benchmark: CIFAR (Krizhevsky et al., 2009), and STL10 (Coates et al., 2011). Note that all of our experiments are conducted with 8 Nvidia V100 GPUs on a single machine.

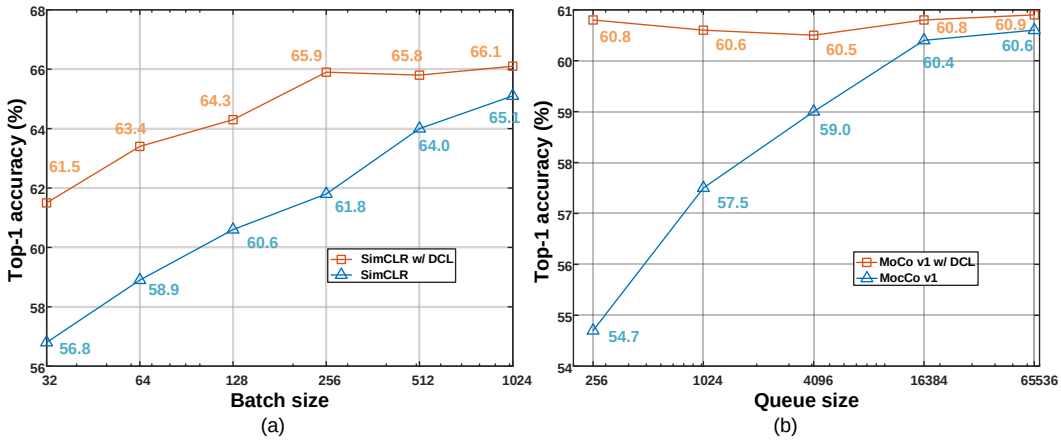


Figure 3: Comparisons on ImageNet-1K with/without DCL under different numbers of (a): batch sizes for SimCLR and (b): queues for MoCo. Without DCL, the top-1 accuracy significantly drops when batch size (SimCLR) or queues (MoCo) becomes very small. Note that the temperature τ of SimCLR is 0.1, and the temperature τ of MoCo is 0.07 in the comparison.

ImageNet For a fair comparison on ImageNet data, we implement our proposed decoupled structure, DCL, by following SimCLR (Chen et al., 2020a) with ResNet-50 (He et al., 2016) as the encoder backbone and use cosine annealing schedule with SGD optimizer. We set the temperature τ to 0.1 and the latent vector dimension to 128. Following the OpenSelfSup benchmark (Zhan et al., 2020), we evaluate the pre-trained models by training a linear classifier with frozen learned embedding on ImageNet data. We further consider evaluating our approach on ImageNet-100, a selected subset of 100 classes of ImageNet-1K.

CIFAR and STL10 For CIFAR10, CIFAR100, and STL10, ResNet-18 (He et al., 2016) is used as the encoder architecture. Following the small-scale benchmark of CLD (Wang et al., 2021b), we set the temperature τ to 0.07. All models are trained for 200 epochs with SGD optimizer, a base $lr = 0.03 * batchsize/256$, and k = 200 for nearest neighbor (kNN) classifier. Note that on STL10, we follow CLD to use both $train$ set and $unlabeled$ set for model pre-training. We further use ResNet-50 as a stronger backbone by adopting the implementation (Ren, 2020), using the same backbone and hyperparameters.

4.2 Experiments and Analysis

DCL on ImageNet This section illustrates the effect of our DCL under different batch sizes and queues. The initial setup is to have 1024 batch size (SimCLR) and 65536 queues (MoCo (He et al., 2020)) and gradually reduce the batch size (SimCLR) and queue (MoCo) to show the corresponding top-1 accuracy by linear evaluation. Figure 3 indicates that without DCL, the top-1 accuracy drastically drops when batch size (SimCLR) or queue (MoCo) becomes very small. While with DCL, the performance keeps steadier than baselines (SimCLR: -4.1% vs. -8.3%, MoCo: -0.4% vs. -5.9%).

Specifically, Figure 3 further shows that in SimCLR, the performance with DCL improves from 61.8% to 65.9% under 256 batch size; MoCo with DCL improves from 54.7% to 60.8% under 256 queues. The comparison fully demonstrates the necessity of DCL, especially when the number of negatives is small. Although batch size is increased to 1024, our DCL (66.1%) still improves over the SimCLR baseline (65.1%).

We further observe the same phenomenon on ImageNet-100 data. Table 1 shows that, while with DCL, the performance only drops 2.3% compare to the SimCLR baseline of 7.1%.

In summary, it is worth noting that, while the batch size is small, the strength of $q_{B,i}$, which is used to push the negative samples away from the positive sample, is also relatively weak. This phenomenon tends to reduce the efficiency of learning representation. While taking advantage of DCL alleviates

Table 1: Comparisons with/without DCL under different numbers of batch sizes from 32 to 512. Results show the effectness of DCL on four widely used benchmarks. The performance of DCL keeps steadier than the SimCLR baseline while the batch size is varied.

Architecture@epoch	ResNet-18@200 epoch									
Dataset	ImageNet-100 (linear)					STL10 (kNN)				
Batch Size	32	64	128	256	512	32	64	128	256	512
SimCLR	74.2	77.6	79.3	80.7	81.3	74.1	77.6	79.3	80.7	81.3
SimCLR w/ DCL	**80.8**	**82.0**	**81.9**	**83.1**	**82.8**	**82.0**	**82.8**	**81.8**	**81.2**	**81.0**
Dataset	CIFAR10 (kNN)					CIFAR100 (kNN)				
Batch Size	32	64	128	256	512	32	64	128	256	512
SimCLR	78.9	80.4	81.1	81.4	81.3	49.4	50.3	51.8	52.0	52.4
SimCLR w/ DCL	**83.7**	**84.4**	**84.4**	**84.2**	**83.5**	**51.1**	**54.3**	**54.6**	**54.9**	**55.0**
Architecture@epoch	ResNet-50@500 epoch									
SimCLR	82.2	-	88.5	-	89.1	49.8	-	59.9	-	61.1
SimCLR w/ DCL	**86.1**	-	**89.9**	-	**90.3**	**54.3**	-	**61.6**	-	**62.2**

Table 2: Comparisons between SimCLR baseline, DCL, and DCLW. The linear and kNN top-1 (%) results indicate that DCL improves the performance of baseline, and DCLW further provides an extra boost. Note that results are under the batch size 256 and epoch 200. All of models are both trained and evaluated with same experimental settings.

Dataset	CIFAR10	CIFAR100	ImageNet-100	ImageNet-1K
SimCLR	81.8	51.8	79.3	61.8
DCL	84.2 (+3.1)	54.6 (+2.8)	81.9 (+2.6)	65.9 (+4.1)
DCLW	**84.8** (+3.7)	**54.9** (+3.1)	**82.8** (+3.5)	**66.9** (+5.1)

the performance gap between small and large batch sizes. Hence, through the analysis, we find out DCL can simply tackle the batch size issue in contrastive learning. With this considerable advantage given by DCL, general SSL approaches can be implemented with fewer computational resources or lower standard platforms.

DCL on CIFAR and STL10 For STL10, CIFAR10, and CIFAR100, we implement our DCL with ResNet-18 as encoder backbone. In Table 1, it is observed that DCL also demonstrates its effectiveness on small-scale benchmarks. In summary, DCL outperforms its baseline by 3.1% (CIFAR10) and 2.8% (CIFAR100) and keeps the performance relatively steady under batch size 256. The kNN accuracy of the SimCLR baseline on STL10 is also improved by 3.9%.

Further experiments are conducted based on the ResNet-50 backbone and large learning epochs (i.e., 500 epochs). The DCL model with kNN eval, batch size 32, and 500 epochs of training could reach 86.1% compared to 82.2%. For the following experiments in Table 1, we show DCL ResNet-50 performance on CIFAR10 and CIFAR100. In these comparisons, we vary the batch size to show the effectiveness of DCL.

Decoupled Objective with Re-Weighting DCLW We only replace L_{DC} with L_{DCW} with no possible advantage from additional tricks. That is, both our approach and the baselines apply the same training instruction of the OpenSelfSup benchmark for fairness. Note that we empirically choose $\sigma = 0.5$ in the experiments. Results in Table 2 indicates that, DCLW achieves extra 5.1% (ImageNet-1K), 3.5% (ImageNet-100) gains compared to the baseline. For CIFAR data, extra 3.7% (CIFAR10), 3.1% are gained from the addition of DCLW. It is worth to note that, trained with 200 epochs, our DCLW reaches 66.9% with batch size 256, surpassing the SimCLR baseline: 66.2% with batch size 8192.

Table 3: kNN top-1 accuracy (%) comparison of SSL approaches on small-scale benchmarks: CIFAR10, CIFAR100, and STL10. Results show that DCL consistently improves its SimCLR baseline. With multi-cropping (Caron et al., 2020), our DCLW reaches competitive performance within other contrastive learning approaches (Chen et al., 2020a; He et al., 2020; Wu et al., 2018; Ye et al., 2019; Dosovitskiy et al., 2015).

kNN (top-1)	SimCLR	MoCo	MoCo + CLD	NPID	NPID + CLD	Inv. Spread	Exemplar	DCL	DCLW w/ mcrop
CIFAR10	81.4	82.1	87.5	80.8	86.7	83.6	76.5	84.1	**87.8**
CIFAR100	52.0	53.1	58.1	51.6	57.5	N/A	N/A	54.9	**58.8**
STL10	77.3	80.8	**84.3**	79.1	83.6	81.6	79.3	81.2	84.1

Table 4: Improve the DCL model performance on ImageNet-1K with better hyperparameters, temperature and learning rate and stronger augmentation.

ImageNet-1K (batch size = 256; epoch = 200)	Linear Top-1 Accuracy (%)
DCL	65.9
+ optimal (τ, l_r) = (0.2, 0.07)	67.8 (+1.9)
+ BYOL augmentation	68.2 (+0.4)

Table 5: ImageNet-1K top-1 accuracy (%) on SimCLR and MoCo v2 with/without DCL under few training epochs. We further list results under 200 epochs for clear comparison. With DCL, the performance of SimCLR trained under 100 epochs nearly reaches its performance under 200 epochs. The MoCo v2 with DCL also reaches higher accuracy than the baseline under 100 epochs.

	SimCLR	SimCLR w/ DCL	MoCo v2	MoCo v2 w/ DCL
100 epoch	57.5	64.6	63.6	64.4
200 epoch	61.8	65.9	67.5	67.7

4.3 Small-scale benchmark results: STL10, CIFAR10, and CIFAR100

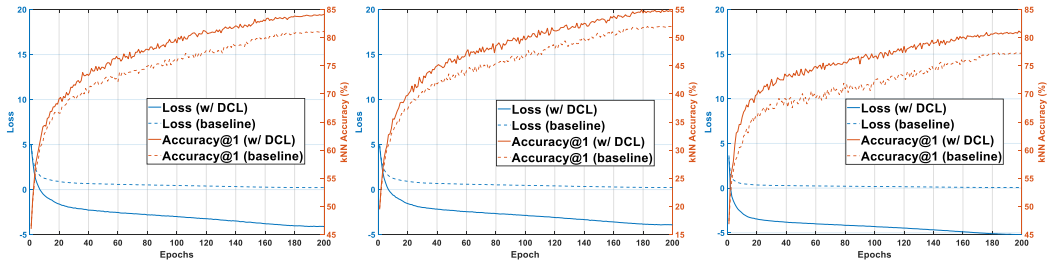
For STL10, CIFAR10, and CIFAR100, we implement our DCL with ResNet-18 (He et al., 2016) as encoder backbone by following the small-scale benchmark of CLD (Wang et al., 2021b). All the models are trained for 200 epochs with 256 batch sizes and evaluate by using kNN accuracies ($k = 200$). Results in Table 3 indicate that, our DCLW with multi-cropping (Caron et al., 2020) consistently outperforms the state-of-the-art baselines on CIFAR10, STL10, and CIFAR100. Our DCL also demonstrates its capability while comparing against other baselines. More analysis of large-scale benchmarks can be found in Appendix.

4.4 Ablations

We perform extensive ablations on the hyperparameters of our DCL and DCLW on both ImageNet data and other small-scale data, i.e., CIFAR10, CIFAR100, and STL10. By seeking better configurations empirically, we see that our approach gives consistent gains over the standard SimCLR baseline. In other ablations, we see that our DCL achieves more gains over both SimCLR and MoCo v2, i.e., contrastive learning baselines, also when training for 100 epochs only.

Ablations of DCL on ImageNet In Table 4, we have slightly improved the DCL model performance on ImageNet-1K: 1) better hyperparameters, temperature τ and learning rate ; 2) stronger augmentation (e.g., BYOL). We conduct an empirical hyperparameter search with batch size 256 and 200 epochs to obtain a stronger baseline. This improves DCL from 65.9% to 67.8% top-1 accuracy on ImageNet-1K. We further adopt the BYOL augmentation policy and improve our DCL from 67.8% to 68.2% top-1 accuracy on ImageNet-1K.

Few learning epochs Our DCL is inspired by the traditional contrastive learning framework, which needs a large batch size, long learning epochs to achieve higher performance. The previous



(a) CIFAR10 (b) CIFAR100 (c) STL10

Figure 4: Comparisons between DCL and SimCLR baseline on (a) CIFAR10, (b) CIFAR100, and (c) STL10 data. During the SSL pre-training, DCL speeds up the model convergence and provides better performance than the baseline on CIFAR and STL10 data.

Table 6: The ablation study of various temperature τ on the CIFAR10.

Temperature τ	0.07	0.1	0.2	0.3	0.4	0.5	Standard deviation
SimCLR	83.6	87.5	89.5	89.2	88.7	89.1	2.04
DCL	88.3	89.4	90.8	89.9	89.6	90.3	**0.78**

state-of-the-art, SimCLR, heavily relies on large quantities of learning epochs to obtain high top-1 accuracy. (e.g., 69.3% with up to 1000 epochs). The purpose of our DCL is to achieve higher learning efficiency with few learning epochs. We demonstrate the effectiveness of DCL in contrastive learning frameworks SimCLR and MoCo v2 (Chen et al., 2020b). We choose the batch size of 256 (queue of 65536) as the baseline and train the model with only 100 epochs instead of the normal number of 200. We make sure other parameter settings are the same for a fair comparison. Table 5 shows the result on ImageNet-1K using linear evaluation. With DCL, SimCLR can achieve 64.6% top-1 accuracy with only 100 epochs compared to SimCLR baseline: 57.5%; MoCo v2 with DCL reaches 64.4% compared to MoCo v2 baseline: 63.6% with 100 epochs pre-training.

We further demonstrate that, with DCL, learning representation becomes faster during the early stage of training. The reason is that DCL successfully solves the decoupled issue between positive and negative pairs. Figure 4 on (a) CIFAR10, CIFAR100, and STL10, show that our DCL improves the speed of convergence and reaches higher performance than the baseline on CIFAR and STL10 data.

Analysis of temperature τ In Table 6, we further provide extensive analysis on temperature τ in the loss to support that the DCL method is not sensitive to hyperparameters compared against the baselines. In the following, show the temperature τ search on both DCL and SimCLR baselines on CIFAR10 data. Specifically, we pretrain the network with temperature τ in $\{0.07, 0.1, 0.2, 0.3, 0.4, 0.5\}$ and report results with kNN eval, batch size 512, and 500 epochs. As shown in Table 6, compared to the SimCLR baseline, DCL is less sensitive to hyperparameters, e.g., temperature τ.

5 Conclusion

In this paper, we identify the negative-positive-coupling (NPC) effect in the wide used InfoNCE loss, making the SSL task significantly easier to solve with smaller batch size. By removing the NPC effect, we reach a new objective function, *decoupled contrastive learning* (DCL). The proposed DCL loss function requires minimal modification to the SimCLR baseline and provides efficient, reliable, and nontrivial performance improvement on various benchmarks. Given the conceptual simplicity of DCL and that it requires neither momentum encoding, large batch sizes, or long epochs to reach competitive performance, we wish that DCL can serve as a strong baseline for the contrastive-based SSL methods. Further, an important lesson we learn from the DCL loss is that a more efficient SSL task shall maintain its complexity when the batch size becomes smaller.

REPRODUCIBILITY STATEMENT

Following the Equation 5 in the main paper, the proposed method is indeed straightforward to reproduce as the essential thing to do is removing a single term, which requires a few lines of code. Furthermore, it is unnecessary to tune the parameters to see an initial performance gain since the method is not sensitive to hyperparameters mentioned in Section 4.4. For theoretical results, explanations of assumptions and a complete proof of the claims can be found in Section 3, Appendix A.1, and Appendix A.2. For experiments, we use several public benchmarks, such as ImageNet, CIFAR, and STL10. A complete description of the data processing steps and the implementation details of our method can be found in Section 4.1 and Appendix A.4.

REFERENCES

Alexei Baevski, Yuhao Zhou, Abdelrahman Mohamed, and Michael Auli. wav2vec 2.0: A framework for self-supervised learning of speech representations. In Hugo Larochelle, Marc'Aurelio Ranzato, Raia Hadsell, Maria-Florina Balcan, and Hsuan-Tien Lin (eds.), *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020.

Adrien Bardes, Jean Ponce, and Yann LeCun. Vicreg: Variance-invariance-covariance regularization for self-supervised learning. *CoRR*, abs/2105.04906, 2021.

Mohamed Ishmael Belghazi, Aristide Baratin, Sai Rajeswar, Sherjil Ozair, Yoshua Bengio, R. Devon Hjelm, and Aaron C. Courville. Mutual information neural estimation. In Jennifer G. Dy and Andreas Krause (eds.), *Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018*, volume 80 of *Proceedings of Machine Learning Research*, pp. 530–539. PMLR, 2018.

Mathilde Caron, Piotr Bojanowski, Armand Joulin, and Matthijs Douze. Deep clustering for unsupervised learning of visual features. In Vittorio Ferrari, Martial Hebert, Cristian Sminchisescu, and Yair Weiss (eds.), *Computer Vision - ECCV 2018 - 15th European Conference, Munich, Germany, September 8-14, 2018, Proceedings, Part XIV*, volume 11218 of *Lecture Notes in Computer Science*, pp. 139–156. Springer, 2018.

Mathilde Caron, Ishan Misra, Julien Mairal, Priya Goyal, Piotr Bojanowski, and Armand Joulin. Unsupervised learning of visual features by contrasting cluster assignments. In Hugo Larochelle, Marc'Aurelio Ranzato, Raia Hadsell, Maria-Florina Balcan, and Hsuan-Tien Lin (eds.), *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020.

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey E. Hinton. A simple framework for contrastive learning of visual representations. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*, volume 119 of *Proceedings of Machine Learning Research*, pp. 1597–1607. PMLR, 2020a.

Wei Chen, Tie-Yan Liu, Yanyan Lan, Zhiming Ma, and Hang Li. Ranking measures and loss functions in learning to rank. In Yoshua Bengio, Dale Schuurmans, John D. Lafferty, Christopher K. I. Williams, and Aron Culotta (eds.), *Advances in Neural Information Processing Systems 22: 23rd Annual Conference on Neural Information Processing Systems 2009. Proceedings of a meeting held 7-10 December 2009, Vancouver, British Columbia, Canada*, pp. 315–323. Curran Associates, Inc., 2009.

Xinlei Chen and Kaiming He. Exploring simple siamese representation learning. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2021, virtual, June 19-25, 2021*, pp. 15750–15758. Computer Vision Foundation / IEEE, 2021.

Xinlei Chen, Haoqi Fan, Ross B. Girshick, and Kaiming He. Improved baselines with momentum contrastive learning. *CoRR*, abs/2003.04297, 2020b.

Adam Coates, Andrew Ng, and Honglak Lee. An analysis of single-layer networks in unsupervised feature learning. In *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, pp. 215–223. JMLR Workshop and Conference Proceedings, 2011.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pp. 248–255. Ieee, 2009.

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*, pp. 1422–1430, 2015.

Alexey Dosovitskiy, Philipp Fischer, Jost Tobias Springenberg, Martin Riedmiller, and Thomas Brox. Discriminative unsupervised feature learning with exemplar convolutional neural networks. *IEEE transactions on pattern analysis and machine intelligence*, 38(9):1734–1747, 2015.

Debidatta Dwibedi, Yusuf Aytar, Jonathan Tompson, Pierre Sermanet, and Andrew Zisserman. With a little help from my friends: Nearest-neighbor contrastive learning of visual representations. *CoRR*, abs/2104.14548, 2021.

Spyros Gidaris, Praveer Singh, and Nikos Komodakis. Unsupervised representation learning by predicting image rotations. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net, 2018.

Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in neural information processing systems*, pp. 2672–2680, 2014.

Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre H. Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Ávila Pires, Zhaohan Guo, Mohammad Gheshlaghi Azar, Bilal Piot, Koray Kavukcuoglu, Rémi Munos, and Michal Valko. Bootstrap your own latent - A new approach to self-supervised learning. In Hugo Larochelle, Marc'Aurelio Ranzato, Raia Hadsell, Maria-Florina Balcan, and Hsuan-Tien Lin (eds.), *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020.

Raia Hadsell, Sumit Chopra, and Yann LeCun. Dimensionality reduction by learning an invariant mapping. In *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2006), 17-22 June 2006, New York, NY, USA*, pp. 1735–1742. IEEE Computer Society, 2006.

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

Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 9729–9738, 2020.

R. Devon Hjelm, Alex Fedorov, Samuel Lavoie-Marchildon, Karan Grewal, Philip Bachman, Adam Trischler, and Yoshua Bengio. Learning deep representations by mutual information estimation and maximization. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net, 2019.

Prannay Khosla, Piotr Teterwak, Chen Wang, Aaron Sarna, Yonglong Tian, Phillip Isola, Aaron Maschinot, Ce Liu, and Dilip Krishnan. Supervised contrastive learning. In Hugo Larochelle, Marc'Aurelio Ranzato, Raia Hadsell, Maria-Florina Balcan, and Hsuan-Tien Lin (eds.), *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020.

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

Loren Lugosch, Mirco Ravanelli, Patrick Ignoto, Vikrant Singh Tomar, and Yoshua Bengio. Speech model pre-training for end-to-end spoken language understanding. In Gernot Kubin and Zdravko Kacic (eds.), *Interspeech 2019, 20th Annual Conference of the International Speech Communication Association, Graz, Austria, 15-19 September 2019*, pp. 814–818. ISCA, 2019.

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*, pp. 6707–6717, 2020.

Arsha Nagrani, Joon Son Chung, Weidi Xie, and Andrew Zisserman. Voxceleb: Large-scale speaker verification in the wild. *Comput. Speech Lang.*, 60, 2020.

Mehdi Noroozi and Paolo Favaro. Unsupervised learning of visual representations by solving jigsaw puzzles. In Bastian Leibe, Jiri Matas, Nicu Sebe, and Max Welling (eds.), *Computer Vision - ECCV 2016 - 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part VI*, volume 9910 of *Lecture Notes in Computer Science*, pp. 69–84. Springer, 2016.

Sherjil Ozair, Corey Lynch, Yoshua Bengio, Aäron van den Oord, Sergey Levine, and Pierre Sermanet. Wasserstein dependency measure for representation learning. In Hanna M. Wallach, Hugo Larochelle, Alina Beygelzimer, Florence d'Alché-Buc, Emily B. Fox, and Roman Garnett (eds.), *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pp. 15578–15588, 2019.

Alec Radford, Luke Metz, and Soumith Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. In Yoshua Bengio and Yann LeCun (eds.), *4th International Conference on Learning Representations, ICLR 2016, San Juan, Puerto Rico, May 2-4, 2016, Conference Track Proceedings*, 2016.

Hao Ren. A pytorch implementation of simclr. https://github.com/leftthomas/SimCLR, 2020.

Igor Susmelj, Matthias Heller, Philipp Wirth, Jeremy Prescott, Malte Ebner, and et al. Lightly. *GitHub. Note: https://github.com/lightly-ai/lightly*, 2020.

Yonglong Tian, Dilip Krishnan, and Phillip Isola. Contrastive multiview coding. In Andrea Vedaldi, Horst Bischof, Thomas Brox, and Jan-Michael Frahm (eds.), *Computer Vision - ECCV 2020 - 16th European Conference, Glasgow, UK, August 23-28, 2020, Proceedings, Part XI*, volume 12356 of *Lecture Notes in Computer Science*, pp. 776–794. Springer, 2020a.

Yonglong Tian, Chen Sun, Ben Poole, Dilip Krishnan, Cordelia Schmid, and Phillip Isola. What makes for good views for contrastive learning? In Hugo Larochelle, Marc'Aurelio Ranzato, Raia Hadsell, Maria-Florina Balcan, and Hsuan-Tien Lin (eds.), *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020b.

Yao-Hung Hubert Tsai, Martin Q. Ma, Muqiao Yang, Han Zhao, Louis-Philippe Morency, and Ruslan Salakhutdinov. Self-supervised representation learning with relative predictive coding. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net, 2021.

Aäron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *CoRR*, abs/1807.03748, 2018.

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

Xudong Wang, Ziwei Liu, and Stella X. Yu. Unsupervised feature learning by cross-level instance-group discrimination. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2021, virtual, June 19-25, 2021*, pp. 12586–12595. Computer Vision Foundation / IEEE, 2021a.

Xudong Wang, Ziwei Liu, and Stella X. Yu. Unsupervised feature learning by cross-level instance-group discrimination. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2021, virtual, June 19-25, 2021*, pp. 12586–12595. Computer Vision Foundation / IEEE, 2021b.

Zhirong Wu, Yuanjun Xiong, Stella X Yu, and Dahua Lin. Unsupervised feature learning via non-parametric instance discrimination. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3733–3742, 2018.

Mang Ye, Xu Zhang, Pong C Yuen, and Shih-Fu Chang. Unsupervised embedding learning via invariant and spreading instance feature. In *Proceedings of the IEEE Conference on computer vision and pattern recognition*, pp. 6210–6219, 2019.

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

Xiaohang Zhan, Jiahao Xie, Ziwei Liu, Dahua Lin, and Chen Change Loy. OpenSelfSup: Open mmlab self-supervised learning toolbox and benchmark. https://github.com/open-mmlab/openselfsup, 2020.

Richard Zhang, Phillip Isola, and Alexei A. Efros. Colorful image colorization. In Bastian Leibe, Jiri Matas, Nicu Sebe, and Max Welling (eds.), *Computer Vision - ECCV 2016 - 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part III*, volume 9907 of *Lecture Notes in Computer Science*, pp. 649–666. Springer, 2016.

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

A APPENDIX

A.1 PROOF OF PROPOSITION 1

Proposition 1. There exists a negative-positive coupling (NPC) multiplier $q_{B,i}^{(1)}$ in the gradient of $L_i^{(1)}$:

$$\begin{cases} -\nabla_{\mathbf{z}_i^{(1)}} L_i^{(1)} = \frac{q_{B,i}^{(1)}}{\tau}\left[\mathbf{z}_i^{(2)} - \sum_{l\in\{1,2\},j\in[\![1,N]\!],j\neq i} \frac{\exp\langle \mathbf{z}_i^{(1)},\mathbf{z}_j^{(l)}\rangle/\tau}{\sum_{q\in\{1,2\},j\in[\![1,N]\!],j\neq i}\exp(\langle \mathbf{z}_i^{(1)},\mathbf{z}_j^{(q)}\rangle/\tau)}\cdot\mathbf{z}_j^{(l)}\right] \\ -\nabla_{\mathbf{z}_i^{(2)}} L_i^{(1)} = \frac{q_{B,i}^{(1)}}{\tau}\cdot\mathbf{z}_i^{(1)} \\ -\nabla_{\mathbf{z}_j^{(l)}} L_i^{(1)} = -\frac{q_{B,i}^{(1)}}{\tau}\frac{\exp\langle \mathbf{z}_i^{(1)},\mathbf{z}_j^{(l)}\rangle/\tau}{\sum_{q\in\{1,2\},j\in[\![1,N]\!],j\neq i}\exp(\langle \mathbf{z}_i^{(1)},\mathbf{z}_j^{(q)}\rangle/\tau)}\cdot\mathbf{z}_i^{(1)} \end{cases}$$

where the NPC multiplier $q_{B,i}^{(1)}$ is:

$$q_{B,i}^{(1)} = 1 - \frac{\exp(\langle \mathbf{z}_i^{(1)},\mathbf{z}_i^{(2)}\rangle/\tau)}{\sum_{q\in\{1,2\},j\in[\![1,N]\!],j\neq i}\exp(\langle \mathbf{z}_i^{(1)},\mathbf{z}_j^{(q)}\rangle/\tau)}$$

Due to the symmetry, a similar NPC multiplier $q_{B,i}^{(k)}$ exists in the gradient of $L_i^{(k)}, k\in\{1,2\}, i\in[\![1,N]\!]$.

Proof.

$$\begin{aligned} -\nabla_{\mathbf{z}_i^{(1)}} L_i^{(1)} &= \frac{\mathbf{z}_i}{\tau} - \frac{1}{Y}\cdot\exp(\langle \mathbf{z}_i^{(1)},\mathbf{z}_i^{(2)}\rangle/\tau)\cdot\frac{\mathbf{z}_i^{(2)}}{\tau} - \frac{1}{Y}\cdot\sum_{q\in\{1,2\},j\in[\![1,N]\!],j\neq i}\exp(\langle \mathbf{z}_i^{(1)},\mathbf{z}_j^{(q)}\rangle/\tau)\frac{\mathbf{z}_j^{(q)}}{\tau} \\ &= (1-\frac{1}{Y}\cdot\exp(\langle \mathbf{z}_i^{(1)},\mathbf{z}_i^{(2)}\rangle/\tau))\frac{\mathbf{z}_i^{(2)}}{\tau} - \frac{1}{Y}\cdot\sum_{q\in\{1,2\},j\in[\![1,N]\!],j\neq i}\exp(\langle \mathbf{z}_i^{(1)},\mathbf{z}_j^{(q)}\rangle/\tau)\frac{\mathbf{z}_j^{(q)}}{\tau} \\ &= \frac{1}{\tau}(1-\frac{1}{Y}\cdot\exp(\langle \mathbf{z}_i^{(1)},\mathbf{z}_i^{(2)}\rangle/\tau))\left[\mathbf{z}_i^{(2)} - \sum_{q\in\{1,2\},j\in[\![1,N]\!],j\neq i}\frac{\exp(\langle \mathbf{z}_i^{(1)},\mathbf{z}_j^{(q)}\rangle/\tau)}{U}\cdot\mathbf{z}_j^{(q)}\right] \\ &= \frac{q_{B,i}^{(1)}}{\tau}\left[\mathbf{z}_i^{(2)} - \sum_{q\in\{1,2\},j\in[\![1,N]\!],j\neq i}\frac{\exp(\langle \mathbf{z}_i^{(1)},\mathbf{z}_j^{(q)}\rangle/\tau)}{U}\cdot\mathbf{z}_j^{(q)}\right] \end{aligned}$$

where $Y = \exp(\langle \mathbf{z}_i^{(1)},\mathbf{z}_i^{(2)}\rangle/\tau) + \sum_{q\in\{1,2\},j\in[\![1,N]\!],j\neq i}\exp(\langle \mathbf{z}_i^{(1)},\mathbf{z}_j^{(q)}\rangle/\tau)$, $U = \sum_{q\in\{1,2\},j\in[\![1,N]\!],j\neq i}\exp(\langle \mathbf{z}_i^{(1)},\mathbf{z}_j^{(q)}\rangle/\tau)$.

$$\begin{aligned} -\nabla_{\mathbf{z}_i^{(2)}} L_i^{(1)} &= \frac{1}{\tau}\mathbf{z}_i^{(1)} - \frac{1}{Y}\exp(\langle \mathbf{z}_i^{(1)},\mathbf{z}_i^{(2)}\rangle/\tau)\cdot\frac{\mathbf{z}_i^{(1)}}{\tau} \\ &= \frac{q_{B,i}^{(1)}}{\tau}\cdot\mathbf{z}_i^{(1)} \end{aligned}$$

$$\begin{aligned} -\nabla_{\mathbf{z}_j^{(l)}} L_i^{(1)} &= \frac{1}{Y}\exp(\langle \mathbf{z}_i^{(1)},\mathbf{z}_j^{(q)}\rangle/\tau)\cdot\frac{\mathbf{z}_i^{(1)}}{\tau} \\ &= \frac{q_{B,i}^{(1)}}{\tau}\cdot\frac{\exp(\langle \mathbf{z}_i^{(1)},\mathbf{z}_j^{(q)}\rangle/\tau)}{U}\mathbf{z}_i^{(1)} \end{aligned}$$

□

Table 7: ImageNet-1K top-1 accuracies (%) of linear classifiers trained on representations of different SSL methods with ResNet-50 backbone. The results in the lower section are the same methods with a larger experiment setting.

Method	Param. (M)	Batch size	Epochs	Top-1 (%)
Relative-Loc. (Doersch et al., 2015)	24	256	200	49.3
Rotation-Pred. (Gidaris et al., 2018)	24	256	200	55.0
DeepCluster (Caron et al., 2018)	24	256	200	57.7
NPID (Wu et al., 2018)	24	256	200	56.5
Local Agg. (Zhuang et al., 2019)	24	256	200	58.8
MoCo (He et al., 2020)	24	256	200	60.6
SimCLR (Chen et al., 2020a)	28	256	200	61.8
CMC (Tian et al., 2020a)	47	256	280	64.1
MoCo v2 (Chen et al., 2020b)	28	256	200	67.5
SwAV (Caron et al., 2020)	28	4096	200	69.1
SimSiam (Chen & He, 2021)	28	256	200	70.0
InfoMin (Tian et al., 2020b)	28	256	200	70.1
BYOL (Grill et al., 2020)	28	4096	200	70.6
Hypersphere (Wang & Isola, 2020)	28	256	200	68.2
DCL	28	256	200	67.8
DCL+BYOL aug.	28	256	200	68.2
PIRL (Misra & Maaten, 2020)	24	256	800	63.6
SimCLR (Chen et al., 2020a)	28	4096	1000	69.3
MoCo v2 (Chen et al., 2020b)	28	256	800	71.1
SwAV (Caron et al., 2020)	28	4096	400	70.7
SimSiam (Chen & He, 2021)	28	256	800	71.3
DCL	28	256	400	69.5

A.2 Proof of Proposition 2

Proposition 2. Removing the positive pair from the denominator of Equation 2 leads to a decoupled contrastive learning loss. If we remove the NPC multiplier $q_{B,i}^{(k)}$ from Equation 2, we reach a decoupled contrastive learning loss $L_{DC} = \sum_{k\in\{1,2\},i\in[\![1,N]\!]} L_{DC,i}^{(k)}$, where $L_{DC,i}^{(k)}$ is:

$$L_{DC,i}^{(k)} = -\log \frac{\exp(\langle \mathbf{z}_i^{(1)}, \mathbf{z}_i^{(2)} \rangle / \tau)}{\cancel{\exp(\langle \mathbf{z}_i^{(1)}, \mathbf{z}_i^{(2)} \rangle / \tau)} + \sum_{l\in\{1,2\},j\in[\![1,N]\!],j\neq i} \exp(\langle \mathbf{z}_i^{(k)}, \mathbf{z}_j^{(l)} \rangle / \tau)}$$

$$= -\langle \mathbf{z}_i^{(1)}, \mathbf{z}_i^{(2)} \rangle / \tau + \log \sum_{l\in\{1,2\},j\in[\![1,N]\!],j\neq i} \exp(\langle \mathbf{z}_i^{(k)}, \mathbf{z}_j^{(l)} \rangle / \tau)$$

Proof. By removing the positive term in the denominator of Equation 4, we can repeat the procedure in the proof of Proposition 1 and see that the coupling term disappears.

□

A.3 Linear classification on ImageNet-1K

Top-1 accuracies of linear evaluation in Table 7 shows that, we compare with the state-of-the-art SSL approaches on ImageNet-1K. For fairness, we list each individual approach's batch size and learning epoch, which are shown in the original paper. During pre-training, our DCL is based on a ResNet-50 backbone, with two views with the size 224×224. Our DCL relies on its simplicity to reach competitive performance without relatively huge batch sizes or other pre-training schemes, i.e., momentum encoder, clustering, and prediction head. We report both 200-epoch and 400-epoch versions of our DCL. It achieves 69.5% under the batch size of 256 and 400-epoch pre-training, which is better than SimCLR (Chen et al., 2020a) in their optimal case, i.e., batch size of 4096 and 1000-epoch. Note that SwAV (Caron et al., 2018), BYOL (Grill et al., 2020), SimCLR, and PIRL (Misra &

Maaten, 2020) need huge batch size of 4096, and SwAV further applies multi-cropping as generating extra views to reach optimal performance.

A.4 Implementation details

Default DCL augmentations. We follow the settings of SimCLR to set up the data augmentations. We use $RandomResizedCrop$ with scale in [0.08, 1.0] and follow by $RandomHorizontalFlip$. Then, $ColorJittering$ with strength in [0.8, 0.8, 0.8, 0.2] with probability of 0.8, and $RandomGrayscale$ with probability of 0.2. $GaussianBlur$ includes Gaussian kernel with standard deviation in [0.1, 2.0].

Strong DCL augmentations. We follow the augmentation pipeline of BYOL to replace default DCL augmentation in ablations. Table 4 demonstrates that the ImageNet-1K top-1 performance is increased from 67.8% to 68.2% by applying BYOL's augmentations.

Linear evaluation. Following the OpenSelfSup benchmark (Zhan et al., 2020), we first train the linear classifier with batch size 256 for 100 epochs. We use the SGD optimizer with momentum = 0.9, and weight decay = 0. The base lr is set to 30.0 and decay by 0.1 at epoch [60, 80]. We further demonstrate the linear evaluation protocol of SimSiam (Chen & He, 2021), which raises the batch size to 4096 for 90 epochs. The optimizer is switched to LARS optimizer with base $lr = 1.2$ and cosine decay schedule. The momentum and weight decay have remained unchanged. We found the second one slightly improves the performance.

A.5 Relation to alignment and uniformity

In this section, we provide a thorough discussion of the connection and difference between DCL and Hypersphere (Wang & Isola, 2020), which does not have negative-positive coupling either. However, there is a critical difference between DCL and Hypersphere, and the difference is that the order of the expectation and exponential is swapped. Let us assume the latent embedding vectors z are normalized for analytical convenience. When z_i, z_j are normalized, $\exp(\langle \mathbf{z}_i^{(k)}, \mathbf{z}_i^{(l)} \rangle / \tau)$ and $\exp(-||\mathbf{z}_i^{(k)} - \mathbf{z}_i^{(l)}||^2/\tau)$ are the same, except for a trivial scale difference. Thus we can write L_{DCL} and $L_{align-uni}$ in a similar fashion:

$$L_{DCL} = L_{DCL,pos} + L_{DCL,neg}, L_{DCL,neg} = \sum_i \log(\sum_{j \neq i} \exp(\langle \mathbf{z}_i^{(k)}, \mathbf{z}_i^{(l)} \rangle / \tau))$$

$$L_{align-uni} = L_{align} + L_{uniform}, L_{uniform} = \log(\sum_i \sum_{j \neq i} \exp(\langle \mathbf{z}_i^{(k)}, \mathbf{z}_i^{(l)} \rangle / \tau))$$

With the right weight factor, L_{align} can be made exactly the same as $L_{DCL,pos}$. So let's focus on $L_{DCL,neg}$ and $L_{uniform}$:

$$L_{DCL,neg} = \sum_i \log(\sum_{j \neq i} \exp(\langle \mathbf{z}_i^{(k)}, \mathbf{z}_i^{(l)} \rangle / \tau))$$

$$L_{uniform} = \log(\sum_i \sum_{j \neq i} \exp(\langle \mathbf{z}_i^{(k)}, \mathbf{z}_i^{(l)} \rangle / \tau))$$

Similar to our earlier analysis in the manuscript, the latter $L_{uniform}$ introduces a negative-negative coupling between the negative samples of different positive samples. If two negative samples of z_i are close to each other, the gradient for z_i would also be attenuated. This behaves similarly to the negative-positive coupling. That being said, while Hypersphere does not have a negative-positive coupling, it has a similarly problematic negative-negative coupling. Next, we provide a

Table 8: STL10 comparisons Hypersphere and DCL under the same experiment setting.

STL10	fc7+Linear	fc7+5-NN	Output + Linear	Output + 5-NN
Hypersphere	83.2	76.2	80.1	79.2
DCL	**84.4 (+1.2)**	**77.3 (+1.1)**	**81.5 (+1.4)**	**80.5 (+1.3)**

Table 9: ImageNet-100 comparisons of Hypersphere and DCL under the same setting (MoCo).

ImageNet-100	Epoch	Memory Queue Size	Linear Top-1 Accuracy (%)
Hypersphere	240	16384	75.6
DCL	240	16384	**76.8 (+1.2)**

Table 10: ImageNet-100 comparisons of Hypersphere and DCL under the same setting (MoCo v2) except for memory queue size.

ImageNet-100	Epoch	Memory Queue Size	Linear Top-1 Accuracy (%)
Hypersphere	200	16384	77.7
DCL	200	8192	**80.5 (+2.7)**

Table 11: ImageNet-1K comparisons of and DCL under the best setting. In this experiment both of the methods used their optimized hyperparameters.

ImageNet-1K	Epoch	Batch Size	Linear Top-1 Accuracy (%)
Hypersphere	200	256 (Memory queue = 16384)	67.7
DCL	200	256	**68.2 (+0.5)**

Table 12: STL10 comparisons of Hypersphere and DCL under different batch sizes.

Batch Size	32	64	128	256	768
Hypersphere	78.9	81.0	81.9	82.6	83.2
DCL	**81.0 (+2.1)**	**82.9 (+1.9)**	**83.7 (+1.8)**	**84.2 (+1.6)**	**84.4 (+1.2)**

comprehensive empirical comparison. The empirical experiments match our analytical prediction: DCL outperforms Hypersphere with a more considerable margin under a smaller batch size.

The comparisons of DCL to Hypersphere are evaluated on STL10, ImageNet-100, ImageNet-1K under various settings. For STL10 data, we implement DCL based on the official code of Hypersphere. The encoder and the hyperparameters are the same as Hypersphere, which has not been optimized for DCL in any way. We have found that Hypersphereha s did a pretty thorough hyperparameter search. We believe the default hyperparameters are relatively optimized for Hypersphere.

In Table 8, DCL reaches 84.4% (fc7+Linear) compared to 83.2% (fc7+Linear) reported in Hypersphere on STL10. In Table 9 and Table 10, our DCL achieves better performance than Hypersphere under the same setting (MoCo & MoCo v2) on ImageNet-100 data. Our DCL further shows strong results compared against Hypersphere on ImageNet-1K in Table 11. We also provide the STL10 comparisons of our DCL and Hypersphere under different batch sizes in Table 12. The experiment shows the advantage of DCL becomes larger with smaller batch size. Please note that we did not tune the parameters for DCL at all. This should be a more than fair comparison.

In every single one of the experiments, DCL outperforms Hypersphere. We hope these results show the unique value of DCL compared to Hypersphere.

A.6 LIMITATIONS OF THE PROPOSED DCL

We summarize two limitations of the proposed DCL method. First, the performance of DCL appears to have less gain compared to the SimCLR baseline when the batch size is large. According to Figure 1 and the theoretical analysis, the reason is that the NPC multiplier $q_B \to 0$ when the batch size is large (e.g., 1024). As we have shown in the analysis, the baseline SimCLR loss converges to the DCL loss as the batch size approaches infinity. With 400 training epochs, the ImageNet-1K top-1 accuracy slightly increases from 69.5% to 69.9% when the batch size is increased from 256 to 1024. Please refer to Table 13.

Second, the scenario of DCL focuses on contrastive learning-based methods, where we decouple the positive and negative terms to achieve better learning efficiency. However, non-negative methods, e.g., VICReg (Bardes et al., 2021), does not rely on negative samples. While the non-contrastive methods have achieved better performance on large-scale benchmarks like ImageNet, competitive contrastive methods, like NNCLR (Dwibedi et al., 2021), have recently been proposed. The DCL method can be potentially combined with the method NNCLR to achieve further improvement. The SOTA SSL speech models, e.g., wav2vec 2.0 (Baevski et al., 2020) still uses contrastive loss in the objective. In Table 14, we show the effectiveness of DCL with wav2vec 2.0 (Baevski et al., 2020). We replace the contrastive loss with the DCL loss and train a wav2vec 2.0 base model (i.e., 7-Conv + 24-Transformer) from scratch.[1] After the training, we evaluate the representation on two downstream tasks, speaker identification and intent classification. Table 14 shows the representation improvement.

After all, there is not a consensus that non-contrastive methods would lead to the SOTA universally on different datasets. In fact, in Table 15, we can use CIFAR-10 as an example to show that DCL achieves competitive results compared to BYOL, SimSiam, and Barlow Twins.

Table 13: Results of DCL with large batch size and learning epochs.

ImageNet-1K (ResNet-50)	Batch Size	Epoch	Top-1 Accuracy (%)
DCL†	256	200	67.8
DCL†	256	400	69.5 (+1.7)
DCL	1024	400	69.9 (+0.4)

† The values are taken from Table 7

Table 14: Results of DCL on wav2vec 2.0 be evaluated on two downstream tasks.

Downstream task (Accuracy)	Speaker Identification† (%)	Intent Classification‡ (%)
wav2vec 2.0 Base Baseline	74.9	92.3
wav2vec 2.0 Base w/ (DCL)	**75.2**	**92.5**

† In the downstream training process, the pre-trained representation first mean-pool and forward a fully a connected layer with cross-entropy loss on the VoxCeleb1 (Nagrani et al., 2020).

‡ In the downstream training process, the pre-trained representation first mean-pool and forward a fully a connected layer with cross-entropy loss on Fluent Speech Commands (Lugosch et al., 2019).

[1]The experiment is downscaled to 8 V100 GPUs rather than 64.

Table 15: CIFAR-10 as an example to show that DCL achieves competitive results compared to BYOL, SimSiam, and Barlow Twins.

CIFAR-10 (ResNet-18)	Batch Size	Epoch	kNN Accuracy (%)
BYOL†	128	200	**85**
SimSiam†	128	200	73
Barlow Twins†	128	200	84
DCL	128	200	84
BYOL†	512	200	**84**
SimSiam†	512	200	81
Barlow Twins†	512	200	78
DCL	512	200	**84**

† The method is implemented by (Susmelj et al., 2020).