A Appendix

A.1 Proof of proposition 1

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

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:

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

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

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

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

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

where we can easily see that $\sum_{q \in \{1,2\}, j \in [\![1,N]\!], j \neq i} \frac{\exp(\langle \mathbf{z}_i^{(1)}, \mathbf{z}_j^{(q)} \rangle / \tau)}{U_{i,1}} = 1$.

A.2 Proof of proposition 2

Proposition 2. Removing the positive pair from the denominator of Equation 3 leads to a decoupled contrastive learning loss. If we remove the NPC multiplier $q_{B,i}^{(k)}$ from Equation 3, 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:

$$\begin{aligned} L_{DC,i}^{(k)} &= -\log \frac{\exp(\langle \mathbf{z}_i^{(1)}, \mathbf{z}_i^{(2)}\rangle/\tau)}{\cancel{\exp(\langle \mathbf{z}_i^{(1)}, \mathbf{z}_i^{(2)}\rangle/\tau)} + U_{i,k}} \\ &= -\langle \mathbf{z}_i^{(1)}, \mathbf{z}_i^{(2)}\rangle/\tau + \log U_{i,k} \end{aligned}$$

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

Proof. By removing the positive term in the denominator of Equation 1, 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 1 shows that, we compare with the state-of-the-art SSL approaches on ImageNet-1K. For fairness, we list each approach's batch size and learning epoch, shown in the original paper. During pre-training, DCL is based on a ResNet-50 backbone, with two views with the size 224 × 224. DCL relies on its simplicity to reach competitive performance without relatively huge batch sizes and epochs or other pre-training schemes, i.e., momentum encoder, clustering, and prediction head. We report 400-epoch versions of DCL combined with NNCLR [7]. It achieves 71.1% under the batch size of 256 and 400-epoch pre-training, which is better than NNCLR [7] in their optimal case, 68.7% with a batch size of 256 and 1000-epoch. Note that SwAV [2], BYOL [8], SimCLR, and PIRL [12] need a huge batch size of 4096, and SwAV further applies multi-cropping extra views to reach optimal performance. The results of SwAV are taken from SimSiam that multi-cropping is not included.

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 asymmetric image augmentation of BYOL to replace default DCL augmentation in ablations. Table 3 demonstrates that the ImageNet-1K top-1 performance is increased from 67.8% to 68.2% by applying asymmetric augmentations.

Table 1. 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 large-scale experiment setting. We find that given lower computational budget, DCL model are better than other state-of-the-arts approaches. Its effectiveness **does not rely on** large batch size and learning epochs (SimCLR [4], NNCLR [7]), momentum encoding (BYOL [8], MoCo-v2 [5]), or other tricks such as stop-gradient (SimSiam [6]) and multi-cropping (SwAV [3]).

Method	Param. (M)	Batch Size	Epochs	Top-1 Linear (%)
NPID [17]	24	256	200	56.5
MoCo [9]	24	256	200	60.6
CMC [14]	47	256	280	64.1
MoCo-v2 [5]	28	256	200	67.5
SwAV [3]	28	4096	200	69.1
SimSiam [6]	28	256	200	70.0
InfoMin [15]	28	256	200	70.1
BYOL [8]	28	4096	200	70.6
SiMo [20]	28	256	200	68.0
Hypersphere [16]	28	256	200	67.7
SimCLR [4]	28	256	200	61.8
SimCLR+DCL	28	256	200	67.8
SimCLR+DCL(w/ BYOL aug.)	28	256	200	68.2
PIRL [12]	24	256	800	63.6
BYOL [8]	28	4096	400	73.2
SwAV [3]	28	4096	400	70.7
MoCo-v2 [5]	28	256	400	71.0
SimSiam [6]	28	256	400	70.8
Barlow Twins [18]	28	256	300	70.7
SimCLR [4]	28	4096	1000	69.3
SimCLR+DCL	28	256	400	69.5
NNCLR [7]	28	256	1000	68.7
NNLCR+DCL	28	256	400	71.1
NNCLR [7]	28	512	1000	71.7
NNCLR+DCL	28	512	400	72.3

Linear evaluation. Following the OpenSelfSup benchmark [19], 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 [6], 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 [16], 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:

$$L_{DCL} = L_{DCL,pos} + L_{DCL,neg}$$

$$L_{align-uni} = L_{align} + L_{uniform}$$

where

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

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 $L_{uniform}$:

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

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

Similar to the 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.

A case can simply demonstrate the negative-negative coupling in [16]. Let's assume the model has the batch size of 3, and temperature τ is 1. Both $L_{DCL,neg}$ and $L_{uniform}$ can be formulated as follows:

$$\begin{aligned} L_{DCL,neg} = \log(&\exp(\langle \mathbf{z}_1^{(k)}, \mathbf{z}_2^{(l)} \rangle) + \exp(\langle \mathbf{z}_1^{(k)}, \mathbf{z}_3^{(l)} \rangle))+ \\ \log(&\exp(\langle \mathbf{z}_2^{(k)}, \mathbf{z}_1^{(l)} \rangle) + \exp(\langle \mathbf{z}_2^{(k)}, \mathbf{z}_3^{(l)} \rangle))+ \\ \log(&\exp(\langle \mathbf{z}_3^{(k)}, \mathbf{z}_1^{(l)} \rangle) + \exp(\langle \mathbf{z}_3^{(k)}, \mathbf{z}_2^{(l)} \rangle)) \end{aligned}$$

$$\begin{aligned} L_{uniform} = \log(&\exp(\langle \mathbf{z}_1^{(k)}, \mathbf{z}_2^{(l)} \rangle) + \exp(\langle \mathbf{z}_1^{(k)}, \mathbf{z}_3^{(l)} \rangle) + \exp(\langle \mathbf{z}_2^{(k)}, \mathbf{z}_1^{(l)} \rangle)+ \\ &\exp(\langle \mathbf{z}_2^{(k)}, \mathbf{z}_3^{(l)} \rangle) + \exp(\langle \mathbf{z}_3^{(k)}, \mathbf{z}_1^{(l)} \rangle) + \exp(\langle \mathbf{z}_3^{(k)}, \mathbf{z}_2^{(l)} \rangle)) \end{aligned}$$

Table 2. 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 3. 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)**

If the value of $\exp(\langle \mathbf{z}_1^{(k)}, \mathbf{z}_3^{(l)} \rangle)$ is much larger (e.g., hard negatives) than other terms, there would be a huge difference between $L_{DCL,neg}$ and $L_{uniform}$. Since $L_{uniform}$ first sums up all the negative pair samples in the batch together, it may cause the loss to be dominated by a specific negative pair sample. Thus, in the DCL loss, the negative samples from different positives are not coupled in contrast to the uniformity loss in [16].

Next, we provide a comprehensive empirical comparison. The empirical experiments match the 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 Hypersphere did a pretty thorough hyperparameter search. We believe the default hyperparameters are relatively optimized for Hypersphere.

In Table 2, DCL reaches 84.4% (fc7+Linear) compared to 83.2% (fc7+Linear) reported in Hypersphere on STL10. In Table 3 and Table 4, DCL achieves better performance than Hypersphere under the same setting (MoCo & MoCo-v2) on ImageNet-100 data. DCL further shows strong results compared against Hypersphere on ImageNet-1K in Table 5. We also provide the STL10 comparisons of DCL and Hypersphere under different batch sizes in Table 6. 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. Although the difference between the DCL and Hypersphere is slight, it makes DCL more easier to alleviate the domination from a specific negative pair in a batch. We hope these results show the unique value of DCL compared to Hypersphere.

Table 4. 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 5. 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 (%)
MoCo-v2 Baseline	200	256 (Memory queue = 65536)	67.5
Hypersphere	200	256 (Memory queue = 65536)	67.7 (+0.2)
DCL	200	256	**68.2 (+0.7)**

Table 6. 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)**

Table 7. 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 [13].

‡ 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 [11].

Table 8. Comparisons between the cross entropy and DCL in supervised classifier under different numbers of batch sizes (32, 128, and 256).

Architecture@epoch	ResNet-20@200 epoch					
Batch Size	32	128	256	32	128	256
Dataset	CIFAR10 (top-1)			CIFAR100 (top-1)		
Cross entropy	91.5	92.3	91.0	61.9	62.7	61.8
DCL	89.2	91.4	91.2	60.2	61.8	61.4

A.6 DCL on speech models

The SOTA SSL speech models, e.g., wav2vec 2.0 [1] still uses contrastive loss in the objective function. In Table 7, we show the effectiveness of DCL with wav2vec 2.0 [1]. We replace the InfoNCE loss with the DCL loss and train a wav2vec 2.0 base model (i.e., 7-Conv + 24-Transformer) from scratch.[1] After the pre-training of model, we evaluate the representation on two downstream tasks, speaker identification and intent classification. Table 7 shows the representation improvement of DCL.

A.7 Supervised classifier: DCL vs Cross-Entropy

The idea of DCL, removing positive from the denominator, can also be applied for learning objective function in the supervised classifier. By following [10], we implement the proposed DCL idea on cross entropy loss by removing the positive logits from the denominator of the softmax function. In Table 8, it is observed that our supervised DCL achieves slightly lower performance while comparing to the cross-entropy on CIFAR data. One possible reason for undermined performance of DCL in supervised learning might be the different feature interaction between supervised and unsupervised classifiers, which are referred to as parametric and non-parametric classifiers in [17].

Under the parametric formulation in [17], the logits equal to $w^T z$, where w is a weight vector for each class and z is the output embedding of the neural network. While in contrastive learning (i.e., non-parametric classifier), the logits equal to $z^{(1)} z^{(2)}$, where $z^{(1)}$ and $z^{(2)}$ are two augmented views of the same sample. In the embedding space of the early training stage, w is relatively far away from z compared to the relation between $z^{(1)}$ and $z^{(2)}$. Consider the effect of NPC multiplier q_b into parametric and non-parametric classifier, $q_b \to 1$ in parametric classifier might diminish the effectiveness of DCL idea as the coupling effect is already tiny.

A.8 Ablations of DCLW

Based on weighting function for the positive pairs in the Section 3 of the manuscript, we provide an another weighting function of DCLW:

$$L_{DCLW} = \sum_{k \in \{1,2\}, i \in [\![1,N]\!]} L_{DCLW,i}^{(i,k)} \tag{2}$$

$$L_{DCLW,i}^{(k)} = -w(\mathbf{z}_i^{(1)}, \mathbf{z}_i^{(2)})(\langle \mathbf{z}_i^{(1)}, \mathbf{z}_i^{(2)} \rangle / \tau) + \log U_{i,k} \tag{3}$$

where $w(\mathbf{z}_i^{(1)}, \mathbf{z}_i^{(2)}) = \delta \cdot \exp(-\sigma \cdot \langle \mathbf{z}_i^{(1)}, \mathbf{z}_i^{(2)} \rangle)$. Basically, the goal is similar to DCLW that we provide larger weight to hard positives (e.g., a positive pair of samples are far away from each other).

The results indicate that $\delta = 3$ and $\sigma = 0.5$ can achieve 85.4% kNN top-1 accuracy on CIFAR-10, and outperform the InfoNCE baseline (SimCLR) by 4%.

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

Table 9. The ablation study of various temperatures τ on CIFAR10.

Temperature τ	0.07	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1.0	Std
SimCLR	83.6	87.5	89.5	89.2	88.7	89.1	88.5	87.6	86.8	85.9	85.3	1.44
SimCLR w/ DCL	**88.3**	**89.4**	**90.8**	**89.9**	**89.6**	**90.3**	**89.6**	**89.0**	**88.5**	**88.0**	**87.7**	**0.98**

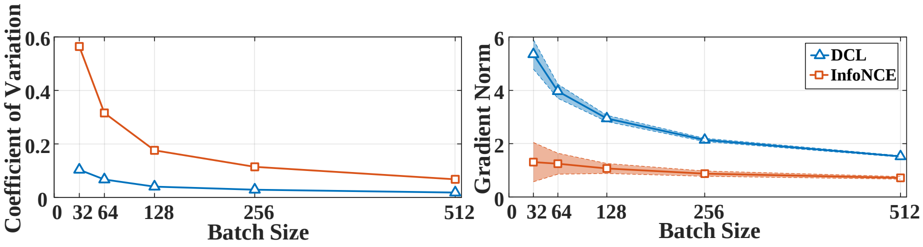

Fig. 1. (a) The coefficient of variation ($C_v = \sigma/\mu$) of gradient and (b) the mean gradient norm with its std of baseline (InfoNCE) and proposed method (DCL) under different batch sizes.

A.9 Additional Discussion

Analysis of Temperature τ. In Table 9, we further provide extensive analysis on temperature τ in the objective function to support that the DCL method is not sensitive to hyperparameters compared against the InfoNCE-based baselines. In the following, show the temperature τ search on both DCL and SimCLR on CIFAR10 data. Specifically, we pre-train the network with temperature τ in $\{0.07, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1.0\}$ and report results with kNN eval, batch size 512, and 500 epochs. As shown in Table 9, compared to SimCLR, DCL is less sensitive to hyperparameters, e.g., temperature τ.

Analysis of Gradient. For further analysis of the phenomenon of DCL, we visualize the mean norm with its std of the last convolutional layers from the last two residual blocks of ResNet-18 trained on CIFAR-100 under different batch sizes. The results in Figure 1 show that DCL constantly achieves larger gradients than baseline (InfoNCE) loss, especially under small batch sizes.

References

1. Baevski, A., Zhou, Y., Mohamed, A., Auli, M.: wav2vec 2.0: A framework for self-supervised learning of speech representations. In: Advances in Neural Information Processing Systems (NeurIPS) (2020) 8
2. Caron, M., Bojanowski, P., Joulin, A., Douze, M.: Deep clustering for unsupervised learning of visual features. In: European Conference on Computer Vision (ECCV) (2018) 3
3. Caron, M., Misra, I., Mairal, J., Goyal, P., Bojanowski, P., Joulin, A.: Unsupervised learning of visual features by contrasting cluster assignments. In: Advances in Neural Information Processing Systems (NeurIPS) (2020) 4

4. Chen, T., Kornblith, S., Norouzi, M., Hinton, G.E.: A simple framework for contrastive learning of visual representations. In: Proceedings of the International Conference on Machine Learning (ICML) (2020) 4
5. Chen, X., Fan, H., Girshick, R.B., He, K.: Improved baselines with momentum contrastive learning. CoRR **abs/2003.04297** (2020) 4
6. Chen, X., He, K.: Exploring simple siamese representation learning. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2021) 4
7. Dwibedi, D., Aytar, Y., Tompson, J., Sermanet, P., Zisserman, A.: With a little help from my friends: Nearest-neighbor contrastive learning of visual representations. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 9588–9597 (2021) 3, 4
8. Grill, J., Strub, F., Altché, F., Tallec, C., Richemond, P.H., Buchatskaya, E., Doersch, C., Pires, B.Á., Guo, Z., Azar, M.G., Piot, B., Kavukcuoglu, K., Munos, R., Valko, M.: Bootstrap your own latent - A new approach to self-supervised learning. In: Advances in Neural Information Processing Systems (NeurIPS) (2020) 3, 4
9. He, K., Fan, H., Wu, Y., Xie, S., Girshick, R.: Momentum contrast for unsupervised visual representation learning. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2020) 4
10. Idelbayev, Y.: Proper ResNet implementation for CIFAR10/CIFAR100 in PyTorch. https://github.com/akamaster/pytorch_resnet_cifar10, accessed: 20xx-xx-xx 8
11. Lugosch, L., Ravanelli, M., Ignoto, P., Tomar, V.S., Bengio, Y.: Speech model pretraining for end-to-end spoken language understanding. In: the Annual Conference of the International Speech Communication Association (InterSpeech) (2019) 7
12. Misra, I., Maaten, L.v.d.: Self-supervised learning of pretext-invariant representations. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2020) 3, 4
13. Nagrani, A., Chung, J.S., Xie, W., Zisserman, A.: Voxceleb: Large-scale speaker verification in the wild. Comput. Speech Lang. **60** (2020) 7
14. Tian, Y., Krishnan, D., Isola, P.: Contrastive multiview coding. In: European Conference on Computer Vision (ECCV) (2020) 4
15. Tian, Y., Sun, C., Poole, B., Krishnan, D., Schmid, C., Isola, P.: What makes for good views for contrastive learning? In: Advances in Neural Information Processing Systems (NeurIPS) (2020) 4
16. Wang, T., Isola, P.: Understanding contrastive representation learning through alignment and uniformity on the hypersphere. In: International Conference on Machine Learning (ICML) (2020) 4, 5, 6
17. Wu, Z., Xiong, Y., Yu, S.X., Lin, D.: Unsupervised feature learning via nonparametric instance discrimination. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2018) 4, 8
18. Zbontar, J., Jing, L., Misra, I., LeCun, Y., Deny, S.: Barlow twins: Self-supervised learning via redundancy reduction. In: International Conference on Machine Learning. pp. 12310–12320. PMLR (2021) 4
19. Zhan, X., Xie, J., Liu, Z., Lin, D., Change Loy, C.: OpenSelfSup: Open mmlab self-supervised learning toolbox and benchmark. https://github.com/open-mmlab/openselfsup (2020) 4
20. Zhu, B., Huang, J., Li, Z., Zhang, X., Sun, J.: Eqco: Equivalent rules for self-supervised contrastive learning. arXiv preprint arXiv:2010.01929 (2020) 4