ReSSL: Relational Self-Supervised Learning with Weak Augmentation

Anonymous Author(s)
Affiliation
Address
email

Abstract

Self-supervised Learning (SSL) including the mainstream contrastive learning has achieved great success in learning visual representations without data annotations. However, most of methods mainly focus on the instance level information (*i.e.*, the different augmented images of the same instance should have the same feature or cluster into the same class), but there is a lack of attention on the relationships between different instances. In this paper, we introduced a novel SSL paradigm, which we term as relational self-supervised learning (ReSSL) framework that learns representations by modeling the relationship between different instances. Specifically, our proposed method employs sharpened distribution of pairwise similarities among different instances as *relation* metric, which is thus utilized to match the feature embeddings of different augmentations. Moreover, to boost the performance, we argue that weak augmentations matter to represent a more reliable relation, and leverage momentum strategy for practical efficiency. Experimental results show that our proposed ReSSL significantly outperforms the previous state-of-the-art algorithms in terms of both performance and training efficiency.

1 Introduction

Recently, self-supervised learning (SSL) has shown its superiority and achieved promising results for unsupervised visual representation learning in computer vision tasks [38, 26, 30, 7, 10, 44, 23, 24]. The purpose of a typical self-supervised learning algorithm is to learn general visual representations from a large amount of data without human annotations, which can be transferred or leveraged in downstream tasks (*e.g.*, classification, detection, and segmentation). Some previous works [6, 23] even have proven that a good unsupervised pretraining can lead to a better downstream performance than supervised pretraining.

Among various SSL algorithms, contrastive learning [44, 42, 7] serves as a state-of-the-art framework, which mainly focuses on learning an invariant feature from different views. For example, instance discrimination is a widely adopted pre-text task as in [7, 24, 44], which utilizes the noisy contrastive estimation (NCE) to encourage two augmented views of the same image to be pulled closer on the embedding space but pushes apart all the other images away. Deep Clustering [5, 45, 6] is an alternative pre-text task that forces different augmented views of the same instance to be clustered into the same class. However, instance discrimination based methods will inevitably induce a class collision problem [2, 33, 11], where similar images should be pulled closer instead of being pushed away. Deep clustering based methods cooperated with traditional clustering algorithms to assign a label for each instance, which relaxed the constraint of instance discrimination, but most of these algorithms adopt a strong assumption, *i.e.*, the labels must induce an equipartition of the data, which might introduce some noise and hurt the learned representations.

Submitted to 35th Conference on Neural Information Processing Systems (NeurIPS 2021). Do not distribute.

In this paper, we introduce a novel Relational Self-Supervised Learning framework (ReSSL), which does not encourage explicitly to push away different instances, but uses *relation* as a manner to investigate the inter-instance relationships and highlight the intra-instance invariance. Concretely, we aim to maintain the consistency of pairwise similarities among different instances for two different augmentations. For example, if we have three instances $\mathbf{x}^1$, $\mathbf{x}^2$, $\mathbf{y}$ and $\mathbf{z}$ where $\mathbf{x}^1$, $\mathbf{x}^2$ are two different augmentations of $\mathbf{x}$, $\mathbf{y}$ and $\mathbf{z}$ are different samples. Then, if $\mathbf{x}^1$ is similar to $\mathbf{y}$ but different to $\mathbf{z}$, we wish $\mathbf{x}^2$ can maintain such relationship and vice versa. In this way, the relation can be modelled as a similarity distribution between a set of augmented images, and then use it as a metric to align the same images with different augmentations, so that the relationship between different instances could be maintained across different views.

However, this simple manner induces unexpectedly horrible performance if we following the same training recipe as other contrastive learning methods [7, 24]. We argue that construction of a proper relation matters for ReSSL; aggressive data augmentations as in [7, 8, 39] are usually leveraged by default to generate diverse positive pairs that increase the difficulty of the pre-text task. However, this hurts the reliability of the target relation. Views generated by aggressive augmentations might cause the loss of semantic information, so the target relation might be noisy and not that reliable. In this way, we propose to leverage weaker augmentations to represent the relation, since much lesser disturbances provide more stable and meaningful relationships between different instances. Besides, we also sharpen the target distribution to emphasize the most important relationship and utilize the memory buffer with a momentum update network to reduce the demand of large batch size for more efficiency. Experimental results on multiple benchmark datasets show the superiority of ReSSL in terms of both performance and efficiency. For example, with only 200 epochs of pre-training, our ReSSL achieved 69.6% on ImageNet [15] linear evaluation protocol, which is 2.1% higher than our baseline method (MoCoV2 [9]).

Our contributions can be summarized as follows.

- We proposed a novel SSL paradigm, which we term it as relational self-supervised learning (ReSSL). ReSSL maintains the relational consistency between the instances under different augmentations instead of explicitly pushing different instances away.
- Our proposed weak augmentation and sharpening distribution strategy provide a stable and high quality target similarity distribution, which makes the framework works well.
- ReSSL is a simple and effective SSL framework since it replaces the widely adopted contrastive loss with our proposed relational consistency loss. It achieved state-of-the-art performance under the same training cost.

2 Related Works

Self-Supervised Learning. Early works in self-supervised learning methods rely on all sorts of pretext to learn visual representations. For example, colorizing gray-scale images [46], image jigsaw puzzle [37], image super-resolution [31], image inpainting [19], predicting a relative offset for a pair of patches [17], predicting the rotation angle [32], and image reconstruction [3, 22, 4, 18]. Although these methods have shown their effectiveness, they lack the generality of the learned representations.

Instance Discrimination. The recent contrastive learning methods [30, 38, 7, 24, 39, 36, 26, 28] have made a lot of progress in the field of self-supervised learning. Most of the previous contrastive learning methods are based on the instance discrimination [44] task in which positive pairs are defined as different views of the same image, while negative pairs are formed by sampling views from different images. SimCLR [7, 8] shows that image augmentation (*e.g.*Grayscale, Random Resized Cropping, Color Jittering, and Gaussian Blur), nonlinear projection head and large batch size plays a critical role in contrastive learning. Since large batch size usually requires a lot of GPU memory, which is not very friendly to most of researchers. MoCo [24, 9] proposed a momentum contrast mechanism that forces the query encoder to learn the representation from a slowly progressing key encoder and maintain a memory buffer to store a large number of negative samples. InfoMin [39] proposed a set of stronger augmentation that reduces the mutual information between views while keeping task-relevant information intact. AlignUniform [42] shows that alignment and uniformity are two critical properties of contrastive learning.

Deep Clustering. In contrast to instance discrimination which treats every instance as a distinct class, deep clustering [5] adopts the traditional clustering method (*e.g.*KMeans) to label each image iteratively. Eventually, similar samples will be clustered into the same class. Simply apply the KMeans algorithm might lead to a degenerate solution where all data points are mapped to the same cluster; SeLa [45] solved this issue by adding the constraint that the labels must induce equipartition of the data and proposed a fast version of the Sinkhorn-Knopp to achieve this. SwAV [6] further extended this idea and proposed a scalable online clustering framework. PCL [33] reveals the class collision problem and simply performed instance discrimination and unsupervised clustering simultaneously; although it gets the same linear classification accuracy with MoCoV2, it has better performance on downstream tasks.

Contrastive Learning Without Negatives. Most previous contrastive learning methods prevent the model collapse in an explicit manner (*e.g.*push different instances away from each other or force different instances to be clustered into different groups.) BYOL [23] can learn a high-quality representation without negatives. Specifically, it trains an online network to predict the target network representation of the same image under a different augmented view and using an additional predictor network on top of the online encoder to avoiding the model collapse. SimSiam [10] shows that simple Siamese networks can learn meaningful representations even without the use of negative pairs, large batch size, and momentum encoders.

3 Methodology

In this section, we will first revisit the preliminary work on contrastive learning; then, we will introduce our proposed relational self-supervised learning framework. After that, the algorithm and the implementation details will also be explained.

3.1 Preliminaries on Self-supervised Learning

Given N unlabeled samples $\mathbf{x}$, we randomly apply a composition of augmentation functions $T(\cdot)$ to obtain two different views $\mathbf{x}^1$ and $\mathbf{x}^2$ through $T(\mathbf{x}, \theta_1)$ and $T(\mathbf{x}, \theta_2)$ where θ is the random seed for T. Then, a convolutional neural network based encoder $\mathcal{F}(\cdot)$ is employed to extract the information from these samples, i.e., $\mathbf{h} = \mathcal{F}(T(\mathbf{x}, \theta))$. Finally, a two-layer non-linear projection head $g(\cdot)$ is utilized to map $\mathbf{h}$ into embedding space, which can be written as: $\mathbf{z} = g(\mathbf{h})$). SimCLR [7] and MoCo [24] style framework adopt the noise contrastive estimation (NCE) objective for discriminating different instances in the dataset. Suppose $\mathbf{z}_i^1$ and $\mathbf{z}_i^2$ are the representations of two augmented views of $\mathbf{x}_i$ and $\mathbf{z}_k$ is a different instance. The NCE objective can be expressed by Eq. (1), where the similarity function $sim(\cdot)$ represents the dot product between L_2 normalized vectors $sim(\mathbf{u}, \mathbf{v}) = \mathbf{u}^T\mathbf{v}/\|\mathbf{u}\|\|\mathbf{v}\|$ and τ is the temperature parameter.

$$\mathcal{L}_{NCE} = -\log \frac{\exp(sim(\mathbf{z}^1, \mathbf{z}^2)/\tau)}{\exp(sim(\mathbf{z}_i^1, \mathbf{z}_i^2)/\tau) + \sum_{k=1}^{N} \exp(sim(\mathbf{z}_i^1, \mathbf{z}_k)/\tau)}. \tag{1}$$

BYOL [23] and SimSiam [10] style framework add an additional non-linear predictor head $q(\cdot)$ which further maps $\mathbf{z}$ to $\mathbf{p}$. The model will minimize the negative cosine similarity (equivalent to minimize the L2 distance) between $\mathbf{z}$ to $\mathbf{p}$.

$$\mathcal{L}_{cos} = -\frac{\mathbf{p}^1}{\|\mathbf{p}^1\|} \cdot \frac{\mathbf{z}^2}{\|\mathbf{z}^2\|}, \qquad \mathcal{L}_{mse} = \|\mathbf{p}^1 - \mathbf{z}^2\|_2^2. \tag{2}$$

Tricks like stop-gradient and momentum teacher are often applied to avoid model collapsing.

3.2 Relational Self-Supervised Learning

In classical self-supervised learning, different instances are to be pushed away from each other, and augmented views of the same instance is expected to be of exactly the same features. However, both constrains are too restricted because of the existence of similar samples and the distorted semantic information if aggressive augmentation is adopted. In this way, we do not encourage explicit negative instances (those to be pushed away) for each instance; instead, we leverage the pairwise similarities as a manner to explore their relationships. And we pull the features of two different augmentations in this sense of relation metric. As a result, our method relaxes both (1) and (2), where different

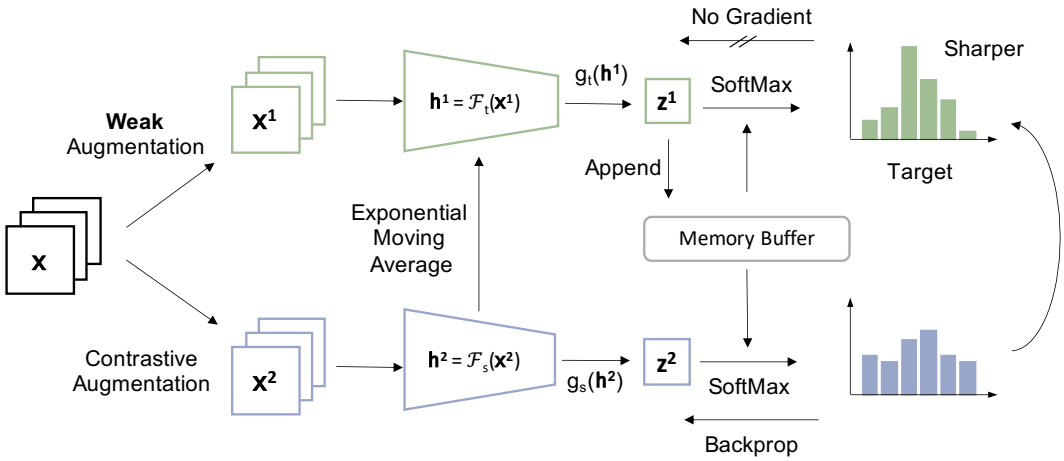

Figure 1: The overall framework of our proposed method. We adopt the student-teacher framework where the student is trained to predict the representation of the teacher, and the teacher is updated with a "momentum update" (exponential moving average) of the student. The relationship consistency is achieve by align the conditional distribution for student and teacher model. Please see more details in our method part.

instances do not always need to be pushed away from each other; and augmented views of the same instance only need to share the similar but not exactly the same features.

Concretely, given a image $\mathbf{x}$ in a batch of samples , two different augmented views can be obtained by $\mathbf{x}^1 = T(\mathbf{x}, \theta_1)$, $\mathbf{x}^2 = T(\mathbf{x}, \theta_2)$ and calculate the corresponds embedding $\mathbf{z}^1 = g(\mathcal{F}(\mathbf{x}^1))$, $\mathbf{z}^2 = g(\mathcal{F}(\mathbf{x}^2))$. Then, we calculate the similarities between the instances of the first augmented images. Which can be measured by $sim(\mathbf{z}^1, \mathbf{z}_i)$. A softmax layer can be adopted to process the calculated similarities, which then produces a relationship distribution:

$$\mathbf{p}_i^1 = \frac{\exp(sim(\mathbf{z}^1, \mathbf{z}_i)/\tau_t)}{\sum_{k=1}^{K} \exp(sim(\mathbf{z}^1, \mathbf{z}_k)/\tau_t,)}. \tag{3}$$

where τ_t is the temperature parameter. At the same time, we can calculate the relationship between $\mathbf{x}^2$ and the i-th instance as $sim(\mathbf{z}^2, \mathbf{z}_i)$. The resulting relationship distribution can be written as:

$$\mathbf{p}_i^2 = \frac{\exp(sim(\mathbf{z}^2, \mathbf{z}_i)/\tau_s)}{\sum_{k=1}^{K} \exp(sim(\mathbf{z}^2, \mathbf{z}_k)/\tau_s,)}. \tag{4}$$

where τ_s is a different temperature parameter. We propose to push the relational consistency between p_i^1 and p_i^2 by minimizing the Kullback–Leibler divergence, which can be formulated as:

$$\mathcal{L}_{relation} = D_{KL}(\mathbf{p}^1 || \mathbf{p}^2) = H(\mathbf{p}^1, \mathbf{p}^2) - H(\mathbf{p}^1). \tag{5}$$

Since the $\mathbf{p}^1$ will only be used as a target, the gradient will be clipped here to avoid the model collapsing, thus we only minimize the cross-entropy term $H(\mathbf{p}^1, \mathbf{p}^2)$ in our implementation.

More efficiency with Momentum targets. However, the quality of the target similarity distribution $\mathbf{p}^1$ is crucial, to make the similarity distribution reliable and stable, we usually require a large batch size which is very unfriendly to GPU memories. To resolve this issue, we utilize a "momentum update" network as in [24, 9], and maintain a large memory buffer $\mathcal{Q}$ of K past samples $\{\mathbf{z}_k | k = 1, ..., K\}$ (follow the FIFO principle) for storing the feature embeddings from the past batches, which can then be used for simulating the large batch size relationship and providing a stable similarity distribution.

$$\mathcal{F}_t \leftarrow m\mathcal{F}_t + (1-m)\mathcal{F}_s, \quad g_t \leftarrow mg_t + (1-m)g_s, \tag{6}$$

where $\mathcal{F}_s$ and g_s denote the most latest encoder and head, respectively, so we name them as the student model with a subscript s. On the other hand, $\mathcal{F}_t$ and g_t stand for ensembles of the past encoder and head, respectively, so we name them as the teacher model with a subscript t. m represents the momentum coefficient which controls how fast the teacher $\mathcal{F}_t$ will be updated.

Sharper Distribution as Target. Note, the value of τ_t has to be smaller than τ_s since τ_t will be used to generate the target distribution. A smaller τ will result in a "sharper" distribution which can be

interpreted as highlight the most similar feature for $\mathbf{z}^1$. Align $\mathbf{p}^2$ with $\mathbf{p}^1$ can be regarded as pulling $\mathbf{z}^2$ towards the features that are similar with $\mathbf{z}^1$.

Weak Augmentation Strategy for Teacher. To further improve the quality and stability of the target distribution, we adopt a weak augmentation strategy for the teacher model since the standard contrastive augmentation is too aggressive, which introduced too many disturbances and will mislead the student network. Please refers to more details in our empirical study.

Compare with SEED and CLSA. SEED [21] follows the standard Knowledge Distillation (KD) paradigm where it aims to distill the knowledge from a larger network into a smaller architecture. The knowledge transfer happens in the same view but between different models. In our framework, we are trying to maintain the relational consistency between different augmentations; the knowledge transfer happens between different views but in the same network. CLSA [43] also introduced the concept of using weak augmentation to guide a stronger augmentation. However, the "weak" augmentation in CLSA is equivalent to the "strong" augmentation in our method (We do not use any stronger augmentations such as [13, 14]). On the other hand, CLSA still adopts the InfoNCE loss (1) for instance discrimination, where our proposed method only utilized the relational consistency loss (5). Finally, CLSA requires at least one additional sample during training, which will slow down the training speed.

Algorithm 1 Relational Self-supervised Learning with Weak Augmentation (ReSSL)

Input : $\mathbf{x}$: a batch of samples. $T_w(\cdot)$: Weak augmentation function. $T_c(\cdot)$: Contrastive augmentation function. $\mathcal{F}_t$ and $\mathcal{F}_s$: the teacher and student backbone network. g_t and g_s : the non-linear projection head for teacher and student. Q: the memory buffer

while *network not converge* **do**

  for *i=1 to step* **do**

    Fetch $\mathbf{x}$ from current batch $\mathcal{B}$

    $\mathbf{z}^1 = g_t(\mathcal{F}_t(T_w(\mathbf{x}, \theta_1)))$; $\mathbf{z}^2 = g_s(\mathcal{F}_s(T_c(\mathbf{x}, \theta_2)))$;

    $\mathbf{p}^1$ = SoftMax($\mathbf{z}^1 Q^T$ / τ_t); $\mathbf{p}^2$ = SoftMax($\mathbf{z}^2 Q^T$ / τ_s) ; // Eq. (3)(4)

    Calculate $\mathcal{L}_{relation}$ loss by CrossEntropy($\mathbf{p}^1$, $\mathbf{p}^2$) ; // Eq. (5)

    Update $\mathcal{F}_s$ and g_s with loss $\mathcal{L}_{relation}$

    Update $\mathcal{F}_t$ and g_t by $\mathcal{F}_t \leftarrow m\mathcal{F}_t + (1-m)\mathcal{F}_s$ and $g_t \leftarrow mg_t + (1-m)g_s$; // Eq. (6)

    Update the memory buffer Q by $\mathbf{z}^1$

  end

end

Output : The well trained model $\mathcal{F}_s$

4 Empirical Study

In this section, we will empirically study our proposed method on 4 popular self-supervised learning benchmarks and compare to previous state-of-the-art algorithms (SimCLR [7], BYOL [23], SimSiam [10], MoCoV2 [9]).

Small Dataset. CIFAR-10 and CIFAR-100 [29]. The CIFAR-10 dataset consists of 60000 32x32 colour images in 10 classes, with 6000 images per class. There are 50000 training images and 10000 test images. CIFAR-100 is just like the CIFAR-10, except it has 100 classes containing 600 images each. There are 500 training images and 100 testing images per class.

Medium Dataset. STL-10 [12] and Tiny ImageNet [1]. STL10 [12] dataset is composed of 96x96 resolution images of 10 classes, 5K labeled training images, 8K validation images, and 100K unlabeled images. The Tiny ImageNet dataset is composed of 64x64 resolution images of 200 classes with 100K training images and 10k validation images.

Implementation Details We adopt the ResNet18 [25] as our backbone network. Because most of our dataset contains low-resolution images, we replace the first 7x7 Conv of stride 2 with 3x3 Conv of stride 1 and remove the first max pooling operation for a small dataset. For data augmentations, we use the random resized crops (the lower bound of random crop ratio is set to 0.2), color distortion (strength=0.5) with a probability of 0.8, and Gaussian blur with a probability of 0.5. The images from the small and medium datasets will be resized to 32x32 and 64x64 resolution respectively. Our method is based on MoCoV2 [9]; in order to simulate the shuffle BN trick on one GPU, we simply

divide a batch of data into different groups and then calculate BN statistics within each group. The momentum value and memory buffer size are set to 0.99/0.996 and 4096/16384 for small and medium datasets respectively. Moreover, The model is trained using SGD optimizer with a momentum of 0.9 and weight decay of $5e^{-4}$. We linear warm up the learning rate for 5 epochs until it reaches $0.06 \times BatchSize/256$, then switch to the cosine decay scheduler [34]. For the implementation details of other methods, please refers to the supplementary material.

Evaluation Protocol. All the models will be trained for 200 epochs. For testing the representation quality, we evaluate the pre-trained model on the widely adopted linear evaluation protocol - We will freeze the encoder parameters and train a linear classifier on top of the average pooling features for 100 epochs. To test the classifier, we use the center crop of the test set and computes accuracy according to predicted output. We train the classifier with a learning rate of 30, no weight decay, and momentum of 0.9. The learning rate will be times 0.1 in 60 and 80 epochs. Note, for STL-10; the pretraining will be applied on both labeled and unlabeled images. During the linear evaluation, only the labeled 5K images will be used.

Table 1: Compare to other SSL algorithms on small and medium dataset.

Method	BackProp	EMA	CIFAR-10	CIFAR-100	STL-10	Tiny ImageNet
Supervised	-	-	94.22	74.66	82.55	59.26
SimCLR [7]	2x	No	84.92	59.28	85.48	44.38
BYOL [23]	2x	Yes	85.82	57.75	87.45	42.70
SimSiam [10]	2x	No	88.51	60.00	87.47	37.04
MoCoV2 [9]	1x	Yes	86.18	59.51	85.88	43.36
ReSSL (Ours)	1x	Yes	**90.20**	**63.79**	**88.25**	**46.60**

Result. As we can see the result in Table 1, our proposed method outperforms the previous method on all four benchmarks. Reminder, most of the previous method requires twice back-propagation, which results in a much higher training cost than MoCoV2 and our method.

4.1 A Properly Sharpened Relation is A Better Target

The temperature parameter is very crucial in most contrastive learning algorithms. To verify the effective of τ_s and τ_t for our proposed method, we fixed $\tau_s = 0.1$ or 0.2, and sweep over $\tau_t = \{0.01, 0.02, ..., 0.07\}$. The result is shown in Table 2. For τ_t, the optimal value is either 0.04 or 0.05 across all different datasets. As we can see, the performance is increasing when we increase τ_t from 0 to 0.04 and 0.05. After that, the performance will start to decrease. Note, $\tau_t \to 0$ correspond to the Top-1 or $argmax$ operation which produce a one-hot distribution as the target. On the other hand, when $\tau_t \to 0.1$, the target will be a much flatter distribution that cannot highlight the most similar features for students. Hence, τ_t can not be either too small or too large, but it has to be smaller than τ_s ($\mathbf{p}^1$ has to be sharper than $\mathbf{p}^2$) so that the target distribution can provide effective guidance to the student model.

Table 2: Effect of different τ_t and τ_s for ReSSL

Dataset	τ_s	$\tau_t = 0.01$	$\tau_t = 0.02$	$\tau_t = 0.03$	$\tau_t = 0.04$	$\tau_t = 0.05$	$\tau_t = 0.06$	$\tau_t = 0.07$
CIFAR-10	0.1	89.35	89.74	90.09	90.04	**90.20**	90.18	88.67
CIFAR-10	0.2	89.52	89.67	89.24	89.50	89.22	89.40	89.50
CIFAR-100	0.1	62.34	62.79	62.71	**63.79**	63.46	63.20	61.31
CIFAR-100	0.2	60.37	60.05	60.24	60.09	59.09	59.12	59.76
STL-10	0.1	86.65	86.96	87.16	87.32	**88.25**	87.83	87.08
STL-10	0.2	85.17	86.12	85.01	85.67	85.21	85.51	85.28
Tiny ImageNet	0.1	45.20	45.40	46.30	**46.60**	45.08	45.24	44.18
Tiny ImageNet	0.2	43.28	42.98	43.58	42.12	42.70	42.76	42.60

For τ_t, it is clearly to see that the result of $\tau_s = 0.1$ can always result a much higher performance than $\tau_s = 0.2$, which is different to MoCoV2 where $\tau_s = 0.2$ is the optimal value. According to [40, 41, 16], a greater temperature will result in a larger angular margin in the hypersphere. Since MoCoV2 adopts instance discrimination as the pretext task, a large temperature can enhance the compactness for the same instance and discrepancy for different instances. In contrast to instance

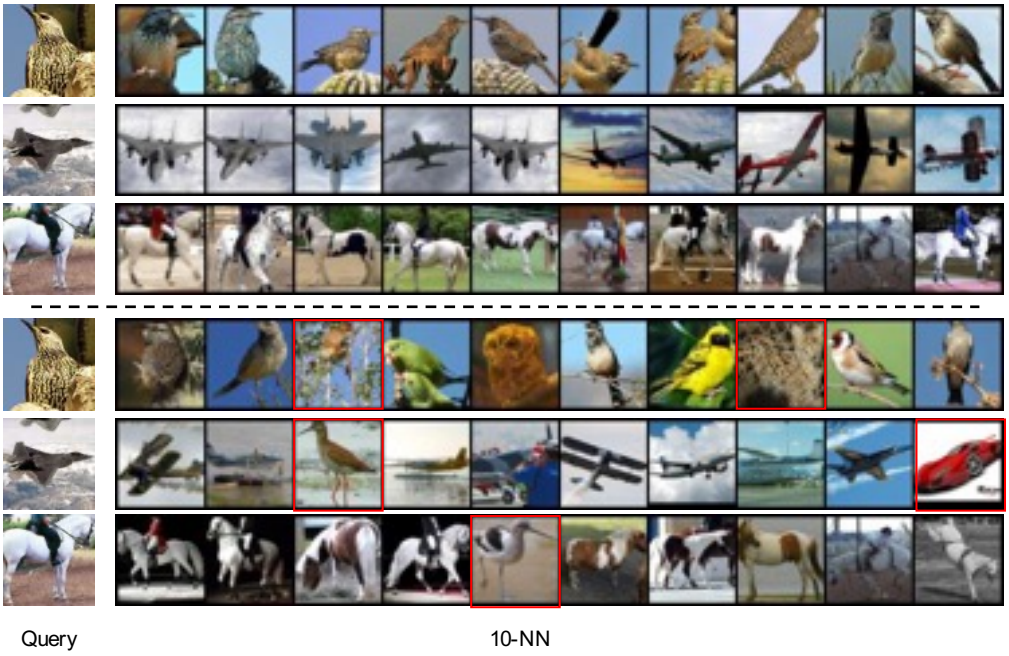

Figure 2: Visualization of the 10 nearest neighbour of the query image. The top half is the result when we apply the weak augmentation. The bottom half is the case when the typical contrastive augmentation is adopted. Note, we use the red square to highlight the images that has different ground truth label with the query image.

discrimination, our method can be interpreted as pulling similar instances closer on the hypersphere; when the ground truth label is not available, the large angular margin might hurt the performance.

4.2 Weak Augmentation Makes Better Relation

As we have mentioned, the weaker augmentation strategy for the teacher model is the key to the success of our framework. Here, We implement the weak augmentation as a random resized crop (the random ratio is set to $(0.2, 1)$) and a random horizontal flip. For temperature parameter, we simply adopt the same setting as in Table 2 and report the performance of the best setting. The result is shown in Table 3, as we can see that when we use the weak augmentation for the teacher model, the performance is significantly boosted across all datasets. We believe that this phenomenon is because relatively small disturbances in the teacher model can provide more accurate similarity guidance to the student model. To further verify this hypothesis, we random sampled three image from STL-10 training set as the query images, and then find the 10 nearest neighbour based on the weak / contrastive augmented query. We visualized the result in Figure 2,

Table 3: Effect of weak augmentation guided ReSSL

Teacher Aug	Student Aug	CIFAR-10	CIFAR-100	STL-10	Tiny ImageNet
Contrastive	Contrastive	86.17	57.60	84.71	40.38
Weak	Contrastive	**90.20**	**63.79**	**88.25**	**46.60**

4.3 Dimension of the Relation

Since we also adopt the memory buffer as in MoCo [24], the buffer size will be equivalent to the dimension of the distribution $\mathbf{p}^1$ $\mathbf{p}^2$. Thus, it will be one of the crucial points in our framework. To verify the effect the memory buffer size, we simply keep $\tau_s = 0.1$ and $\tau_t = 0.04$, then varying the memory buffer size from 256 to 32768. The result is shown in Table 4, as we can see that a larger memory buffer can significantly boost the performance. However, a further increase in the buffer size can only bring a marginal improvement when the buffer is large enough.

Table 4: Effect of different memory buffer size on small and medium dataset

Dataset (Small)	$K = 256$	$K = 512$	$K = 1024$	$K = 4096$	$K = 8192$	$K = 16384$
CIFAR-10	89.37	89.53	89.83	90.04	90.15	90.35
CIFAR-100	61.17	62.47	63.20	63.79	63.84	64.06
Dataset (Medium)	$K = 256$	$K = 1024$	$K = 4096$	$K = 8192$	$K = 16384$	$K = 32768$
STL-10	85.88	87.23	87.72	87.42	87.32	87.47
Tiny ImageNet	43.08	45.32	45.78	45.42	46.60	46.48

(a) MoCoV2 (b) Ours

Figure 3: t-SNE visualizations on CIFAR-10. Classes are indicated by colors.

4.4 Visualization of Learned Representations

We also show the t-SNE [35] visualizations of the representations learned by our proposed method and MoCov2 on the training set of CIFAR-10. Our proposed relational consistency loss leads to better class separation than the contrastive loss.

5 Results on Large-scale Datasets

We also performed our algorithm on the large-scale ImageNet-1k dataset [15]. Specifically, we will perform ReSSL on two different settings.

Table 5: Top-1 accuracy under the linear evaluation on ImageNet with the ResNet-50 backbone. The table compares the methods over 200 epochs of pretraining. ReSSL* denotes our stronger setting. The result of SwAV, BYOL with 200 epochs pretraining are copied from [10]

Method	Arch	Backprop	EMA	Batch Size	Param	Epochs	Top-1
Supervised	R50	1x	No	256	24	120	76.5
1x Backprop Methods							
InstDisc [44]	R50	1x	No	256	24	200	58.5
LocalAgg [47]	R50	1x	No	128	24	200	58.8
MoCo [24]	R50	1x	Yes	256	24	200	60.8
MoCo v2 [9]	R50	1x	Yes	256	24	200	67.5
MoCHi [28]	R50	1x	Yes	512	24	200	68.0
CPC v2 [30]	R50	1x	No	512	24	200	63.8
PCL v2 [33]	R50	1x	Yes	256	24	200	67.6
ReSSL (Ours)	R50	1x	Yes	256	24	200	**68.7**
ReSSL* (Ours)	R50	1x	Yes	256	24	200	**69.6**
2x Backprop Methods							
CLSA [27]	R50	2x	Yes	256	24	200	69.4
SimCLR [7]	R50	2x	No	4096	24	200	66.8
SwAV [6]	R50	2x	No	4096	24	200	69.1
SimSiam [23]	R50	2x	No	256	24	200	70.0
BYOL [23]	R50	2x	Yes	4096	24	200	70.6

1). The first training strategy is exactly the same with MoCoV2 [9], except we replace the InfoNCE loss (1) to our proposed relational consistency loss (5), and replace the contrastive augmentation with a weaker augmentation for teacher. This is for a fair comparison with our baseline (MoCoV2). For τ_t and τ_s, we simply adopt the best setting from Table 2 where $\tau_t = 0.04$ and $\tau_s = 0.1$. **2)**. Some previous works [8, 23] have shown that increasing the projection head size, embedding size, and the number of projection layers could further improve the models' performance. Thus, in this setting, we proposed using larger base learning rates $0.05 * BatchSize/256$, a larger memory buffer size 130k and a bigger non-linear projection head with a hidden dimension 4096 and output dimension 512.

Linear Evaluation. For the linear evaluation of ImageNet-1k, we strictly follow the setting in MoCoV2 [9], except we normalize the feature from the avg pooling layer since we found it will give the same result but less sensitive to different learning rates. The result is shown in table 5. As we can see clearly that in *1x* backprop methods, our ReSSL is significantly better than all previous algorithms. Comparing to the *2x* backprop methods, our ReSSL* can outperform CLSA, SimCLR, and SwAV with much lesser training costs.

Training Cost. As in Table 5, the framework with similar or better performance (than ReSSL) always requires an additional back-propagation. We further analyze the training efficiency for these methods. We test all the methods on V100 GPU with 32G memory. The details is shown in Table 6, the methods with additional back-propagation is roughly 56% - 70% slower than ReSSL and ReSSL*. More epochs of training or multi-crops strategy [11, 6] should further boost the performance our proposed ReSSL and ReSSL*.

Table 6: Training Cost Comparison, the result is reported in averaged GPU hours

Method	Epochs	Batch Size	Backprop	GPU	(GPU·Time)/Epoch
SimCLR	200	4096	2x	32 x V100	3.55
BYOL	200	4096	2x	32 x V100	3.88
SimSiam	200	256	2x	8 x V100	3.51
MoCoV2	200	256	1x	8 x V100	2.25
ReSSL (Ours)	200	256	1x	8 x V100	2.25
ReSSL* (Ours)	200	256	1x	8 x V100	2.33

Transfer Learning. Finally, We further evaluate the quality of the learned representations by transferring them to other datasets. Following [33], we perform linear classification on the PASCAL VOC2007 dataset [20]. Specifically, we resize all images to 256 pixels along the shorter side and taking a 224 × 224 center crop. Then, we train a linear SVM on top of corresponding global average pooled final representations. To study the transferability of the representations in few-shot scenarios, we vary the number of labeled examples K and report the mAP. Table 7 shows the comparison between our method with previous works. We report the average performance over 5 runs (except for k=full). For fair comparison, we use ReSSL (the first setting) to compare with MoCoV2 and PCLV2. It's clearly to see that our proposed method is consistently outperform MoCoV2 and PCLV2 across all different K.

Table 7: Transfer learning on low-shot image classification

Method	Epochs	ImageNet	K=4	K=8	K=16	K=32	K=64	Full
Random	-	-	10.10	10.42	10.82	11.34	11.96	12.42
Supervised	90	76.1	73.79	79.51	82.26	84.00	85.13	87.27
MoCo V2 [9]	200	67.5	64.85	72.47	76.14	79.16	81.52	84.60
PCL V2 [33]	200	67.5	66.21	74.45	78.34	80.72	82.67	85.43
ReSSL (Ours)	200	68.7	**67.16**	**75.09**	**79.32**	**81.57**	**83.45**	**85.96**

6 Conclusion

In this work, we propose relational self-supervised learning (ReSSL), a new paradigm for unsupervised visual representation learning framework that maintains the relational consistency between instances under different augmentations. Our proposed ReSSL relaxes the typical constraints in contrastive learning where different instances do not always need to be pushed away on the embedding space, and the augmented views do not need to share exactly the same feature. An extensive empirical study shows the effect of each component in our framework. The experiments on large-scaled datasets demonstrate the efficiency and state-of-the-art performance for unsupervised representation learning.

References

[1] Abai, Z., Rajmalwar, N.: Densenet models for tiny imagenet classification (2020) 5

[2] Arora, S., Khandeparkar, H., Khodak, M., Plevrakis, O., Saunshi, N.: A theoretical analysis of contrastive unsupervised representation learning. ArXiv **abs/1902.09229** (2019) 1

[3] Baldi, P.: Autoencoders, unsupervised learning and deep architectures. In: Proceedings of the 2011 International Conference on Unsupervised and Transfer Learning Workshop - Volume 27. p. 37–50. UTLW'11, JMLR.org (2011) 2

[4] Brock, A., Donahue, J., Simonyan, K.: Large scale gan training for high fidelity natural image synthesis. ArXiv **abs/1809.11096** (2019) 2

[5] Caron, M., Bojanowski, P., Joulin, A., Douze, M.: Deep clustering for unsupervised learning of visual features. In: European Conference on Computer Vision (2018) 1, 3

[6] Caron, M., Misra, I., Mairal, J., Goyal, P., Bojanowski, P., Joulin, A.: Unsupervised learning of visual features by contrasting cluster assignments (2020) 1, 3, 8, 9

[7] Chen, T., Kornblith, S., Norouzi, M., Hinton, G.: A simple framework for contrastive learning of visual representations. arXiv preprint arXiv:2002.05709 (2020) 1, 2, 3, 5, 6, 8

[8] Chen, T., Kornblith, S., Swersky, K., Norouzi, M., Hinton, G.: Big self-supervised models are strong semi-supervised learners. arXiv preprint arXiv:2006.10029 (2020) 2, 9

[9] Chen, X., Fan, H., Girshick, R., He, K.: Improved baselines with momentum contrastive learning. arXiv preprint arXiv:2003.04297 (2020) 2, 4, 5, 6, 8, 9

[10] Chen, X., He, K.: Exploring simple siamese representation learning (2020) 1, 3, 5, 6, 8

[11] Chuang, C.Y., Robinson, J., Lin, Y.C., Torralba, A., Jegelka, S.: Debiased contrastive learning. In: Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M.F., Lin, H. (eds.) Advances in Neural Information Processing Systems. vol. 33, pp. 8765–8775. Curran Associates, Inc. (2020), https://proceedings.neurips.cc/paper/2020/file/63c3ddcc7b23daa1e42dc41f9a44a873-Paper.pdf 1, 9

[12] Coates, A., Ng, A., Lee, H.: An analysis of single-layer networks in unsupervised feature learning. Proceedings of Machine Learning Research, vol. 15, pp. 215–223. JMLR Workshop and Conference Proceedings, Fort Lauderdale, FL, USA (11–13 Apr 2011), http://proceedings.mlr.press/v15/coates11a.html 5

[13] Cubuk, E.D., Zoph, B., Mane, D., Vasudevan, V., Le, Q.V.: Autoaugment: Learning augmentation policies from data (2019) 5

[14] Cubuk, E.D., Zoph, B., Shlens, J., Le, Q.V.: Randaugment: Practical automated data augmentation with a reduced search space (2019) 5

[15] Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: ImageNet: A Large-Scale Hierarchical Image Database. In: CVPR09 (2009) 2, 8

[16] Deng, J., Guo, J., Zafeiriou, S.: Arcface: Additive angular margin loss for deep face recognition. 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) pp. 4685–4694 (2019) 6

[17] Doersch, C., Gupta, A., Efros, A.A.: Unsupervised visual representation learning by context prediction. In: International Conference on Computer Vision (ICCV) (2015) 2

[18] Donahue, J., Simonyan, K.: Large scale adversarial representation learning. In: NeurIPS (2019) 2

[19] ElHarrouss, O., Almaadeed, N., Al-Máadeed, S., Akbari, Y.: Image inpainting: A review. Neural Processing Letters **51**, 2007–2028 (2019) 2

[20] Everingham, M., Van Gool, L., Williams, C.K.I., Winn, J., Zisserman, A.: The PASCAL Visual Object Classes Challenge 2007 (VOC2007) Results. http://www.pascal-network.org/challenges/VOC/voc2007/workshop/index.html 9

[21] Fang, Z., Wang, J., Wang, L., Zhang, L., Yang, Y., Liu, Z.: {SEED}: Self-supervised distillation for visual representation. In: International Conference on Learning Representations (2021), https://openreview.net/forum?id=AHm3dbp7D1D 5

[22] Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., Bengio, Y.: Generative adversarial nets. In: Ghahramani, Z., Welling, M., Cortes, C., Lawrence, N., Weinberger, K.Q. (eds.) Advances in Neural Information Processing Systems. vol. 27, pp. 2672–2680. Curran Associates, Inc. (2014), https://proceedings.neurips.cc/paper/2014/file/5ca3e9b122f61f8f06494c97b1afccf3-Paper.pdf 2

[23] Grill, J.B., Strub, F., Altché, F., Tallec, C., Richemond, P.H., Buchatskaya, E., Doersch, C., Pires, B.A., Guo, Z.D., Azar, M.G., Piot, B., Kavukcuoglu, K., Munos, R., Valko, M.: Bootstrap your own latent: A new approach to self-supervised learning (2020) 1, 3, 5, 6, 8, 9

[24] He, K., Fan, H., Wu, Y., Xie, S., Girshick, R.: Momentum contrast for unsupervised visual representation learning. arXiv preprint arXiv:1911.05722 (2019) 1, 2, 3, 4, 7, 8

[25] He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. arXiv preprint arXiv:1512.03385 (2015) 5

[26] Hjelm, R.D., Fedorov, A., Lavoie-Marchildon, S., Grewal, K., Bachman, P., Trischler, A., Bengio, Y.: Learning deep representations by mutual information estimation and maximization. arXiv preprint arXiv:1808.06670 (2018) 1, 2

[27] Hu, Q., Wang, X., Hu, W., Qi, G.J.: Adco: Adversarial contrast for efficient learning of unsupervised representations from self-trained negative adversaries. arXiv preprint arXiv:2011.08435 (2020) 8

[28] Kalantidis, Y., Sariyildiz, M.B., Pion, N., Weinzaepfel, P., Larlus, D.: Hard negative mixing for contrastive learning. In: Neural Information Processing Systems (NeurIPS) (2020) 2, 8

[29] Krizhevsky, A.: Learning multiple layers of features from tiny images (2009) 5

[30] Lai, C.I.: Contrastive predictive coding based feature for automatic speaker verification. arXiv preprint arXiv:1904.01575 (2019) 1, 2, 8

[31] Ledig, C., Theis, L., Huszár, F., Caballero, J., Aitken, A., Tejani, A., Totz, J., Wang, Z., Shi, W.: Photo-realistic single image super-resolution using a generative adversarial network. 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR) pp. 105–114 (2017) 2

[32] Lee, H., Hwang, S.J., Shin, J.: Self-supervised label augmentation via input transformations. In: ICML 2020 (2020) 2

[33] Li, J., Zhou, P., Xiong, C., Hoi, S.: Prototypical contrastive learning of unsupervised representations. In: International Conference on Learning Representations (2021), https://openreview.net/forum?id=KmykpuSrjcq 1, 3, 8, 9

[34] Loshchilov, I., Hutter, F.: Sgdr: Stochastic gradient descent with warm restarts. arXiv preprint arXiv:1608.03983 (2016) 6

[35] van der Maaten, L., Hinton, G.: Visualizing data using t-sne (2008) 8

[36] 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) (June 2020) 2

[37] Noroozi, M., Favaro, P.: Unsupervised learning of visual representations by solving jigsaw puzzles. In: ECCV (2016) 2

[38] Tian, Y., Krishnan, D., Isola, P.: Contrastive multiview coding. arXiv preprint arXiv:1906.05849 (2019) 1, 2

[39] Tian, Y., Krishnan, D., Isola, P.: Contrastive multiview coding. arXiv preprint arXiv:1906.05849 (2019) 2

[40] Wang, F., Xiang, X., Cheng, J., Yuille, A.: Normface: L2 hypersphere embedding for face verification. Proceedings of the 25th ACM international conference on Multimedia (2017) 6

[41] Wang, H., Wang, Y., Zhou, Z., Ji, X., Li, Z., Gong, D., Zhou, J., Liu, W.: Cosface: Large margin cosine loss for deep face recognition. 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition pp. 5265–5274 (2018) 6

[42] Wang, T., Isola, P.: Understanding contrastive representation learning through alignment and uniformity on the hypersphere. arXiv preprint arXiv:2005.10242 (2020) 1, 2

[43] Wang, X., Qi, G.J.: Contrastive learning with stronger augmentations (2021) 5

[44] Wu, Z., Xiong, Y., Stella, X.Y., Lin, D.: Unsupervised feature learning via non-parametric instance discrimination. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (2018) 1, 2, 8

[45] YM., A., C., R., A., V.: Self-labelling via simultaneous clustering and representation learning. In: International Conference on Learning Representations (2020), https://openreview.net/forum?id=Hyx-jyBFPr 1, 3

[46] Zhang, R., Isola, P., Efros, A.A.: Colorful image colorization. In: ECCV (2016) 2

[47] Zhuang, C., Zhai, A.L., Yamins, D.: Local aggregation for unsupervised learning of visual embeddings (2019) 8

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes] Please refers to our supplementary materials.
   (b) Did you describe the limitations of your work? [Yes] Please refers to our supplementary materials.
   (c) Did you discuss any potential negative societal impacts of your work? [Yes] Please refers to our supplementary materials.
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [Yes] Please refers to our supplementary materials.
   (b) Did you include complete proofs of all theoretical results? [Yes] Please refers to our supplementary materials.
3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [Yes]
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes] Please see Section 4 and Section-5.
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [Yes] Please see supplementary material.
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [Yes] PLease see Table 6.
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [Yes]
   (b) Did you mention the license of the assets? [Yes]
   (c) Did you include any new assets either in the supplemental material or as a URL? [Yes]
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [Yes]
   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [Yes]
5. If you used crowdsourcing or conducted research with human subjects...
   (a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]
   (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]
   (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]