Weak Augmentation Guided Relational Self-Supervised Learning

Mingkai Zheng, Shan You, Fei Wang, Chen Qian, Changshui Zhang, Fellow, IEEE, Xiaogang Wang, and Chang Xu, Senior Member, IEEE

Abstract—Self-supervised Learning (SSL) including the mainstream contrastive learning has achieved great success in learning visual representations without data annotations. However, most 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 introduce 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. To boost the performance, we argue that weak augmentations matter to represent a more reliable relation, and leverage momentum strategy for practical efficiency. The designed asymmetric predictor head and an InfoNCE warm-up strategy enhance the robustness to hyper-parameters and benefit the resulting performance. Experimental results show that our proposed ReSSL substantially outperforms the state-of-the-art methods across different network architectures, including various lightweight networks (e.g., EfficientNet and MobileNet).

Index Terms—Contrastive learning, unsupervised learning, self-supervised learning, representation learning.

I. INTRODUCTION

Recently, self-supervised learning (SSL) has shown its superiority and achieved promising results for unsupervised visual representation learning in computer vision tasks [10], [13], [38], [40], [44], [54], [74], [88]. 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 [8], [38] even have proven that a good unsupervised pretraining can lead to a better downstream performance than supervised pretraining.

Among various SSL algorithms, contrastive learning [10], [82], [88] 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 [10], [40], [88], 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 [7], [8], [91] 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], [17], [58], 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.

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 $x^1$, $x^2$, $y$ and $z$ where $x^1$, $x^2$ are two different augmentations of $x$, $y$, and $z$ is a different sample. Then, if $x^1$ is similar to $y$ but different to $z$, we wish $x^2$ 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 augmented views.

However, this simple manner induces unexpectedly horrible performance if we follow the same training recipe as other contrastive learning methods [10], [40]. We argue that construction of a proper relation matters for ReSSL; aggressive data augmentations as in [10], [11], [75] are usually leveraged
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 simply replaces the widely adopted contrastive loss with our proposed relational consistency loss. With a simple modification on MoCo, ReSSL achieved state-of-the-art performance with much lesser training cost.
- ReSSL is a lightweight friendly framework. Unlike previous contrastive learning algorithms that have a poor performance on small architectures, ReSSL consistently improves the performance for various lightweight networks and achieves better performance than KD algorithms.

II. RELATED WORKS

**Self-Supervised Learning With Pretext Tasks:** Self-supervised learning is a general representation learning framework that relies on the pretext tasks that can be formulated using only unlabeled data. The concept is “Obtain labels from the data itself by using a semi-automatic process” and “Predict part of the data from other parts” [61]. For this reason, many previous works focused on how to utilize the structural features of the images to design better pretext tasks that can learn more general representations. Many pretext tasks were found to be conducive to learning image features. For example, context prediction [24], jigsaw puzzles [67], and rotation angle prediction [36]. These methods can learn desired and transferable representations that achieve promising results in downstream tasks. Some of these tasks can even be used as an auxiliary task to improve the performance of the semi-supervised and supervised learning [50], [57], [97].

**Generative Based Model:** Image generation is always one of the most popular ideas for self-supervised learning. The early generative base model aims to encode the images to a latent vector and then decode it to reconstruct the original images [4], [51]. With the development of GAN [26], [37], some works [6], [26] have started to leverage the power of the discriminator to jointly differentiate both images and latent vectors. In this way, the generator not only maps latent samples to generated data but also has an inverse mapping from data to the latent representation. There are also some works aim to reconstruct the original image from the corrupted image, e.g., image colorization [98], super-resolution [56], and image inpainting [30].

**Instance Discrimination:** The recent contrastive learning methods [10], [40], [44], [47], [49], [54], [65], [74], [75] 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 [88] 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 [10], [11] shows that image augmentation (e.g., Grayscale, Random Resized Cropping, Color Jittering, and Gaussian Blur), nonlinear projection head and large batch size play a critical role in contrastive learning. Since large batch size usually requires a lot of GPU memory, which is not very friendly to most researchers. MoCo [12], [40] 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, it achieves better performance with smaller batch size. NNCLR [29] reverses the purpose of the memory buffer, where the goal is to find the most similar samples from the memory buffer and treat it as the positive pair to further increase the difficulty of the pretext task. InfoMin [75] proposed a set of stronger augmentation that reduces the mutual information between views while keeping task-relevant information intact. AlignUniform [82] 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 [7] adopts the traditional clustering method (e.g., KMeans) to label each image iteratively. Eventually, similar samples will be clustered into the same class. Simply applying the KMeans algorithm might lead to a degenerate solution where all data points are mapped to the same cluster; SeLa [91] 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 [8] further extended this idea and proposed a scalable online clustering framework by maintaining a set of prototypes and assigning the samples equally to each prototype. DINO [9] is another type of clustering framework that has a very similar objective. It replaces the Sinkhorn-Knopp with the sharpening and centering tricks, where the sharpening aims to minimize the entropy so that each sample will be assigned to a prototype, and centering aims to maximize the entropy to prevent all samples from being assigned into one prototype.

Class Collision Problem PCL [58] reveals the class collision problem in contrastive learning, since instance discrimination treats every sample as a single class, the similar samples will be undesirably pushed apart by the contrastive loss. PCL solved this problem by performing the Expectation-Maximization algorithm iteratively and performing the contrastive loss and clustering loss simultaneously, although it gets the same linear classification accuracy with MoCo V2 [12], it has better performance on downstream tasks. CLD [84] proposed a very similar idea where the only difference is that the clustering process will be applied on the fly in each batch. WCL [100] proposed a novel clustering algorithm by generating the connected component from the nearest neighbor graph. Different from the clustering-based method, AdpCLR [99] and FNC [48] directly find the top-K closest instances on the embedding space and treat these instances as their positive pairs to learn a much more compact representation for similar samples.

Contrastive Learning Without Negative Samples: 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 [38] 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 uses an additional predictor network on top of the online encoder to avoid model collapse. SimSiam [13] shows that simple Siamese networks can learn meaningful representations even without the use of negative pairs, large batch sizes, and momentum encoders. AdCo [46] shows that a set of negative features can replace the negative samples, and these features can be adversarially learned by maximizing the contrastive loss. Barlow Twins [96] proposed a redundancy-reduction principle that aims to make the cross-correlation matrix computed from twin representations as close to the identity matrix as possible. This objective can be regarded as a dual form of CL, since CL aims to decouple the features for each instance, and Barlow Twins aims to decouple the feature dimensions.

Contrastive Learning for Lightweight Architectures: Although contrastive learning has achieved remarkable performance for large networks (e.g., ResNet50 or larger), however, most of these methods have a horrible performance on lightweight architecture since it is pretty hard for smaller models to learn instance-level discriminative representation with a large amount of data. SEED [33] first proposed a knowledge distillation (KD) framework for contrastive learning; it requires the student network to mimic the similarity distribution inferred by the teacher over the memory buffer. Disco [35] introduced a different KD method which aims to constrain the last embedding of the student to be consistent with that of the teacher; it also alleviates the distilling bottleNeck problem and presents a large projection head can increase the capability of the model generalization. BINGO [90] utilized the teacher network to construct a group of similar instances; the objective is to aggregate compact representations over the student with respect to instances in the same group.

Relation-Based Knowledge Distillation This study is closely related to knowledge distillation, which aims to transfer knowledge from a larger teacher model to a small student model. Notably, works such as [15] and [94] have advanced this field by demonstrating the transfer of knowledge through the preservation of ranking orders among sample similarities. [94] enriches this approach by integrating hint and attention transfer mechanisms to maintain consistency in feature maps between the teacher and student models. In a related vein, [77] introduces a methodology for preserving the consistency of similarity matrices derived from the activation maps between student and teacher models. Moreover, [68] explores a congruent concept by ensuring the consistency of similarity distribution between the representations of different samples within a mini-batch. This approach, most akin to our current work, operates within a supervised learning paradigm, leveraging a pre-trained teacher model and explicit supervisory signals. However, unlike our approach, it does not confront the challenges associated with model collapse, a critical consideration in unsupervised settings.
III. METHODOLOGY

In this section, we will first revisit the preliminary works on contrastive learning; then, we will introduce our proposed relational self-supervised learning framework (ReSSL) (Fig. 1). After that, the algorithm and the implementation details will also be explained.

A. Preliminaries on Self-Supervised Learning

Given $N$ unlabeled samples $x$, we randomly apply a composition of augmentation functions $T(\cdot)$ to obtain two different views $x^1$ and $x^2$ through $T(x, \theta_1)$ and $T(x, \theta_2)$ where $\theta$ 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., $h = \mathcal{F}(T(x, \theta))$. Finally, a two-layer non-linear projection head $g(\cdot)$ is utilized to map $h$ into embedding space, which can be written as: $z = g(h)$. SimCLR [10] and MoCo [40] style framework adopt the noise contrastive estimation (NCE) objective for discriminating different instances in the dataset. Suppose $z^1_i$ and $z^2_i$ are the representations of two augmented views of $x_i$ and $z_k$ is a different instance. The NCE objective can be expressed by (1), where $\tau$ is the temperature parameter. Noted that we take the $L_2$ normalized vectors for representation $z$ by default.

$$
\mathcal{L}_{NCE} = -\log \frac{\exp(z^1_i \cdot z^2_i / \tau)}{\exp(z^1_i \cdot z^2_i / \tau) + \sum_{k=1}^{N} \exp(z^1_i \cdot z_k / \tau)}.
$$

BYOL [38] and SimSiam [13] style framework add an additional non-linear predictor head $q(\cdot)$ which further maps $z$ to $p$. The model will minimize the negative cosine similarity (equivalent to minimize the $L_2$ distance) between $z$ to $p$.

$$
L_{cos} = -\frac{p^1 \cdot z^2}{\|p^1\| \cdot \|z^2\|}, \quad L_{mse} = \|p^1 - z^2\|_2^2.
$$

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

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

$$
\begin{align*}
\mathcal{P}_i^1 &= \frac{\exp(z^1 \cdot z_i / \tau)}{\sum_{k=1}^{N} \exp(z^1 \cdot z_k / \tau)}, \\
\mathcal{P}_i^2 &= \frac{\exp(z^2 \cdot z_i / \tau)}{\sum_{k=1}^{N} \exp(z^2 \cdot z_k / \tau)}.
\end{align*}
$$

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

$$
\begin{align*}
\mathcal{P}_i &= \mathcal{P}_i^1 \cdot \mathcal{P}_i^2, \\
\mathcal{L}_{relation} &= D_{KL}(\mathcal{P}^1 || \mathcal{P}^2) = H(\mathcal{P}^2, \mathcal{P}^1) - H(\mathcal{P}^2).
\end{align*}
$$
Since the \( H(p^2) \) will be a constant value, we only minimize \( H(p^2, p^1) \) in our implementation.

**More Efficiency With Momentum Targets:** However, the quality of the target similarity distribution \( p^2 \) 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 [12], [40], and maintain a large memory buffer \( Q \) of \( K \) past samples \( \{z_k|k = 1, \ldots, K\} \) (following 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.

\[
F_t \leftarrow mF_t + (1 - m)F_s, \quad g_t \leftarrow mg_t + (1 - m)g_s, \tag{6}
\]

where \( F_s \) 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, \( F_t \) 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 \( F_t \) will be updated.

**Sharper Distribution as Target:** Note, the value of \( \tau_t \) has to be smaller than \( \tau_s \) since \( \tau_t \) will be used to generate the target distribution. A smaller \( \tau \) will result in a “sharper” distribution which can be interpreted as highlighting the most similar feature for \( z^2 \), since the similar features will have a higher value and the dissimilar features will have a lower value. Align \( p^1 \) with \( p^2 \) can be regarded as pulling \( z^2 \) towards the features that are similar with \( z^2 \). In this way, ReSSL makes the similar samples more similar, and dissimilar ones be more dissimilar.

**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 refer to more details in our empirical study.

**Compare With SEED and CLSA:** SEED [33] follows the standard Knowledge Distillation (KD) paradigm [28], [43], [92] 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 [83] 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 [20], [21] in our default setting). On the other hand, CLSA requires at least one additional sample during training, which requires a lot more GPU memories and slow down the training speed.

C. Improving ReSSL With Predictor and InfoNCE

**Asymmetric Head for Student:** For the student network, we adopt an additional predictor head as in [13], [38], which breaks the symmetry of the Siamese network. [13] demonstrates that the typical contrastive learning frameworks are just an Expectation-Maximization like algorithm; the predictor head helps fill the gaps between an augmented view and the expectation. In ReSSL, Breaking the symmetry could make the system become robust to the selection of the temperature, and also slightly improve the performance. Please see more empirical analysis in Section IV.

**Warm-Up With InfoNCE:** As we have described that the sharpening strategy is a crucial point in ReSSL, it can be interpreted as pulling \( z^1 \) towards to features that are similar with \( z^2 \). However, the network is not able to capture high-quality similar samples especially in the early stage of the training, which makes the target distribution \( p^2 \) unreliable. To resolve this issue, we add an additional InfoNCE loss to cooperate with our relational consistency loss, which can be expressed by (7)

\[
\mathcal{L}_{total} = \alpha \mathcal{L}_{relation} + (1 - \alpha) \mathcal{L}_{InfoNCE} \tag{7}
\]

Where \( \alpha \) is the hyper-parameter to control the weight of the two losses. Here, we set \( \alpha = \text{current} \_\text{step}/\text{warmup}\_\text{step} \). In this case, \( \alpha \) will increase from 0 to 1 along our optimization step. Basically, \( z^1 \) will be pulled towards to \( z^2 \) in the early stage since \( p^2 \) is not reliable, then, the objective function will gradually shift from (1) to (5). Although InfoNCE still introduces the clear positives and negatives, the weight will decrease along with the training. Experimental results show that warm-up with InfoNCE significantly improves the results of the linear evaluation. Please see more details in Section IV below.

IV. EMPIRICAL STUDY OF RESSL

In this section, we will empirically study our ReSSL on 4 popular self-supervised learning benchmarks. Specifically, we ablate the effectiveness of the sharpening, weak augmentation, predictor, and the InfoNCE warm-up strategy.

**Small Dataset:** CIFAR-10 and CIFAR-100 [53]. 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 [19] and Tiny ImageNet [55]. STL10[19] dataset is composed of 96x96 resolution images with 10 classes. It contains 5 K labeled training images, 8 K validation images, and 100 K unlabeled images. The Tiny ImageNet dataset is composed of 64x64 resolution images with 200 classes, which has 100 K training images and 10 K validation images.

**Implementation Details:** We adopt the ResNet18 [42] 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 [12]; we use the global BN and shuffle BN for with and without predictor setting. The momentum value and memory
buffer size are set to 0.99 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 $5\times 10^{-4}$. We linear warm up the learning rate for 10 epochs until it reaches $0.06 \times \text{Batch Size}/256$, then switch to the cosine decay scheduler [62].

**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 10, 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 5 K images will be used.

### A. A Properly Sharpened Relation is a Better Target

The temperature parameter is very crucial in most contrastive learning algorithms. To verify the effective of $\tau_s$ and $\tau_t$ for our proposed method, we fixed $\tau_s = 0.1$ or 0.2, and sweep over $\tau_t = \{0.01, 0.02, \ldots, 0.10\}$. The result is shown in Table I. Note, here we show the result for both with and without predictor setting. For $\tau_t$, the optimal value is 0.03 and 0.04 across all different datasets. As we can see, the performance is increasing when we increase $\tau_t$ from 0. After the optimal value, the performance will start to decrease.

Note, $\tau_t \to 0$ correspond to the Top-1 or $\text{argmax}$ operation, which produce a one-hot distribution as the target; this actually makes our relational consistent loss degrades to a standard InfoNCE objective where the positive pair is the most similar sample from the memory buffer instead of the augmentation of the original image. In this case, the system will be pretty similar to MoCoV2 where $\tau_s = 0.2$ is the optimal value. According to [23], [79], [81], 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 discrimination, our method can be interpreted as pulling similar instances closer on the hypersphere; the similar instances might not be very reliable, especially in the early stage of the training process. Thus, the large angular margin might hurt the performance when the ground truth label is not available.

### B. Asymmetric Structure Makes Robuster System

Our default setting adopts the predictor structure as in [13], [38] which breaks the symmetry of the system. It is however interesting to evaluate the performance with the typically symmetric system by removing the predictor structure from ReSSL. Thus, we follow the same experiment setting and present the result in Table II. As we can observe that in the case of $\tau_s = 0.1$ and $\tau_t \geq 0.08$ (the gray numbers), the symmetric system will collapse since the network can simply output a constant vector to minimize our loss function. Although ReSSL also works well without the predictor, we would like to keep it since it makes the system more robust to the temperature selection. We can also notice that the performance will be slightly better with the predictor in various temperature settings.

### C. Working With InfoNCE Warm-Up

Although a stand-alone relation metric already works well, the obvious problem is that the teacher network cannot capture high-quality relationships among different instances in the early stage of the training process. We addressed this issue by including the InfoNCE during the warm-up stage, as described in Section III-C. We show the result in Table IV below. As can be seen, the warm-up strategy significantly improves the performance on CIFAR-100, STL-10, and TinyImageNet.

### D. 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

| Dataset       | $\tau_s$ | $\tau_t$ |
|---------------|----------|----------|
| CIFAR-10      | 0.1      | 0.01     |
| CIFAR-10      | 0.2      | 0.02     |
| CIFAR-100     | 0.1      | 0.03     |
| CIFAR-100     | 0.2      | 0.04     |
| STL-10        | 0.1      | 0.05     |
| STL-10        | 0.2      | 0.06     |
| Tiny ImageNet | 0.1      | 0.07     |
| Tiny ImageNet | 0.2      | 0.08     |

| Dataset       | $\tau_s$ | $\tau_t$ |
|---------------|----------|----------|
| CIFAR-10      | 0.1      | 0.09     |
| CIFAR-100     | 0.2      | 0.10     |

The table shows the result for both with and without predictor setting. For $\tau_t$, the optimal value is 0.03 and 0.04 across all different datasets. As we can see, the performance is increasing when we increase $\tau_t$ from 0. After the optimal value, the performance will start to decrease.
flip. For temperature parameter, we simply adopt the same setting as in Table I and report the performance of the best setting. The result is shown in Table V, 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 images 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 Fig. 2.

E. More Experiments on Weak Augmentation

Since the weak augmentation for the teacher model is one of the crucial points in ReSSL, we further analyze the effect of applying different augmentations on the teacher model. In this experiment, we simply adopt the best temperature setting from Table I for each datasets and report the linear evaluation performance across the four benchmark dataset. The results are shown in Table III. The first row is the baseline, where we simply resize all images to the same resolution (no extra augmentation is applied). Then, we applied random resized crops, random flip,
color jitter, grayscale, gaussian blur, and various combinations. We empirically find that if we use no augmentation (e.g., no random resized crops) for the teacher model, the performance tends to degrade. This might result from that the gap of features between two views is way too smaller, which undermines the learning of representations. However, too strong augmentations of teacher model will introduce too much noise and make the target distribution inaccurate (see Fig. 2). Thus mildly weak augmentations are better option for the teacher, and random resized crops with random flip is the combination with the highest performance as Table III shows.

F. Dimension of the Relation

Since we also adopt the memory buffer as in MoCo [40], the buffer size will be equivalent to the dimension of the distribution $p^t p^s$. Thus, it will be one of the crucial points in our framework. To verify the effect of the memory buffer size, we simply adopt the best temperature setting from Table I for each dataset, then varying the memory buffer size from 256 to 32768. The result is shown in Table VI, as we can see that the performance could be significantly improved when we increase the buffer size from 256 to 8192. However, a further increase in the buffer size (i.e., 16384) can only bring a marginal improvement when the buffer is large enough. We can also observe that the performance will be slightly worse when $K = 32768$. This is possibly due to a too large memory buffer involving a lot of stale embeddings, which hurts the performance.

G. Visualization of Learned Representations

We also show the t-SNE [78] visualizations (Fig. 3) of the representations learned by ReSSL with contrastive augmentation, ReSSL with weak augmentation, and MoCo V2 on the test set of CIFAR-10. Obviously, ReSSL with weak augmentation leads to better class separation than the contrastive loss.

V. Experiments

A. Results on Large-Scale Datasets

We also performed our algorithm on the large-scale ImageNet-1k dataset [22]. In the experiments, we use a batch size of 1024, and adopt the LARS [93] optimizer with learning rate equals to $0.6 \times \frac{\text{BatchSize}}{256}$, momentum 0.9, and weight decay $1 \times 10^{-6}$. The optimizer will be scheduled by a cosine learning rate scheduler [62] with 10 epochs of Warm-up. We also adopt a memory buffer to save 65536 past examples. The momentum coefficient is $m = 0.996$ and increases to 1 with a cosine schedule. For the projector, we use a 2-layer non-linear MLP network with BN and ReLU; the hidden dimension is 4096, and the output dimension is 256. The predictor consists of the same structure as the projector, but input dimension is 256. For $\tau_t$ and $\tau_s$, we simply adopt the best setting from Table I where $\tau_t = 0.03$ and $\tau_s = 0.1$. For contrastive augmentation, we take the strategy from [38] with a few modifications on the parameters. Specifically, we set the probability of blurring / solarization to 50% / 10%, and also change the minimum crop area to 14%. For weak augmentation, we follow the best setting in Table III.

Linear Evaluation: For the linear evaluation of ImageNet-1k, we follow the same setting in SimCLR [8]. The results are shown in Table VII. Specifically, we train the linear classifier layer for 90 epochs using a SGD optimizer with a Nesterov momentum value of 0.9, weight decay of 0, learning rate with $0.1 \times \frac{\text{BatchSize}}{256}$, and a cosine-annealed learning rate scheduler. The result is shown in Table VII. For a fair comparison, we divide previous methods into 1x backprop and 2x backprop settings since 2x backprop methods generally require more GPU memories and longer training. As we can see clearly that ReSSL consistently outperforms previous methods on both setting. For 1x setting, ReSSL surpass previous state-of-the-art (InfoMin) 1.9%. For 2x setting, ReSSL has 0.4% improvements.
TABLE VII

| Method                  | Backprop | Batch Size | Epochs | Top-1 |
|-------------------------|----------|------------|--------|-------|
| Supervised              | 1x       | 256        | 120    | 76.5  |
| TX Backprop Methods     |          |            |        |       |
| NPID [88]               | 1x       | 256        | 200    | 58.5  |
| LocalAgg [102]          | 1x       | 128        | 200    | 58.8  |
| MoCo V2 [12]            | 1x       | 256        | 200    | 67.5  |
| MoCo V2 - WeakerAug [12]| 1x       | 256        | 200    | 68.3  |
| MoChI [49]              | 1x       | 512        | 200    | 68.0  |
| CPC v2 [54]             | 1x       | 512        | 200    | 63.8  |
| PCL v2 [58]             | 1x       | 256        | 200    | 67.6  |
| AdCo [46]               | 1x       | 256        | 200    | 68.6  |
| InfoMin [75]            | 1x       | 256        | 200    | 69.8  |
| ISD [73]                | 1x       | 256        | 200    | 69.9  |
| ReSSL * (1 Crop) [101]  | 1x       | 1024       | 200    | 72.0  |

2x Backprop Methods

| Method                  | Backprop | Batch Size | Epochs | Top-1 |
|-------------------------|----------|------------|--------|-------|
| NPID [88]               | 2x       | 256        | 200    | 69.9  |
| SimCLR [10]             | 2x       | 4096       | 200    | 66.8  |
| SimSiam [13]            | 2x       | 256        | 200    | 70.0  |
| MoCo V2 - Symmetric [13]| 2x       | 256        | 200    | 69.9  |
| WCL [100]               | 2x       | 4096       | 200    | 70.3  |
| NNCLR [29]              | 2x       | 256        | 200    | 70.7  |
| BYOL [38]               | 2x       | 4096       | 300    | 72.5  |
| BarlowTwins [96]        | 2x       | 2048       | 300    | 71.4  |
| MoCo V3 [14]            | 2x       | 4096       | 300    | 72.8  |
| SwAV [8]                | 2x       | 4096       | 400    | 70.1  |
| W-MSE 4 [31]            | 2x       | 1024       | 400    | 72.6  |
| ReSSL * (2 Crop) [101]  | 2x       | 256        | 200    | 71.4  |

ReSSL (2 Crops)

| Method                  | Backprop | Batch Size | Epochs | Top-1 |
|-------------------------|----------|------------|--------|-------|
| ReSSL                   | 2x       | 1024       | 200    | 73.2  |

ReSSL with * denote the results without predictor and InfoNCE warm-up.

over MoCo v3 with only 2/3 training costs and smaller batch size. We also report the performance when working without the predictor and InfoNCE warm-up. In this case, the performance drops 2.1% and 1.8% for 1x and 2x setting, respectively, which further verified the effectiveness of these two strategies.

Working With Multi-Crops and Stronger Augmentations: We also performed ReSSL with multi-Crop strategy and stronger augmentations. Specifically, we follow the strategy from [46] which adopts 5 augmented views with the resolution of 224 × 224, 192 × 192, 160 × 160, 128 × 128, 96 × 96, and set the min / max crop area to (0.14, 0.117, 0.095, 0.073, 0.05) / (1.0, 0.86, 0.715, 0.571, 0.429) for the 5 views respectively, the rest of augmentation strategy is consistent with our 1x and 2x setting. The result is shown in Table VIII. Notably, with 200 epochs of training, ReSSL achieves 76.0% Top-1 accuracy on ImageNet, which is significantly better than previous methods with even 800 or 1000 epochs of training. We tested our ReSSL model with extended training, specifically for 400 epochs, and found that it achieved the same performance as when trained for 200 epochs. This indicates that ReSSL converges more easily compared to other methods. Meanwhile, we also explored the effective of stronger augmentation to ReSSL. Concretely, we follow the same augmentation setting in [83] and optimize ReSSL for 200 epoch. In this case, ReSSL achieves 76.3% Top-1 accuracy which is almost on pair with the supervised baseline (76.5%).

Working With Vision Transformer: To underscore the broad applicability of our methodology, we additionally applied ReSSL to the Vision Transformer architecture [27], [76]. This included experiments with both the ViT-Small and ViT-Base models, with findings presented in Table IX. The results, particularly under the 2 crops setting, highlight ReSSL’s robust performance and its adaptability to various architectures, further attesting to its wide-ranging efficacy.

Transfer Learning on Linear Classification: We show representations learned by ReSSL are effective for transfer learning on multiple downstream classification tasks. We follow the linear evaluation setup described in [10], [38]. Specifically, we trained an L2-regularized multinomial logistic regression classifier on features extracted from the frozen pretrained network, then we used L-BFGS [60] to optimize the softmax cross-entropy objective, we did not apply data augmentation. We selected the best L2-regularization parameter and learning rate from validation splits and applied it to the test sets. The datasets used in this benchmark are as follows: CIFAR-10 [53], CIFAR-100 [53], Food101 [5], Cars [52], DTD [18], Oxford-IIIT-Pets [69], Aircraft [63], Oxford-Flowers [66], and Caltech-101 [34]. We present the results in Table X. It is clearly to see that ReSSL achieves state-of-the-art performance on CIFAR-10, Food-101,
Cars, DTD, Aircraft, and Caltech101, which is significantly better than SimCLR, BYOL, and NNCLR. We also assess the performance of ReSSL in the fine-tuning scenario, adhering strictly to the experimental setup outlined in previous works such as [10], [38]. This involves employing random crops resized to 224 × 224 pixels and random flips during training. During testing, images are resized to 256 pixels along the shorter side, followed by extraction of a central crop sized 224 × 224 pixels. We conduct a grid search for optimal hyperparameters on a validation set, including the learning rate and weight decay. The learning rate ranges logarithmically from 0.0001 to 0.1, while weight decay ranges between $10^{-6}$ and $10^{-3}$ as well as 0, with the latter values being divided by the learning rate. The results, presented in the second section of Table X, indicate that our ReSSL consistently outperforms prior methods on 6 datasets and achieves the highest mean accuracy across all 9 datasets.

**Semi-Supervised Learning:** Next, we evaluate the performance obtained when fine-tuning the model representation using only 1% and 10% labeled examples. We directly adopt the labeled and unlabeled file list from [10] for fair comparison. For 1%, we freeze the backbone layers and set the learning rate to 0.1 for the final linear layer. For 10%, we use a learning rate of 0.02 and 0.2 for the backbone layers and the final linear layer. The model will be optimized for 60 and 20 epochs for 1% and 10% setting with a cosine learning rate scheduler. We do not apply any weight decay in this experiment. The result has been shown in Table XI. With only 200 epochs of training, ReSSL outperforms NNCLR 1.4% and SwAV 1% with 1% and 10% labeled examples, demonstrating the superiority of the representation quality learned by ReSSL.

**Fine-Tuning on Full ImageNet:** We also investigate the efficacy of fine-tuning on the complete ImageNet dataset. In this experiment, we set the learning rate at 0.04 for the backbone layers and 0.01 for the linear layer. The model is trained over 30 epochs, with no weight decay. The results are presented in Table XII. Essentially, our ReSSL surpasses previous benchmarks by a margin of 0.6%, thereby demonstrating its superior performance.

**Transfer Learning on Low-Shot Classification:** We further evaluate the quality of the learned representations by transferring them to the low-shot classification task. Following [58], we perform linear classification on the PASCAL VOC2007 dataset [32]. 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 XIII shows the comparison between our method with previous works. We report the average performance over 5 runs (except for $k = \text{full}$). Notably, ReSSL consistently has a higher performance than other methods across different settings, and it also surpasses the supervised setting when $k$ is greater than 64.

**Object Detection and Instance Segmentation:** We finally evaluate the learned representations for the localization based tasks of object detection and instance segmentation on COCO [59] dataset by following the experiment setting in [40]. Specifically,
we use the ReSSL pretrained weight to initialize the Mask R-CNN [41] C4 backbone, the model will be finetuned on the COCO 2017 train split and report the results on the val split. We use a learning rate of 0.04 and keep the other parameters the same as in the default 2x schedule in detectron2 [87]. The results in Table XIV demonstrates that our ReSSL has a competitive performance with the state-of-the-art contrastive learning methods for these localization tasks.

Transfer Learning on Taskonomy: Recently, numerous studies, including those by [3], [85], [86], have begun to explore the transferability of pre-trained models for 3D vision tasks. To ensure the thoroughness of this research, we conducted a transfer of our pre-trained ViT-Base model (referenced in Table IX) to the Taskonomy [95] dataset, aiming to assess its performance on 3D vision tasks. We follow the same setting as in [3], where we use the taskonomy splits and fine-tune the model on a subset of 800 training images, we finally report L1 losses on the tiny-split test set, as reported in Table XV. Our findings indicate that while our ReSSL model outperforms DINO, but methods based on Masked Image Modeling (MIM) such as [3], [39], [86] generally surpass contrastive-based approaches. This aligns with our intuition, given that contrastive-based methods like DINO, MoCo, and our ReSSL tend to focus on capturing the global semantics of an image, which is advantageous for tasks like classification. Conversely, MIM-based methods emphasize predicting masked patches or patch features, directing more attention to the interactions among local image patches. This focus is particularly beneficial for tasks requiring detailed geometric information. Integrating MIM-based methods with contrastive-based approaches could potentially enhance the transfer performance for 3D vision tasks. We would like to explore this idea in our future work.

B. Results on Lightweight Architectures

We also applied our ReSSL on various lightweight architectures to further demonstrate the generality. By following the experimental setting in [3], [35], we performed ReSSL on ResNet-18, ResNet-34 [42], EfficientNet-B0, EfficientNet-B1 [71], and MobileNetV3-Large [45]. The linear evaluation result has been shown in Table XVI. We first present the result for SimCLR, MoCo V2, BYOL and SwAV on ResNet-18. Obviously, these methods have significant gaps with the supervised performance even with 800 epochs of training. Although the Knowledge distillation (KD) based methods (e.g., SEED, Compress, DisCo and BINGO) can somewhat improve the performance, our ReSSL is much more efficient since it does not require a large teacher. More importantly, the performance of ReSSL is consistently better than these KD methods especially on those extremely small networks (e.g., EfficientNet-B0 and Mobilenet-V3-Large).

Apart from the linear evaluation, we also report the performance for semi-supervised learning. We follow the same training strategy as our ResNet-50 experiments. Table XVII shows that our ReSSL achieves promising results for various lightweight architectures and outperforms the latest self-supervised KD methods by a clear margin.

C. More Ablation Studies on ImageNet

In this section, we delve deeper with extra ablation studies to show the impact that the InfoNCE warm-up and predictor strategy have on ImageNet performance. Additionally, we explore the intriguing properties similar to those discussed in [9]. For these experiments, we simplify the process by training our model over 100 epochs using a single-crop setting.

Effect of InfoNCE Warm-up and Predictor We present the impact of InfoNCE warm-up and predictor on ImageNet experiments in Table XVIII. It is observed that these two strategies enhance the performance of ReSSL, elevating it from 68.0% to 70.3%, which underscores their effectiveness.

Effect of Temperature Warm-up Dino proposed an innovative warm-up strategy for temperature, starting with a lower temperature for the teacher logits to avert model collapse, and then gradually increasing it. This method showed a slight improvement. To evaluate this approach, we conducted two experiments, as outlined in Table XIX. However, unlike DINO, this strategy negatively impacted the performance of ReSSL, leading us to not incorporate it in our research.

Teacher Performance versus Student Performance We also examine the performance of the teacher and student models, with the outcomes detailed in Table XX. Showing that the student model outperforms the teacher model when using ResNet-50. This discrepancy arises from the different augmentations applied to the teacher and student models, which leads to variations in Batch Norm statistics. For the teacher model, the parameter is optimized with strong augmentation, while its Batch Norm statistics are calculated using weak augmentation, resulting in performance disparities. To further investigate this phenomenon, we also evaluated the teacher and student models using the ViT-Small architecture, which substitutes the Batch Norm layer with
### TABLE XV
TRANSFER LEARNING PERFORMANCE ON TASKONOMY

| Method | Curvature | Depth | Edges | 2D-keypoints | 3D-keypoints | Normals | Oclusion | Reshading | Average |
|--------|-----------|-------|-------|--------------|--------------|---------|----------|-----------|---------|
| Contrastive Based | | | | | | | | | |
| DINO [9] | 43.04 | 38.42 | 3.80 | 0.16 | 45.85 | 65.71 | 0.57 | 115.02 | 39.07 |
| ReSSL (Ours) | 43.10 | 35.96 | 2.71 | 0.14 | 46.76 | 64.49 | 0.35 | 111.48 | 38.15 |
| MIM Based | | | | | | | | | |
| MAE [39] | 41.59 | 35.83 | 1.19 | 0.08 | 44.18 | 59.20 | 0.55 | 106.08 | 36.09 |
| MultiMAE [3] | 41.42 | 35.38 | 2.17 | 0.07 | 44.03 | 60.35 | 0.56 | 105.25 | 36.17 |
| CreCo (Habitat) [86] | 42.91 | 34.31 | 1.74 | 0.08 | 45.69 | 54.13 | 0.25 | 95.58 | 33.00 |

We report the L1 loss scaled by 1000. The lower number indicates the better performance. All methods employ pre-training with the Vit-Base on ImageNet, with the exception of CreCo [86], which is trained on habitat [64].

### TABLE XVI
TOP-1 ACCURACY UNDER THE LINEAR EVALUATION ON IMAGENET WITH VARIOUS LIGHTWEIGHT BACKBONES

| Method | Teacher Arch | Teacher Top-1 | Teacher Epochs | Student Top-1 |
|--------|---------------|---------------|----------------|---------------|
| ResNet-18 | - | - | 800 | 53.4 |
| SimCLR [10] | - | - | 800 | 56.1 |
| MoCo V2 [40] | ResNet-50 | 67.5 | 200 | 62.4 |
| SwAV [8] | ResNet-50 | 67.5 | 200 | 61.4 |
| BYOL [38] | ResNet-50 | 67.5 | 200 | 61.4 |
| SEED [33] | ResNet-50 | 67.5 | 200 | 61.4 |
| DISCO [35] | ResNet-50 | 67.5 | 200 | 61.4 |
| BINGO [90] | ResNet-50 | 67.5 | 200 | 61.4 |
| CreResNet [1] | ResNet-50 | 71.1 | 200 | 62.6 |
| OSS [16] | ResNet-50 | 75.3 | 200 | 64.1 |
| SimDis-Off [16] | ResNet-50 | 74.3 | 300 | 65.2 |
| ReSSL (1 Crop) | - | - | 200 | 62.4 |
| ReSSL (2 Crops) | - | - | 200 | 63.8 |
| ReSSL (Multi) | - | - | 200 | 66.5 |
| ResNet-34 | ResNet-50 | 67.5 | 200 | 57.4 |
| MoCo V2 [12] | ResNet-50 | 67.5 | 200 | 58.5 |
| SEED [33] | ResNet-152 | 74.2 | 200 | 62.7 |
| DISCO [35] | ResNet-50 | 67.5 | 200 | 62.5 |
| DISCO [35] | ResNet-152 | 74.1 | 200 | 68.1 |
| BINGO [90] | ResNet-50 | 67.5 | 200 | 63.5 |
| BINGO [90] | ResNet-152 | 74.1 | 200 | 69.1 |
| ReSSL (1 Crop) | - | - | 200 | 66.3 |
| ReSSL (2 Crops) | - | - | 200 | 68.4 |
| ReSSL (Multi) | - | - | 200 | 71.0 |
| EfficientNet-B0 | ResNet-50 | 67.5 | 200 | 46.8 |
| MoCo V2 [12] | ResNet-50 | 67.5 | 200 | 61.3 |
| SEED [33] | ResNet-152 | 74.2 | 200 | 65.3 |
| DISCO [35] | ResNet-50 | 67.5 | 200 | 66.5 |
| DISCO [35] | ResNet-152 | 74.1 | 200 | 67.8 |
| OSS [16] | ResNet-50 | 75.3 | 200 | 64.1 |
| ReSSL (1 Crop) | - | - | 200 | 69.7 |
| ReSSL (2 Crops) | - | - | 200 | 70.9 |
| ReSSL (Multi) | - | - | 200 | 72.2 |
| EfficientNet-B1 | ResNet-50 | 67.5 | 200 | 48.4 |
| MoCo V2 [12] | ResNet-50 | 67.5 | 200 | 61.4 |
| SEED [33] | ResNet-152 | 74.2 | 200 | 67.3 |
| DISCO [35] | ResNet-50 | 67.5 | 200 | 66.6 |
| DISCO [35] | ResNet-152 | 74.1 | 200 | 73.1 |
| ReSSL (1 Crop) | - | - | 200 | 71.1 |
| ReSSL (2 Crops) | - | - | 200 | 73.1 |
| ReSSL (Multi) | - | - | 200 | 74.6 |
| MobileNet-V3-Large | ResNet-50 | 67.5 | 200 | 36.2 |
| MoCo V2 [12] | ResNet-50 | 67.5 | 200 | 55.2 |
| SEED [33] | ResNet-152 | 74.2 | 200 | 61.4 |
| DISCO [35] | ResNet-50 | 67.5 | 200 | 64.4 |
| DISCO [35] | ResNet-152 | 74.1 | 200 | 63.7 |
| ReSSL (1 Crop) | - | - | 200 | 64.7 |
| ReSSL (2 Crops) | - | - | 200 | 66.6 |
| ReSSL (Multi) | - | - | 200 | 68.2 |

The numbers for SimCLR, MoCo V2, SwAV, and BYOL on ResNet-18 are copied from [16].

### TABLE XVII
SEMI-SUPERVISED LEARNING FOR LIGHTWEIGHT ARCHITECTURES

| Method | Teacher Arch | Teacher Top-1 | 1% | 10% |
|--------|---------------|---------------|----|-----|
| ResNet-18 | - | - | 28.8 | 54.2 |
| SimCLR [10] | - | - | 25.2 | 54.1 |
| MoCo V2 [12] | - | - | 25.6 | 52.3 |
| BYOL [38] | - | - | 39.7 | 60.4 |
| SwAV [8] | - | - | 67.5 | 41.2 |
| SEED [33] | ResNet-50 | 67.5 | 39.1 | 50.2 |
| DISCO [35] | ResNet-50 | 67.5 | 39.2 | 50.1 |
| BINGO [90] | ResNet-50 | 67.5 | 42.8 | 57.5 |
| ReSSL (Multi) | - | - | 45.6 | 62.3 |
| ResNet-34 | - | - | 55.4 | 67.8 |
| EfficientNet-B0 | ResNet-50 | 67.5 | 37.7 | 53.3 |
| SEED [33] | ResNet-152 | 74.1 | 41.8 | 56.7 |
| DISCO [35] | ResNet-50 | 67.5 | 42.1 | 56.6 |
| DISCO [35] | ResNet-152 | 74.1 | 52.0 | 63.1 |
| ReSSL (Multi) | - | - | 55.6 | 68.2 |
| EfficientNet-B1 | ResNet-50 | 67.5 | 36.0 | 51.7 |
| MobileNet-V3-Large | ResNet-50 | 67.5 | 40.0 | 52.8 |
| SEED [33] | ResNet-152 | 74.1 | 38.8 | 51.3 |
| DISCO [35] | ResNet-50 | 67.5 | 42.9 | 54.4 |
| ReSSL (Multi) | - | - | 50.9 | 62.8 |

### TABLE XVIII
EFFECT OF INFONCE WARM-UP AND PREDICTOR

| InfoNCE Warm-up | Predictor | Top-1 | Improvements |
|-----------------|-----------|------|--------------|
| Baseline | - | 68.0 | - |
| ✓ | 68.4 | +0.4 |
| ✓ | 69.6 | +1.6 |
| ✓ | 70.3 | +2.3 |

### TABLE XIX
EFFECT OF TEMPERATURE WARM-UP

| Baseline | 0.03 → 0.06 | 0.04 → 0.07 |
|----------|-------------|-------------|
| Top-1    | 70.3        | 69.4        | 69.7        |

### TABLE XX
TEACHER PERFORMANCE VS. STUDENT PERFORMANCE

| Teacher | 70.3 | 69.8 |
|---------|------|------|
| VIT-Small | 73.8 | 73.8 |
a Layer Norm layer. This modification addresses the observed discrepancy, resulting in equivalent performance between the teacher and student models. Consequently, the student model will serve as the standard configuration for this research to ensure consistency.

VI. 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] S. A. Koohpayegani, A. Tejankar, and H. Pirsiavash, “Compress: Self-supervised learning by compressing representations,” Adv. Neural Inf. Process. Syst., vol. 33, pp. 12980–12992, 2020.
[2] S. Arora, H. Khandeparkar, M. Khodak, O. Plevrakis, and N. Saunshi, “A theoretical analysis of contrastive unsupervised representation learning,” 2019, arXiv: 1902.09229.
[3] R. Bachmann, D. Mizrahi, A. Atanov, and A. Zamir, “MultiMAE: Multimodal multi-task masked autoencoders,” in Proc. Eur. Conf. Comput. Vis., 2022, pp. 548–567.
[4] P. Baldi, “Autoencoders, unsupervised learning and deep architectures,” in Proc. Int. Conf. Unsupervised Transfer Learn. Workshop, 2011, pp. 37–50.
[5] L. Bossard, M. Guillaumin, and L. Van Gool, “Food-101–mining discriminative components with random forests,” in Proc. Eur. Conf. Comput. Vis., Springer, 2014, pp. 446–461.
[6] A. Brock, J. Donahue, and K. Simonyan, “Large scale GAN training for high fidelity natural image synthesis,” 2019, arXiv:1809.11096.
[7] M. Caron, P. Bojanowski, A. Joulin, and M. Douze, “Deep clustering for unsupervised learning of visual features,” in Proc. Eur. Conf. Comput. Vis., 2018, pp. 132–149.
[8] M. Caron, I. Misra, J. Mairal, P. Goyal, P. Bojanowski, and A. Joulin, “Unsupervised learning of visual features by contrasting cluster assignments,” in Proc. Int. Conf. Neural Inf. Process. Syst., 2020, pp. 9912–9924.
[9] M. Caron et al., “Emerging properties in self-supervised vision transformers,” 2021, arXiv:2104.14294.
[10] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton, “A simple framework for contrastive learning of visual representations,” 2020, arXiv:2002.05709.
[11] T. Chen, S. Kornblith, K. Swersky, M. Norouzi, and G. Hinton, “Big self-supervised models are strong semi-supervised learners,” 2020, arXiv:2006.10029.
[12] X. Chen, H. Fan, R. Girshick, and K. He, “Improved baselines with momentum contrastive learning,” 2020, arXiv:2003.04297.
[13] X. Chen and K. He, “Exploring simple siamese representation learning,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit., 2021, pp. 15750–15758.
[14] X. Chen, S. Xie, and K. He, “An empirical study of training self-supervised visual transformers,” 2021, arXiv:2104.02057.
[15] Y. Chen, N. Wang, and Z. Zhang, “DarkRank: Accelerating deep metric learning via cross sample similarities transfer,” in Proc. AAAI Conf. Artif. Intell., 2018, pp. 2852–2859.
[16] H. M. Choi, H. Kang, and D. Oh, “Unsupervised representation transfer for small networks: I believe i can distill on-the-fly,” Adv. Neural Inf. Process. Syst., vol. 34, pp. 24645–24658, 2021.
C. Tao et al., “Exploring the equivalence of Siamese self-supervised learning via a unified gradient framework,” 2021, arXiv:2112.05141.

A. Tejankar, S. A. Koohpayegani, V. Pillai, P. Favaro, and H. Pirsiavash, “ISD: Self-supervised learning by iterative similarity distillation,” in Proc. IEEE/CVF Int. Conf. Comput. Vis., 2021, pp. 9609–9618.

Y. Tian, D. Krishnan, and P. Isola, “Contrastive multiview coding,” 2019, arXiv:1910.03672.

Y. Tian, C. Sun, B. Poole, D. Krishnan, C. Schmid, and P. Isola, “What makes for good views for contrastive learning?” 2020, arXiv:2005.10243.

H. Touvron, M. Cord, M. Douze, F. Massa, A. Sablayrolles, and H. Jegou, “Training data-efficient image transformers distillation through knowledge evaporation,” in Proc. Int. Conf. Mach. Learn., 2021, pp. 10347–10357.

F. Tong and G. Mori, “Similarity-preserving knowledge distillation,” in Proc. IEEE/CVF Int. Conf. Comput. Vis., 2019, pp. 1365–1374.

L. Van der Maaten and G. Hinton, “Visualizing data using t-SNE,” J. Mach. Learn. Res., vol. 9, no. 11, pp. 2579–2605, 2008.

F. Wang, X. Xiang, J. Cheng, and A. Yuille, “NormFace: L2 hypersphere embedding for face verification,” in Proc. 25th ACM Int. Conf. Multimodal Comm., 2017, pp. 1041–1049.

G. Wang, K. Wang, G. Wang, P. H. S. Torr, and L. Lin, “Solving inefficiency of self-supervised representation learning,” in Proc. IEEE/CVF Int. Conf. Comput. Vis., 2019, pp. 9505–9515.

H. Wang et al., “CosFace: Large margin cosine loss for deep face recognition,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit., 2018, pp. 5256–5264.

T. Wang and P. Isola, “Understanding contrastive representation learning through alignment and uniformity on the hypersphere,” 2020, arXiv:2005.10242.

X. Wang and G.-J. Qi, “Contrastive learning with stronger augmentations,” 2021, arXiv:2107.01271.

X. Wang, Z. Liu, and S. X. Yu, “Unsupervised feature learning by cross-level instance-group discrimination,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit., 2021, pp. 12586–12595.

P. Weinzaepfel et al., “CroCo v2: Improved cross-view completion pre-training for stereo matching and optical flow,” in Proc. Int. Conf. Comput. Vis., 2023, pp. 17969–17980.

P. Weinzaepfel et al., “CroCo: Self-supervised pre-training for 3D vision tasks by cross-view completion,” in Proc. Int. Conf. Neural Inf. Process. Syst., 2022, pp. 3502–3516.

Y. Wu, A. Kirillov, F. Massa, W.-Y. Lo, and R. Girshick, “Detectron2,” 2019. [Online]. Available: https://github.com/facebookresearch/detectron2

Z. Wu, Y. Xiong, X. Y. Stella, and D. Lin, “Unsupervised feature learning via non-parametric instance discrimination,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2018, pp. 3733–3742.

Q. Xie, Z. Dai, E. Hovy, T. Luong, and Q. Le, “Unsupervised data augmentation for consistency training,” Adv. Neural Inf. Process. Syst., vol. 33, pp. 6256–6268, 2020.

Y. Yu et al., “Bag of Tricks: COCO aggregation boosts self-supervised distillation,” in Proc. Int. Conf. Learn. Representations, 2022, pp. 1–8.

Y. M. Asano, C. Rupprecht, and A. Vedaldi, “Self-labelling via simultaneous clustering and representation learning,” in Proc. Int. Conf. Learn. Representations, 2020, pp. 1–8.

S. You, C. Xu, C. Xu, and D. Tao, “Learning from multiple teacher networks,” in Proc. 23rd ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining, 2017, pp. 1235–1249.

Y. You, I. Gitman, and B. Ginsburg, “Large batch training of convolutional networks,” 2017, arXiv:1708.03888.

L. Yu, V. O. Yacizi, X. Liu, J. van de Weijer, Y. Cheng, and A. Ramisa, “Learning metrics from teachers: Compact networks for image embedding,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit., 2019, pp. 2907–2916.

A. R. Zamir, A. Sax, W. B. Shen, L. Guibas, J. Malik, and S. Savarese, “Taskonomy: Disentangling task transfer learning,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2018, pp. 3712–3722.

J. Zbontar, L. Jing, I. Misra, Y. LeCun, and S. Deny, “Barlow twins: Self-supervised learning via redundancy reduction,” in Proc. Int. Conf. Mach. Learn., PMLR, 2021, pp. 12310–12320.

X. Zhai, A. Oliver, A. Kolesnikov, and L. Beyer, “S4L: Self-supervised semi-supervised learning,” in Proc. IEEE/CVF Int. Conf. Comput. Vis., 2019, pp. 1476–1485.

R. Zhang, P. Isola, and A. A. Efros, “Colorful image colorization,” in Proc. Eur. Conf. Comput. Vis., 2016, pp. 649–666.

S. Zhang, J. Yan, and X. Yang, “Self-supervised representation learning via adaptive hard-positive mining,” 2021.
[100] M. Zheng et al., “Weakly supervised contrastive learning,” in Proc. IEEE/CVF Int. Conf. Comput. Vis., 2021, pp. 10042–10051.

[101] M. Zheng et al., “Relational self-supervised learning with weak augmentation,” 2021, arXiv:2107.09282.

[102] C. Zhuang, A. L. Zhai, and D. Yamins, “Local aggregation for unsupervised learning of visual embeddings,” in Proc. IEEE/CVF Int. Conf. Comput. Vis., 2019, pp. 6002–6012.

Mingkai Zheng received the BSc degree in computer science and technology from the University of Sydney (USYD), Australia, in 2019, where he is currently working toward the PhD degree. His research interests include pattern recognition and machine learning fundamentals with a focus on contrastive learning, self-supervised learning, unsupervised learning, and semi-supervised learning.

Shan You received the PhD degree of computer science from Peking University. He is currently a senior research manager with SenseTime. Before that, he finished a postdoc with Tsinghua University. His research interests include fundamental algorithms for machine learning and computer vision, such as multimodal learning, AutoML, representation learning, and human-centric learning. He has published his research outcomes in many top tier conferences and transactions.

Fei Wang received the bachelor’s and master’s degrees from the Beijing University of Posts and Telecommunications. Currently, he is working toward the PhD degree with the University of Science and Technology of China. He is the director of SenseTime Intelligent Automotive Group. He is the head of SenseAuto-Parking engineering and SenseAuto-Cabin research. He leads a vibrant team of 60+ people to develop comprehensive solutions for the intelligent vehicle and deliver 20+ mass production of SenseAuto-Cabin projects in the last 6 years. He has published 20+ papers at CVPR/NIPS/ICCV during the last few years. His research interests include automotive drive system, AI chip, deep learning, etc.

Chang Xu (Senior Member, IEEE) received the PhD degree from Peking University, China. He is an associate professor and ARC Future fellow with the School of Computer Science, University of Sydney. His research interests lie in machine learning algorithms and related applications in computer vision. He has published more than 200 papers in prestigious journals and top tier conferences. He has received several paper awards, including Distinguished Paper Award in AAAI 2023 and Distinguished Paper Award in IJCAI 2018. He served as an area chair of NeurIPS, ICML, ICLR, KDD, CVPR, and MM, as well as a senior PC member of AAAI and IJCAI. In addition, he served as an associate editor for IEEE Transactions on Pattern Analysis and Machine Intelligence, IEEE Transactions on Multimedia, and Transactions on Machine Learning Research. He has been named a Top Ten Distinguished Senior PC Member in IJCAI 2017 and an Outstanding associate editor for IEEE Transactions on Multimedia in 2022.

Changshui Zhang (Fellow, IEEE) received the BSc degree in mathematics from Peking University, Beijing, China, in 1986, and the PhD degree from Tsinghua University, Beijing, in 1992. In 1992, he joined the Department of Automation, Tsinghua University, where he is currently a professor. His interests include pattern recognition, machine learning, etc. He has authored more than 200 papers. He currently serves on the editorial board of the journal Pattern Recognition.

Xiaogang Wang received the BS degree from the University of Science and Technology of China, in 2001, the MS degree from The Chinese University of Hong Kong, in 2003, and the PhD degree from the Computer Science and Artificial Intelligence Laboratory, Massachusetts Institute of Technology, in 2009. He is currently an associate professor with the Department of Electronic Engineering, The Chinese University of Hong Kong. His research interests include computer vision and machine learning.

Chen Qian received the BEng degree from the Institute for Interdisciplinary Information Science, Tsinghua University, in 2012, and the MPphi degree from the Department of Information Engineering, the Chinese University of Hong Kong, in 2014. He is currently working with SenseTime as research director. His research interests include human-related computer vision and machine learning problems.