Weakly Supervised Contrastive Learning

Mingkai Zheng[1*] Fei Wang[2*] Shan You[1,3†] Chen Qian[1]
Changshui Zhang[3] Xiaogang Wang[1,4] Chang Xu[5]
[1]SenseTime Research [2]University of Science and Technology of China
[3]Department of Automation, Tsinghua University,
Institute for Artificial Intelligence, Tsinghua University (THUAI),
Beijing National Research Center for Information Science and Technology (BNRist)
[4]The Chinese University of Hong Kong
[5]School of Computer Science, Faculty of Engineering, The University of Sydney
{zhengmingkai,youshan,qianchen}@sensetime.com, wangfei91@mail.ustc.edu.cn
zcs@mail.tsinghua.edu.cn, xgwang@ee.cuhk.edu.hk, c.xu@sydney.edu.au

Abstract

Unsupervised visual representation learning has gained much attention from the computer vision community because of the recent achievement of contrastive learning. Most of the existing contrastive learning frameworks adopt the instance discrimination as the pretext task, which treating every single instance as a different class. However, such method will inevitably cause class collision problems, which hurts the quality of the learned representation. Motivated by this observation, we introduced a weakly supervised contrastive learning framework (WCL) to tackle this issue. Specifically, our proposed framework is based on two projection heads, one of which will perform the regular instance discrimination task. The other head will use a graph-based method to explore similar samples and generate a weak label, then perform a supervised contrastive learning task based on the weak label to pull the similar images closer. We further introduced a K-Nearest Neighbor based multi-crop strategy to expand the number of positive samples. Extensive experimental results demonstrate WCL improves the quality of self-supervised representations across different datasets. Notably, we get a new state-of-the-art result for semi-supervised learning. With only 1% and 10% labeled examples, WCL achieves 65% and 72% ImageNet Top-1 Accuracy using ResNet50, which is even higher than SimCLRv2 with ResNet101.

1. Introduction

Modern deep convolutional neural networks demonstrate outstanding performance on various computer vision datasets [11, 15, 30] and edge devices [45, 36, 44, 35]. However, most successful methods are trained in the supervised fashion; they usually require a large volume of labeled data that is very hard to collect. Meanwhile, the quality of data annotations dramatically affects the performance. Recently, self-supervised learning shows its superiority and achieves promising results for unsupervised and semi-supervised learning in computer vision (*e.g.* [6, 7, 19, 8, 9, 5, 18, 50]). These methods can learn general-purpose visual representations without labels and have a good performance on linear classification and transferability to different tasks or datasets. Notably, a big part of the recent self-supervised representation learning framework is based on the idea of contrastive learning.

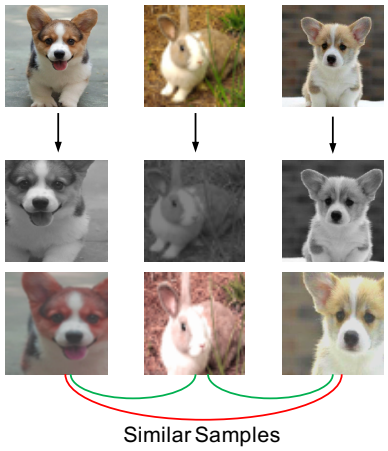

Figure 1. A example of the class collision problem. A typical instance discrimination method will treats the first column and the third column as a negative pair since there are different instance. However, the semantic information of the first column and the third column are very similar, treat them as positive pair should be much more reasonable.

*Equal contributions.
†Corresponding author.

A typical contrastive learning based method adopts the noise contrastive estimation (NCE) [27] to perform the non-parametric instance discrimination [41] as the pretext task, which encourages the two augmented views of the same image to be pulled closer on the embedding space but pushes apart all the other images. Most of the recent works mainly improve the performance of contrastive learning from the image augmentation for positive samples and the exploration for negative samples. However, instance discrimination based methods will inevitably induce class collision problem, which means even for very similar instances, they still need to be pushed apart, as shown in Figure 1. This instance similarities thus tend to hurt the representation quality [1]. In this way, identifying and even leveraging these similar instances plays a key role in the performance of learned representations.

Surprisingly, the class collision problem seems to attract much lesser attention in contrastive learning. As far as we know, there has been little effort to identify similar samples. AdpCLR [49] finds the top-K closest samples on the embedding space and treats these samples as their positives. However, in the early stage of training, the model cannot effectively extract the semantic information from the images; therefore, this method needs to use SimCLR [6] to pre-train for a period of time, and then switch to AdpCLR to get the best performance. FNCancel [23] proposed a similar idea but adopts a very different way to find the top-K similar instances; that is, for each sample, it generates a support set that contains different augmented views from the same image, then use mean or max aggregation strategy over the cosine similarity score between the augmented views in support set and finally identify the top-K similar samples. Nevertheless, the optimal support size is 8 in their experiments, requiring 8 additional forwarding passes to generate the embedding vectors. Obviously, these methods have two shortcomings. Firstly, they are both time-consuming. In the second place, the result of top-K closest samples might not be reciprocal, *i.e.* $\mathbf{x}_i$ is the K closest sample of $\mathbf{x}_j$, but $\mathbf{x}_j$ might not be the K closest sample of $\mathbf{x}_i$. In this case, $\mathbf{x}_j$ will treat $\mathbf{x}_i$ as a positive sample, but $\mathbf{x}_i$ will treat $\mathbf{x}_j$ as a negative sample, which will result in some conflicts.

In this paper, we regard the instance similarities as intrinsically weak supervision in representation learning and propose a weakly supervised contrastive learning framework (WCL) to address the class collision issue accordingly. In WCL, similar instances are assumed to share the same weak label comparing to other instances, and instances with the same weak label are expected to be aggregated. To determine the weak label, we model each batch of instances as a nearest neighbor graph; weak labels are thus determined and reciprocal for each connected component of the graph. Besides, we can further expand the graph by a KNN-based multi-crop strategy to propagate weak labels, such that we can have more positives for each weak label. In this way, similar instances with the same weak label can be pulled closer via the supervised contrastive learning [25] task. Nevertheless, since the mined instance similarities might be noisy and not completely reliable, in practice, we adopt a two-head framework, one of which handles this weakly supervised task while the other is to perform the regular instance discrimination task. Extensive experiments demonstrate the effectiveness of our proposed method across different settings and various datasets.

Our contribution can be summarized as follows:

- We proposed a two-head based framework to address the class collision problem, with one head focusing on the instance discrimination and the other head for attracting similar samples.
- We proposed a simple graph based and parameter-free method to find similar samples adaptively.
- We introduced a K-Nearest Neighbor based multi-crops strategy that can provide much more diverse information than the standard multi-crops strategy.
- The experimental result shows WCL establishes a new state-of-the-art performance for contrastive learning based methods. With only 1% and 10% labeled samples, WCL achieves 65% and 72% Top-1 accuracy on ImageNet using ResNet50. Notably, this result is even higher than SimCLRv2 with ResNet101.

2. Related Work

Self-Supervised Learning. Early work in self-supervised learning mainly focuses on the designing of different pretext tasks. For example, predict a relative offset for a pair of patches [12], solving the jigsaw puzzles [33], colorize the gay-scaled images [48], image inpainting [14], predicting the rotation angle [16], unsupervised deep clustering [4] and image reconstruction [2, 17, 13, 3, 28]. Although these methods have shown their effectiveness, they lack the generality of the learned representations.

Contrastive Learning. Contrastive learning [27, 21, 41, 40] has become one of the most successful methods in the field of self-supervised learning. As we mentioned, most recent works mainly focus on the augmentation for positive samples and the exploration for negative samples. For example, SimCLR [6] proposed composition of data augmentations *e.g.* Grayscale, Random Resized Cropping, Color Jittering, and Gaussian Blur to making the model robust to these transformations. InfoMin [37] further introduced an "InfoMin principle" which suggests that a good augmentation strategy should reduce the mutual information between the positive pairs while keeping the downstream task-relevant information intact. To explore the use of negative samples, InstDisc [41] proposed a memory bank store

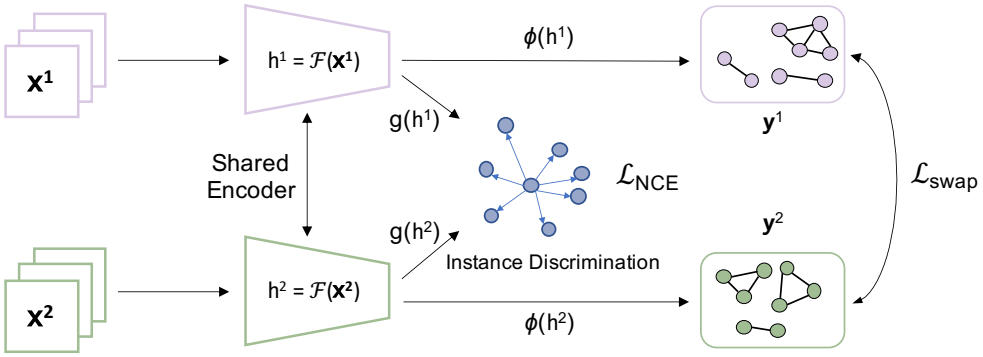

Figure 2. The overall framework of our proposed method. We adopt a two head based structure (g and ϕ). The first head g will play a regular instance discrimination task. The second head ϕ will generate a weak label based on the connected component labeling process, then use the weak label to perform a supervised contrastive learning task. Please see more details in section 3.

the representation of all the images in the dataset. MoCo [19, 8] increasing the number of negatives by using a momentum contrast mechanism that forces the query encoder to learn the representation from a slowly progressing key encoder and maintains a long queue to provide a large number of negative examples.

Contrastive Learning Without Negatives. Unlike the typical contrastive learning framework, BYOL [18] can learn a high-quality visual representation without the negative samples. 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 [9] extends BYOL to explore the siamese structure in contrastive learning further. Surprisingly, SimSiam prevents the model collapse even without the target network and large batch size; although the linear evaluation result is lower than BYOL, it performs better in the downstream tasks.

3. Method

In this section, we will first revisit the preliminary work on contrastive learning and address its limitations. Then we will introduce our proposed weakly supervised contrastive learning framework (WCL), which automatically mines similar samples while doing the instance discrimination. After that, the algorithm and the implementation details will also be explained.

3.1. Revisiting Contrastive Learning

Typical contrastive learning methods adopt the noise contrastive estimation (NCE) objective for discriminating different instance in the dataset. Concretely, NCE objective encourages different augmentations of the same instance to be pulled closer in a latent space yet pushes away different instances' augmentations. Following the setup of SimCLR [6], given a batch of unlabeled samples $\{\mathbf{x}\}_{i=1}^{N}$, we randomly apply a composition of augmentation functions $T(\cdot)$ to obtain two different views of the same instance, which can be written as $\{\mathbf{x}^1\}_{i=1}^{N} = T(\mathbf{x}, \theta_1)$ and $\{\mathbf{x}^2\}_{i=1}^{N} = T(\mathbf{x}, \theta_2)$ where θ is random seed for T. Then, a convolutional neural network based encoder $\mathcal{F}(\cdot)$ will extract the information from different augmentations, that can be expressed by $\{\mathbf{h}^1\} = \mathcal{F}(\{\mathbf{x}^1\}_{i=1}^{N})$ and $\{\mathbf{h}^2\} = \mathcal{F}(\{\mathbf{x}^2\}_{i=1}^{N})$. Finally, a non-linear projection head $\mathbf{z} = g(\mathbf{h})$ maps representations $\mathbf{h}$ to the space where the NCE objective is applied. If we denote $(\mathbf{z}_i, \mathbf{z}_j)$ as a positive pair, the NCE objective can be expressed as

$$\mathcal{L}_{NCE} = -\log \frac{\exp(sim(\mathbf{z}_i, \mathbf{z}_j)/\tau)}{\sum_{k=1}^{N} \mathbb{1}_{[k \neq i]} \exp(sim(\mathbf{z}_i, \mathbf{z}_k)/\tau)}. \quad (1)$$

3.2. Instance Similarities as Weak Supervision

The instance discrimination based methods have already shown promising performance for unsupervised pretraining. However, this line of solution ignores the relationships between different images because only the augmentations from the same image will be regarded as the same class. Inspired by previous works, we can leverage the embedding vectors to explore the relations between different images. Specifically, we will generate a weak label based on the embedding vectors and then use it as a supervisory signal to attract similar samples in the embedding space. However, direct use of weak supervision will cause two problems. First, there is a natural conflict between "instance discrimination" and "similar sample attraction" since one wants to push all the different instances away, and the other wants to pull similar samples closer. Second, there might be noise in the weak label, especially in the early training stages. Simply attracting similar samples based on the weak label will slow down the convergence of the model.

Two-head framework. To resolve these issues, we proposed an auxiliary projection head $\phi(\cdot)$. In this case, the primary projection head $g(\cdot)$ will still perform a regular instance discrimination task to focus on the instance level information; the auxiliary projection head consists of the

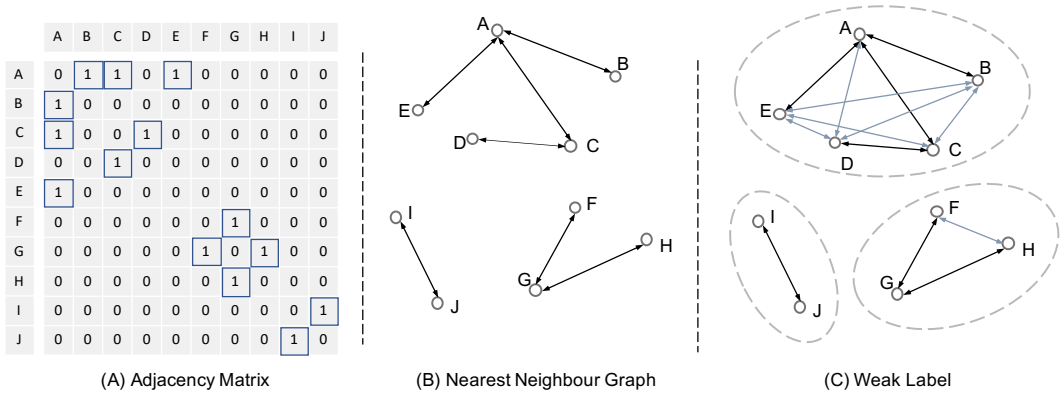

Figure 3. The procedure of weak label generation.

same structure with $g(\cdot)$ and will explore the similar samples and generate a weak label as the supervisory signal to attract similar samples. With these two heads of distinct responsibilities, we can further transform the features extracted by the encoder $\mathcal{F}$ into different embedding spaces to resolve the conflict. Moreover, the primary projection head will ensure the model's convergence even when the weak label has some noise. The information extracted from the auxiliary projection head can be written as

$$\mathbf{v}_i = \phi(\mathcal{F}(T(\mathbf{x}_i, \theta))). \tag{2}$$

Suppose we have obtained a weak label $\mathbf{y} \in \mathbb{R}^{N\times N}$ based on $\mathbf{v}$ which denotes whether a pair of samples is similar (*i.e.* $\mathbf{y}_{ij} = 1$ means $\mathbf{x}_i$ and $\mathbf{x}_j$ are similar). Different from Eq. (1) that naturally forms positive pairs through augmentations, we can then leverage the label $\mathbf{y}_{ij}$ to indicate whether $\mathbf{x}_i$ and $\mathbf{x}_j$ can produce a positive pair or not. By introducing an indicator $\mathbb{1}_{\mathbf{y}_{ij}=1}$ into Eq. (1), we achieve the supervised contrastive loss [25]

$$\mathcal{L}_{sup} = \frac{1}{N}\sum_{i=0}^{N}\mathcal{L}_{sup}^{i} \tag{3}$$

$$\mathcal{L}_{sup}^{i} = -\sum_{j}^{N}\mathbb{1}_{\mathbf{y}_{ij}=1}\log\frac{\exp(sim(\mathbf{v}_i,\mathbf{v}_j)/\tau)}{\sum_{k=1}^{N}\mathbb{1}_{[k\neq i]}\exp(sim(\mathbf{v}_i,\mathbf{v}_k)/\tau)}, \tag{4}$$

which has been shown to be more effective than the traditional supervised cross-entropy loss.

3.3. Weak Label Generation

In this section, we will elaborate how to generate the weak label for the mini-batch of samples. The overall idea can be summarized into two points: First, for each sample, the closest sample can be regarded as a similar sample. Second, if $(\mathbf{x}_i, \mathbf{x}_j)$ and $(\mathbf{x}_j, \mathbf{x}_k)$ are two pairs of similar samples, then we can think that $\mathbf{x}_i$ and $\mathbf{x}_k$ are also similar.

Suppose we use the auxiliary projection head ϕ to map a batch of samples to N embeddings $V = \{\mathbf{v}_1, \mathbf{v}_2, ..., \mathbf{v}_N\}$. Then, for each sample $\mathbf{v}_i$, we find the closest sample $\mathbf{v}_j$ by computing the cosine similarity score. Now, we can define an adjacency matrix by:

$$A(i,j) = \begin{cases} 1, & \text{if } i = k_j^1 \text{ or } j = k_i^1 \\ 0, & \text{otherwise} \end{cases} \tag{5}$$

Here, we use k_i^1 to denote the 1-nearest neighbour of $\mathbf{v}_i$. Basically, Eq.(5) will generate a sparse and symmetric 1-nearest neighbor graph where each vertex is linked with its closest sample. To find out all similar samples, we can convert this problem into a Connected Components Labeling (CCL) process; that is, for each sample, we want to find all the reachable samples based on the 1-nearest neighbor graph. This is a traditional graph problem that can be easily solved by the famous Hoshen–Kopelman algorithm [22] (also known as the two-pass algorithm). We define an undirected graph by $G = (V, E)$ where V is the embedding from ϕ, and edges E connecting the vertex $A(i,j) = 1$. The algorithm adopts a Disjoint-set data structure that consists of three operations: **makeSet**, **union** and **find** (see the definition in Algorithm 1). Basically, it first creates a singleton set for each $\mathbf{v}$ in V, then traverses each edge in E and merges different sets through the edges; finally, it returns the set for each vertex that belongs to. Back to our proposed idea, we will treat the samples in the same set as similar samples. Now, the weak label can be defined as:

$$\mathbf{y}_{ij} = \begin{cases} 1, & \text{if find}(\mathbf{v}_i) = \text{find}(\mathbf{v}_j) \text{ and } i \neq j \\ 0, & \text{otherwise} \end{cases} \tag{6}$$

Such weak label generation method has several advantages.

- This is a parameter-free process, so we do not need any hyperparameter optimization.
- Based on the definition of an undirected graph and connected components, the weak label is always reciprocal. (*i.e.* $\mathbf{y}_{ij} = \mathbf{y}_{ji}$)
- This is a deterministic process; the final result does not depend on any initial state.

Algorithm 1: Connected Components Labeling

Input: An adjacency matrix $G = (V, E)$
Define makeSet(v) : Create a new set with element v
Define union(A, B): Return the set $A \cup B$
Define find(v): Return the set which contains v
for *v in V* **do**
  makeSet(v)
end
for *each* (v_i, v_j) *in E* **do**
  if *find*$(v_i) \neq$ *find*(v_j) **then**
    union(find(v_i), find(v_j))
  end
end
for *each v in V* **do**
  return the set contains v: find(v)
end
Output: The corresponding identification of the connected component for each $\mathbf{v}$.

The weak label will be used as the supervisory signal for the auxiliary projection head ϕ. However, if $\mathbf{v}_i$ and $\mathbf{v}_j$ are in the same set, $sim(\mathbf{v}_i, \mathbf{v}_j)$ is very likely to be a large number. According to Eq. (4), directly using the weak label will cause $\mathcal{L}_{sup}$ to be very small, which is not conducive to the model's optimization. To resolve this issue, we can simply swap the weak label to supervise the same batch of samples with different augmentations. Concretely, we derive embeddings V^1 and V^2 from two types of augmentations, based on which we generate the corresponding weak label $\mathbf{y}^1$, $\mathbf{y}^2$. Then $\mathbf{y}^1$ will be used as the supervisory signal for V^2 and vice versa. The swapped version of Eq. (3) can be written as:

$$\mathcal{L}_{swap} = \mathcal{L}_{sup}(V^1, \mathbf{y}^2) + \mathcal{L}_{sup}(V^2, \mathbf{y}^1). \tag{7}$$

3.4. Label Propagation with Multi-Crops

Since the comparison between random crops of an image plays the key role in contrastive learning, there are lots of previous works [10] pointing out that increasing the number of crops or views can significantly increase the representation quality. SwAV [5] introduced a multi-crop strategy that adds K additional low-resolution crops in each batch. Using low-resolution images can greatly reduce computational costs. However, the multiple crops of the same image may have many overlap areas. In this case, more crops may not provide additional effective information. To address this issue, we proposed a K-Nearest Neighbor based Multi-crops strategy. Specifically, we will store the feature $\mathbf{h}^1$ for every batch and then use these features to find the K closest samples based on the cosine similarity at the end of each epoch. Finally, we will use the low-resolution crops of the K closest images in the next epoch. If we apply the $\mathcal{L}_{swap}$ on the K-NN multi-crops, the number of positive samples can be expended to K times. Note that the K-NN result is unreliable in the early training; hence, we should use the standard multi-crops strategy to warm up the model for a certain number of epochs and then switch to our K-NN multi-crops to get better performance. (See more details in our experiments.) If we use $\mathcal{L}_{cNCE}$ and $\mathcal{L}_{cswap}$ to denote the contrastive loss and swapped loss for the multi-crops images, then the overall training objective for our weakly supervised contrastive learning framework can be expressed as

$$\mathcal{L}_{overall} = \mathcal{L}_{NCE} + \lambda\mathcal{L}_{cNCE} + \beta\mathcal{L}_{swap} + \gamma\mathcal{L}_{cswap}, \tag{8}$$

where λ, β and γ are the hyper-parameters. We simply take $\lambda = 1$, $\beta = 0.5$ and $\gamma = 0.5$ in our implementation. Please see more details in Algorithm 2.

Algorithm 2: Weakly Supervised Contrastive Learning (WCL)

Input: $\{\mathbf{x}^1\}_{i=1}^N$ and $\{\mathbf{x}^2\}_{i=1}^N$: a batch of samples with different augmentations. $\mathcal{F}$: the backbone network. g: the first projection head. ϕ: the auxiliary projection head.
while *network not converge* **do**
  Initialize an empty list L ;
  for *i=1 to step* **do**
    $\mathbf{h}^1 = \mathcal{F}(\{\mathbf{x}^1\}_{i=1}^N)$ $\mathbf{h}^2 = \mathcal{F}(\{\mathbf{x}^2\}_{i=1}^N)$
    $\mathbf{z}^1 = g(\mathbf{h}^1)$ $\mathbf{z}^2 = g(\mathbf{h}^2)$
    $\mathbf{v}^1 = \phi(\mathbf{h}^1)$ $\mathbf{v}^2 = \phi(\mathbf{h}^2)$
    Calculate contrastive loss $\mathcal{L}_{NCE}$ Eq. (1)
    Generate weak label $\mathbf{y}^1$, $\mathbf{y}^2$ based on $\mathbf{v}^1$, $\mathbf{v}^2$
    Calculate swapped loss $\mathcal{L}_{swap}$ Eq. (7)
    Calculate $\mathcal{L}_{cNCE}$ and $\mathcal{L}_{cswap}$
    Optimize the network by $\mathcal{L}_{overall}$ Eq. (8)
    Append $\mathbf{h}^1$ to list L ;
  end
  Compute the K-NN for each sample based on L.
end
Output: The well trained model $\mathcal{F}$

4. Experimental Results

4.1. Ablation Studies

In this section, we will empirically study our Weak Supervised Contrastive Learning (WCL) framework under different batch sizes, epochs, datasets(CIFAR-10, CIFAR-100, ImageNet100) and show the effectiveness of each component by extensive experiments.

CIFAR-10 and CIFAR-100. The CIFAR-10 [26] 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. We use the ResNet50 [20] as our backbone network. Because the training images only contain 32x32 pixels, we replace the first 7x7 Conv of stride 2 with

Table 1. Experiments on CIFAR-10 and CIFAR-100 with different batch size and training epochs

Batch Size	Method	CIFAR10				CIFAR100			
		100 ep	200 ep	300 ep	400 ep	100 ep	200 ep	300 ep	400 ep
64	SimCLR	77.20	80.64	82.77	84.48	52.35	55.86	58.18	59.96
64	WCL (Ours)	79.17 **(+1.97)**	83.54 **(+2.90)**	85.68 **(+2.91)**	86.64 **(+2.16)**	53.54 **(+1.19)**	56.57 **(+0.71)**	59.29 **(+1.11)**	60.76 **(+0.80)**
128	SimCLR	79.64	83.57	85.70	86.72	54.72	59.19	60.88	62.20
128	WCL (Ours)	81.82 **(+2.18)**	85.65 **(+2.08)**	87.81 **(+2.91)**	88.65 **(+1.93)**	55.46 **(+0.74)**	60.30 **(+1.11)**	61.73 **(+0.85)**	63.17 **(+0.97)**
256	SimCLR	81.78	85.34	87.29	88.48	57.16	61.18	63.49	64.20
256	WCL (Ours)	83.12 **(+1.34)**	87.57 **(+2.23)**	88.85 **(+1.56)**	89.47 **(+0.98)**	57.85 **(+0.70)**	62.98 **(+1.80)**	64.21 **(+0.72)**	64.93 **(+0.73)**

3x3 Conv of stride 1 and also remove the first max pooling operation. We use 2-Layer-MLP for the two non-linear projection heads. For data augmentation, we use the random resized crops (the lower bound of random crop ratio is set to 0.2), color distortion (strength=0.5), and leaving out Gaussian blur. The model is trained using LARS optimizer [46] with a momentum of 0.9 and weight decay of $1e^{-6}$. We linear warm up the learning rate for 10 epochs until it reaches $0.25 \times BatchSize/256$, then switch to the cosine decay scheduler [31]. The temperature parameter τ is always set to 0.1. To perform the Connected Components Labeling process, we simply use the "connected_components" function from the Scipy Library [39]. We will use the same training strategy for both CIFAR-10 and CIFAR-100.

ImageNet-100. ImageNet-100 dataset is a randomly chosen subset from ILSVRC2010 ImageNet [11]. (We simply take the first 100 class in our experiments.) For training the ImageNet-100, we strictly follow the training strategy reported in SimCLR [6]. Specifically, we set the $BatchSize = 2048$, and use the LARS optimizer with $lr = 0.075 \times \sqrt{BatchSize}$. Moreover, we found that the default augmentation that used in SimCLR might be too strong, which makes the model very hard to converge in the beginning; thus, we adopt the same but a little bit weaker version of the augmentation (the one that used in MoCoV2[8]) in the first 10 epochs and then switch it back to the original augmentations after warm-up. The model will be optimized for 200 epochs, and the rest of the settings (including temperature, weight decay, etc.) are the same as our CIFAR training.

Evaluation Protocol. For testing the representation quality, we evaluate our well-trained model on the widely adopted linear evaluation protocol - We will freeze the encoder parameters and train a linear classifier on top of it by using the standard SGD optimizer with a momentum of 0.9, learning rate of $0.1 \times BatchSize/256$ and cosine decay scheduler. We don't use any regularization techniques such as weight decay and gradient clipping. The model will be trained for 80 epochs, then evaluated on the testing set.

Effect of weak supervision. We choose SimCLR as our baseline, and compare it with our method on $BatchSize = 64, 128, 256$ and $Epoch = 100, 200, 300, 400$. Note, in these experiments; we do not use any multi-crops strategy; only an additional $\mathcal{L}_{swap}$ is applied on top of the SimCLR. Table 1 shows the results. Obviously, our proposed method substantially outperforms the baseline across all settings. For CIFAR-10, we have various improvements from 0.98% to 2.91% based on different settings. For CIFAR-100, the improvement is from 0.73% to 1.80%.

Table 2. Effectiveness of two-head framework (ImageNet100)

g	ϕ	$\mathcal{L}_{NCE}$	$\mathcal{L}_{swap}$	$\mathcal{L}_{cNCE}$	$\mathcal{L}_{cswap}$	Top-1
✓		✓				75.79
	✓		✓			71.33
✓		✓	✓			75.26
✓	✓	✓	✓			**77.51**
✓	✓	✓	✓	✓		79.06
✓	✓	✓	✓		✓	79.08
✓	✓	✓	✓	✓	✓	**79.77**

Effect of two-head framework. We also perform an extensive ablation study to examine the effectiveness of our two head based framework. The experiments are mainly performed on the ImageNet-100 dataset, and the result is shown in Table 2. Note, the $\mathcal{L}_{cNCE}$ and $\mathcal{L}_{cswap}$ in this experiment is based on the standard multi-crops strategy (without KNN). The first row is the SimCLR baseline. The second row is the case that only $\mathcal{L}_{swap}$ is applied; the model can still learn a meaningful representation but result in a worse accuracy than the baseline. We also try to apply both $\mathcal{L}_{NCE}$ and $\mathcal{L}_{swap}$ on the same head; from the third row, we can see there is a 0.53% performance drop. We doubt this is because of the conflicts between the instance discrimination and similar sample attraction. The fourth row shows our proposed method, which separates the two tasks on different heads. In this case, we get 1.72% improvements over the baseline, which verified our hypothesis. The last three rows show the result with the multi-crops strategy, and the performance can be further improved by 2.26%.

Effect of K-NN Multi-Crops. As we have mentioned, the K-NN result is unreliable in the early training, and we need to use the standard multi-crops strategy to warm up the model for a certain number of epochs. Table 3 shows the result for a different number of warm up epochs. We can see clearly that with 50 epochs of warm up, our K-NN multi-crops strategy has 1% improvements over the standard multi-crops (see the last row in Table 2). Finally, our proposed method achieved 80.78% Top-1 accuracy on linear evaluation, which has 5% improvements than the SimCLR baseline (75.79%).

Table 3. Warm up epochs for K-NN Multi-Crops (K=4)

Epochs	0	25	50	75	100
Accuracy	79.73	80.25	**80.78**	80.63	80.23

Visualization. Figure 4 shows the t-SNE visualization [38] of $\mathbf{h}$ from a randomly selected 10 classes. Compare to SimCLR; our weakly supervised contrastive learning framework can enhance a much better intra-class compactness and inter-class discrepancy.

Figure 4. t-SNE visualization for SimCLR and our method

4.2. Comparison on ImageNet-1K Dataset

We also performed our algorithm on the large-scale ImageNet-1k dataset [11]. The training strategy is the same as our ImageNet-100 training, except we adopt a larger batch size (4096) and use the 3-Layer-MLP for the two projection heads. For the K-NN Multi-crops, we simply take the best strategy from Table 3, which means we will use the standard multi-crops strategy for the first 25% epochs, and then switch to our K-NN version.

Table 4. Compare to FNCancel on ImageNet-1K

Method	Epochs	GPU(time)	Acc
SimCLR	100	1.00	66.4
FNCancel	100	-	68.1
WCL (Ours)	100	1.01	68.1
SimCLR	1000	10.00	70.3
FNCancel + multi-crops	100	2.85	70.4
WCL (Ours) + multi-crops	100	1.31	**71.0**

Compare to FNCancel. [23] Table 4 shows the comparison between our proposed method with FNCancel and SimCLR. Note, for a fair comparison, all models are trained with a 3-Layer-MLP projection head. As we can see, with a negligible additional computational cost (0.01), our proposed method can surpass the SimCLR baseline 1.7% and achieved the same result with FNCancel. FNCancel does not report the standard time usage on the paper, but since it requires 8 additional forward passes to generate the support view embeddings, their actual computational cost will be much higher than ours. We also compare the result with the multi-crops strategy. In this case, we use 2 160×160 images as our main views and 6 additional 96×96 K-NN crops. Look at the last row; our proposed method can achieve 71.0 top-1 accuracy with only 31% more additional cost than SimCLR. This is twice faster than FNCancel and has 0.6% improvements on linear evaluation.

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. * denotes multi-crops strategy.

Method	Arch	Param	Epochs	Top-1
Supervised	R50	24	-	76.5
InstDisc [41]	R50	24	200	58.5
LocalAgg [51]	R50	24	200	58.8
SimCLR [6]	R50	24	200	66.8
MoCo [19]	R50	24	200	60.8
MoCo v2 [8]	R50	24	200	67.5
MoCHi [24]	R50	24	200	68.0
CPC v2 [27]	R50	24	200	63.8
PCL v2 [29]	R50	24	200	67.6
SimSiam [9]	R50	24	200	70.0
SwAV [5]	R50	24	200	69.1
SwAV* [5]	R50	24	200	72.7
WCL (Ours)	R50	24	200	70.3
WCL* (Ours)	R50	24	200	**73.3**

Table 6. Top-1 accuracy under the linear evaluation on ImageNet. The table compares the methods with more epochs of pretraining. * denotes multi-crops strategy.

Method	Arch	Param	Epochs	Top-1
Supervised	R50	24	-	76.5
SeLa [43]	R50	24	400	61.5
SimCLR [6]	R50	24	800	69.1
SimCLR v2 [7]	R50	24	800	71.7
MoCo v2 [8]	R50	24	800	71.1
SimSiam [9]	R50	24	800	71.3
SwAV [5]	R50	24	800	71.8
BYOL [18]	R50	24	1000	74.3
FNCancel* [23]	R50	24	1000	74.4
AdpCLR [49]	R50	24	1100	72.3
WCL (Ours)	R50	24	800	72.2
WCL* (Ours)	R50	24	800	**74.7**
Others				
SwAV* [5]	R50	24	800	**75.3**

Linear Evaluation. For the linear evaluation of ImageNet-1k, we strictly follow the setting in SimCLR [6]. Table 5 and 6 shows our result for 200 epochs and 800 epochs of training. We also report the result with 2 224×224 and 6 additional 96×96 K-NN crops (as in SwAV [5]). We can see clearly that when the model is optimized for 200 epochs, our proposed method achieved state-

Table 7. Low-shot image classification on VOC07

Method	Epochs	k=1	k=2	k=4	k=8	k=16	k=32	k=64	Full
Random	-	8.92	9.33	10.10	10.42	10.82	11.34	11.96	12.42
Supervised	90	54.46	68.15	73.79	79.51	82.26	84.00	85.13	87.27
MoCo v2 [8]	200	46.30	58.40	64.85	72.47	76.14	79.16	81.52	84.60
PCL v2 [29]	200	47.88	59.59	66.21	74.45	78.34	80.72	82.67	85.43
SwAV [5]	200	43.07	55.65	64.82	73.17	78.38	81.86	84.40	87.47
WCL (Ours)	200	**48.06**	**60.12**	**68.52**	**76.16**	**80.24**	**82.97**	**85.01**	**87.75**
SwAV [5]	400	42.14	55.34	64.31	73.08	78.47	82.09	84.62	87.78
SwAV [5]	800	42.85	54.90	64.03	72.94	78.65	82.32	84.90	88.13
WCL (Ours)	800	**48.25**	**60.68**	**68.52**	**76.48**	**81.05**	**83.89**	**85.88**	**88.64**

of-the-art performance among all the recent self-supervised learning frameworks. When the model is trained for 800 epochs, our model can still outperform most recent works but slightly lower than SwAV.

Table 8. ImageNet semi-supervised evaluation.

	1%		10%	
Method	Top-1	Top-5	Top-1	Top-5
Supervised	25.4	56.4	48.4	80.4
Semi-supervised				
S4L [47]	-	53.4	-	83.8
UDA [42]	-	68.8	-	88.5
FixMatch [34]	-	-	71.46	89.1
Self-Supervised				
From AvgPool				
InstDisc [41]	-	39.2	-	77.4
PCL [29]	-	75.6	-	86.2
PIRL [32]	30.7	60.4	57.2	83.8
SimCLR v1 [6]	48.3	75.5	65.6	87.8
BYOL [18]	53.2	78.4	68.8	89.0
SwAV [5]	53.9	78.5	70.2	89.9
WCL (Ours)	**58.3**	**79.9**	**71.1**	**90.3**
From Projection Head				
SimCLR v2 (R50) [7]	57.9	-	68.4	-
SimCLR v2 (R101)[7]	62.1	-	71.4	-
FNCancel [23]	63.7	85.3	71.1	90.2
WCL (Ours)	**65.0**	**86.3**	**72.0**	**91.2**

Semi-Supervised Learning. Next, we evaluate the performance obtained when fine-tuning the model representation using a small subset of labeled data. For a fair comparison, we take the same labeled list from SimCLR [6]. Specifically, we report our results on two different settings. First, we follow the strategy in PCL [29], and fine-tuning from the average pooling layer of the ResNet50 [20] network. In this setting, our model outperforms the previous state-of-the-art (SwAV) 4.4% on 1% labels and 0.9% on 10% labels. Then, we also follow the strategy in SimCLRv2 [7] to fine-tuning from the first layer of the projection head. In this case, our method has 1.3% and 0.9% improvement on 1% and 10% labels over FNCancel. Notably, this result is even higher than the SimCLRv2 with ResNet101 backbone.

Transfer Learning. Finally, We further evaluate the quality of the learned representations by transferring them to other datasets. Following [29, 5], we perform linear classification on the PASCAL VOC2007 dataset [15]. 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). The result of our method and SwAV are both based on the multi-crop version. When the model has 200 epochs of pretraining, our method and SwAV can already outperform the supervised pretraining on the full dataset. Interestingly, our method is significantly better than all other works, especially when k is small. When the model has more pretraining epochs, our method can even surpass the supervised pretraining with k = 64 and consistently has higher performance than SwAV across all different k values.

5. Conclusion

In this work, we proposed a weakly supervised contrastive learning framework that consist of two projection heads, one of which focus on the instance discrimination task, and the other head adopts the Connected Components Labeling process to generate a weak label, then perform the supervised contrastive learning task by swapping the weak label to different augmentations. Finally, we introduced a new K-NN based multi-crops strategy which has much more effective information and expanding the number of positive samples to K times. Experiments on CIFAR-10, CIFAR-100, ImageNet-100 show the effectiveness of each component. The results of semi-supervised learning and transfer learning demonstrate the state-of-the-art performance for unsupervised representation learning.

Acknowledgment

This work is funded by the National Key Research and Development Program of China (No. 2018AAA0100701) and the NSFC 61876095. Chang Xu was supported in part by the Australian Research Council under Projects DE180101438 and DP210101859. Shan You is supported by Beijing Postdoctoral Research Foundation.

References

[1] S. Arora, Hrishikesh Khandeparkar, M. Khodak, Orestis Plevrakis, and Nikunj Saunshi. A theoretical analysis of contrastive unsupervised representation learning. *ArXiv*, abs/1902.09229, 2019. 2

[2] Pierre Baldi. Autoencoders, unsupervised learning and deep architectures. In *Proceedings of the 2011 International Conference on Unsupervised and Transfer Learning Workshop - Volume 27*, UTLW'11, page 37–50. JMLR.org, 2011. 2

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

[4] Mathilde Caron, Piotr Bojanowski, Armand Joulin, and Matthijs Douze. Deep clustering for unsupervised learning of visual features. In *European Conference on Computer Vision*, 2018. 2

[5] Mathilde Caron, Ishan Misra, Julien Mairal, Priya Goyal, Piotr Bojanowski, and Armand Joulin. Unsupervised learning of visual features by contrasting cluster assignments. 2020. 1, 5, 7, 8

[6] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. *arXiv preprint arXiv:2002.05709*, 2020. 1, 2, 3, 6, 7, 8

[7] Ting Chen, Simon Kornblith, Kevin Swersky, Mohammad Norouzi, and Geoffrey Hinton. Big self-supervised models are strong semi-supervised learners. *arXiv preprint arXiv:2006.10029*, 2020. 1, 7, 8

[8] Xinlei Chen, Haoqi Fan, Ross Girshick, and Kaiming He. Improved baselines with momentum contrastive learning. *arXiv preprint arXiv:2003.04297*, 2020. 1, 3, 6, 7, 8

[9] Xinlei Chen and Kaiming He. Exploring simple siamese representation learning, 2020. 1, 3, 7

[10] Ching-Yao Chuang, Joshua Robinson, Lin Yen-Chen, Antonio Torralba, and Stefanie Jegelka. Debiased contrastive learning. In *Advances in neural information processing systems*, 2020. 5

[11] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. ImageNet: A Large-Scale Hierarchical Image Database. In *CVPR09*, 2009. 1, 6, 7

[12] Carl Doersch, Abhinav Gupta, and Alexei A. Efros. Unsupervised visual representation learning by context prediction. In *International Conference on Computer Vision (ICCV)*, 2015. 2

[13] J. Donahue and K. Simonyan. Large scale adversarial representation learning. In *NeurIPS*, 2019. 2

[14] Omar ElHarrouss, Noor Almaadeed, S. Al-Máadeed, and Y. Akbari. Image inpainting: A review. *Neural Processing Letters*, 51:2007–2028, 2019. 2

[15] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The PASCAL Visual Object Classes Challenge 2007 (VOC2007) Results. http://www.pascal-network.org/challenges/VOC/voc2007/workshop/index.html. 1, 8

[16] Spyros Gidaris, Praveer Singh, and Nikos Komodakis. Unsupervised representation learning by predicting image rotations. *ArXiv*, abs/1803.07728, 2018. 2

[17] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In Z. Ghahramani, M. Welling, C. Cortes, N. Lawrence, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems*, volume 27, pages 2672–2680. Curran Associates, Inc., 2014. 2

[18] Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre H. Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Daniel Guo, Mohammad Gheshlaghi Azar, Bilal Piot, Koray Kavukcuoglu, Rémi Munos, and Michal Valko. Bootstrap your own latent: A new approach to self-supervised learning, 2020. 1, 3, 7, 8

[19] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. *arXiv preprint arXiv:1911.05722*, 2019. 1, 3, 7

[20] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *arXiv preprint arXiv:1512.03385*, 2015. 5, 8

[21] R Devon Hjelm, Alex Fedorov, Samuel Lavoie-Marchildon, Karan Grewal, Phil Bachman, Adam Trischler, and Yoshua Bengio. Learning deep representations by mutual information estimation and maximization. *arXiv preprint arXiv:1808.06670*, 2018. 2

[22] J. Hoshen and R. Kopelman. Percolation and cluster distribution. i. cluster multiple labeling technique and critical concentration algorithm. *Phys. Rev. B*, 14:3438–3445, Oct 1976. 4

[23] Tri Huynh, Simon Kornblith, Matthew R. Walter, Michael Maire, and Maryam Khademi. Boosting contrastive self-supervised learning with false negative cancellation. *arXiv:2011.11765*, 11 2020. 2, 7, 8

[24] Yannis Kalantidis, Mert Bulent Sariyildiz, Noe Pion, Philippe Weinzaepfel, and Diane Larlus. Hard negative mixing for contrastive learning. In *Neural Information Processing Systems (NeurIPS)*, 2020. 7

[25] Prannay Khosla, Piotr Teterwak, Chen Wang, Aaron Sarna, Yonglong Tian, Phillip Isola, A. Maschinot, Ce Liu, and Dilip Krishnan. Supervised contrastive learning. *ArXiv*, abs/2004.11362, 2020. 2, 4

[26] A. Krizhevsky. Learning multiple layers of features from tiny images. 2009. 5

[27] Cheng-I Lai. Contrastive predictive coding based feature for automatic speaker verification. *arXiv preprint arXiv:1904.01575*, 2019. 2, 7

[28] C. Ledig, L. Theis, Ferenc Huszár, J. Caballero, Andrew Aitken, Alykhan Tejani, J. Totz, Zehan Wang, and W. Shi. Photo-realistic single image super-resolution using a generative adversarial network. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 105–114, 2017. 2

[29] Junnan Li, Pan Zhou, Caiming Xiong, and Steven Hoi. Prototypical contrastive learning of unsupervised representations. In *International Conference on Learning Representations*, 2021. 7, 8

[30] Tsung-Yi Lin, Michael Maire, Serge Belongie, Lubomir Bourdev, Ross Girshick, James Hays, Pietro Perona, Deva

Ramanan, C. Lawrence Zitnick, and Piotr Dollár. Microsoft coco: Common objects in context, 2014. cite arxiv:1405.0312Comment: 1) updated annotation pipeline description and figures; 2) added new section describing datasets splits; 3) updated author list. 1

[31] Ilya Loshchilov and Frank Hutter. Sgdr: Stochastic gradient descent with warm restarts. *arXiv preprint arXiv:1608.03983*, 2016. 6

[32] Ishan Misra and Laurens van der Maaten. Self-supervised learning of pretext-invariant representations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020. 8

[33] M. Noroozi and P. Favaro. Unsupervised learning of visual representations by solving jigsaw puzzles. In *ECCV*, 2016. 2

[34] Kihyuk Sohn, David Berthelot, Chun-Liang Li, Zizhao Zhang, Nicholas Carlini, Ekin D. Cubuk, Alex Kurakin, Han Zhang, and Colin Raffel. Fixmatch: Simplifying semi-supervised learning with consistency and confidence. *arXiv preprint arXiv:2001.07685*, 2020. 8

[35] Xiu Su, Shan You, Fei Wang, Chen Qian, Changshui Zhang, and Chang Xu. Bcnet: Searching for network width with bilaterally coupled network. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2175–2184, 2021. 1

[36] Xiu Su, Shan You, Mingkai Zheng, Fei Wang, Chen Qian, Changshui Zhang, and Chang Xu. K-shot NAS: learnable weight-sharing for NAS with k-shot supernets. In *ICML*, volume 139 of *Proceedings of Machine Learning Research*, pages 9880–9890. PMLR, 2021. 1

[37] Yonglong Tian, Dilip Krishnan, and Phillip Isola. Contrastive multiview coding. *arXiv preprint arXiv:1906.05849*, 2019. 2

[38] Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-sne, 2008. 7

[39] Pauli Virtanen, Ralf Gommers, Travis E. Oliphant, Matt Haberland, Tyler Reddy, David Cournapeau, Evgeni Burovski, Pearu Peterson, Warren Weckesser, Jonathan Bright, Stéfan J. van der Walt, Matthew Brett, Joshua Wilson, K. Jarrod Millman, Nikolay Mayorov, Andrew R. J. Nelson, Eric Jones, Robert Kern, Eric Larson, C J Carey, İlhan Polat, Yu Feng, Eric W. Moore, Jake VanderPlas, Denis Laxalde, Josef Perktold, Robert Cimrman, Ian Henriksen, E. A. Quintero, Charles R. Harris, Anne M. Archibald, Antônio H. Ribeiro, Fabian Pedregosa, Paul van Mulbregt, and SciPy 1.0 Contributors. SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. *Nature Methods*, 17:261–272, 2020. 6

[40] Tongzhou Wang and Phillip Isola. Understanding contrastive representation learning through alignment and uniformity on the hypersphere. *arXiv preprint arXiv:2005.10242*, 2020. 2

[41] Zhirong Wu, Yuanjun Xiong, X Yu Stella, and Dahua Lin. Unsupervised feature learning via non-parametric instance discrimination. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018. 2, 7, 8

[42] Qizhe Xie, Zihang Dai, Eduard Hovy, Thang Luong, and Quoc Le. Unsupervised data augmentation for consistency training. In H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 6256–6268. Curran Associates, Inc., 2020. 8

[43] Asano YM., Rupprecht C., and Vedaldi A. Self-labelling via simultaneous clustering and representation learning. In *International Conference on Learning Representations*, 2020. 7

[44] Shan You, Tao Huang, Mingmin Yang, Fei Wang, Chen Qian, and Changshui Zhang. Greedynas: Towards fast one-shot nas with greedy supernet. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1999–2008, 2020. 1

[45] Shan You, Chang Xu, Chao Xu, and Dacheng Tao. Learning from multiple teacher networks. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 1285–1294, 2017. 1

[46] Yang You, Igor Gitman, and Boris Ginsburg. Large batch training of convolutional networks, 2017. 6

[47] Xiaohua Zhai, A. Oliver, A. Kolesnikov, and Lucas Beyer. S4l: Self-supervised semi-supervised learning. *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 1476–1485, 2019. 8

[48] Richard Zhang, Phillip Isola, and Alexei A Efros. Colorful image colorization. In *ECCV*, 2016. 2

[49] Shaofeng Zhang, Junchi Yan, and Xiaokang Yang. Self-supervised representation learning via adaptive hard-positive mining, 2021. 2, 7

[50] Mingkai Zheng, Shan You, Fei Wang, Chen Qian, Changshui Zhang, Xiaogang Wang, and Chang Xu. Ressl: Relational self-supervised learning with weak augmentation. *arXiv preprint arXiv:2107.09282*, 2021. 1

[51] Chengxu Zhuang, Alex Lin Zhai, and Daniel Yamins. Local aggregation for unsupervised learning of visual embeddings, 2019. 7