Self-Supervised Representation Learning with Cross-Context Learning between Global and Hypercolumn Features

Zheng Gao Chen Feng Ioannis Patras

Queen Mary University of London, Mile End Road, London, E1 4NS
{z.gao, chen.feng, i.patras}@qmul.ac.uk

Abstract

*Whilst contrastive learning yields powerful representations by matching different augmented views of the same instance, it lacks the ability to capture the similarities between different instances. One popular way to address this limitation is by learning global features (after the global pooling) to capture inter-instance relationships based on knowledge distillation, where the global features of the teacher are used to guide the learning of the global features of the student. Inspired by cross-modality learning, we extend this existing framework that only learns from global features by encouraging the global features and intermediate layer features to learn from each other. This leads to our novel self-supervised framework: **c**ross-context learning between **g**lobal and **h**ypercolumn features (CGH), that enforces the consistency of instance relations between low- and high-level semantics. Specifically, we stack the intermediate feature maps to construct a "hypercolumn" representation so that we can measure instance relations using two contexts (hypercolumn and global feature) separately, and then use the relations of one context to guide the learning of the other. This cross-context learning allows the model to learn from the differences between the two contexts. The experimental results on linear classification and downstream tasks show that our method outperforms the state-of-the-art methods.*

1. Introduction

Representation learning has become a challenging and active topic in computer vision, capable of learning representations that can be transferred to various downstream tasks, such as classification, object detection, segmentation, etc [12, 22, 27, 28, 46]. Due to the capability of leveraging massive amounts of data without requiring annotations, self-supervised representation learning, in particular, has shown the potential to learn representations that generalize well on various downstream tasks.

Contrastive learning has shown promising results in the self-supervised representation learning [14, 16, 19, 42, 44, 48]. Contrastive learning aims to learn invariant representations for different views of the same image instance (augmented views should have similar features while different instances are forced to have dissimilar features). Therefore, it lacks the ability to capture similar semantics shared between different instances [17,49]. As a result, it suffers from the so-called "class collision problem" [5,33]. Recent methods aim to alleviate this limitation by capturing similarity relationships among instances based on the knowledge distillation framework where the student is trained to predict the target similarity distribution from the teacher [15, 49]. The similarity relationships are typically measured with the cosine similarities between the input and the samples in the memory bank, which are normalized with a softmax operation. This leads to a probabilistic distribution where similar instances are emphasized so that the student is trained to produce correlated features for similar samples. However, these methods are limited to use the global features (after the global average pooling) of the teacher to guide the learning of the global features of the student. We term this line of works as "*global-context learning*" in this paper.

Works in cross-modality learning [1,38] have shown that the learning paradigm of one modality can benefit from cross-modal information from multiple modalities. While an additional modality is not available when considering visual-only data, we argue that **whilst features from the intermediate layers and global features from the final layer are correlated, they encode semantics at different levels of abstraction** – the earlier layers capturing lower-level details while the latter layers capturing higher-level semantics. The differences between intermediate layers and global features can facilitate the learning of both compared with global-context learning in current works. Inspired by this, we treat the intermediate layers and global features as two contexts and propose a cross-context learning strategy where these two contexts learn from each other.

More specifically, we construct a "*hypercolumn*" representation [18] by stacking the concatenation of intermediate feature maps as the context of the intermediate layers. Then we measure similarity relationships among instances using two contexts (hypercolumn and global feature) separately, and **use the similarity relationships of one context as supervision to guide the learning of the other**. This leads to a novel self-supervised framework–cross-context learning between global and hypercolumn features (CGH)–that learns representations by capturing cross-context information from global features and hypercolumns.

We highlight our proposed CGH framework degenerates to ReSSL [49] as a special case of global-context learning when the hypercolumn only uses the last layer. The linear classification results on ImageNet show that the proposed CGH outperforms MoCo-v2 [8] and ReSSL [49] by 3.0% and 1.2% respectively with 200 epochs pre-training.

The contributions of this paper can be summarized as follows:

- We address the class collision problem in contrastive learning by capturing similarity relationships among instances with the knowledge distillation framework. In contrast to previous methods that are limited to global feature-based instance relations, we propose a novel cross-context learning scheme in which two contexts, one constructed from intermediate layers (hypercolumn) and one from global features, are used to supervise each other.
- We show that using the proposed hypercolumn-based representations to capture instance relationships is beneficial and leads to learning better global representations. Precision-recall graphs on the similarity distributions show that this leads to significantly higher recall at very similar precision levels (Sec. 4.6).
- Our method is simple and effective. The experimental results on self-supervised benchmarks show that our method achieves superior performance on linear evaluation and downstream tasks compared with state-of-the-art methods, which demonstrates the effectiveness of the cross-context learning strategy.

2. Related work

2.1. Contrastive learning

Contrastive learning based on the Siamese structure aims to learn representations by ensuring positive pairs stay close in the latent space and keeping the negative pairs far away [6–8, 19, 21, 34, 39, 41]. To achieve this, contrastive learning maximizes the correlation between different transformed versions of the same image in the latent space and minimizes that of negative pairs [6]. MoCo [19] performs contrastive learning through a dictionary lookup, whilst SimCLR [6] simplifies MoCo's sampling strategy by generating negative samples from the current batch instead of maintaining a memory bank. However, it can be a challenge to generate meaningful negative samples. Therefore, non-contrastive methods without negative pairs are developed [9, 16] by using techniques like stop-gradient and prediction head to prevent collapsing so that the Siamese network doesn't produce a constant output.

2.2. Deep clustering

Contrastive learning forces every instance to be assigned to a distinct class by pushing different instances apart. By contrast, deep clustering based methods [17, 33] map similar instances to the same class to solve the class collision problem. Typically, deep clustering based methods leverage clustering algorithms like K-Means [36] to assign a pseudo label for each instance so that similar instances can be clustered into the same clustering centroid. DeepCluster [2] uses K-means to generate labels for the samples, which are used as pseudo labels to provide supervisory signals for learning representations. SwAV [3] proposes an online clustering algorithm and enforces the consistency of cluster assignments between different views of the same image. The previous methods only establish a single hierarchy of the images, PCL-v2 [33] discovers the multiple semantic hierarchies of the images and performs instance-wise and instance-cluster contrastive learning to solve the class collision problem. HCSC [17] extends the work of PCL-v2 [33] by selecting high-quality positive and negative pairs based on the similarity between the samples and the centroids. However, most of these works are based on a strong assumption that the labels must induce an equipartition of the data [49].

2.3. Inter-sample relations

Further works try to alleviate class collision by extending the positive sample pair to a set of positive samples [12,31,37]. NNCLR [12] generates an additional positive pair by finding the nearest neighbor of the input image. MSF [31] compares the input image with several nearest neighbors stored in a memory bank. CMSF [37] further generalizes the idea in MSF by refining the search space of nearest neighbors so that the search space is correlated to the query image yet has sufficient variances. A close line of works to this paper aim to solve the class collision problem by capturing instance relations based on self-distillation [29,47]. OBoW [15] trains the student to predict the similarity distribution over the vocabulary generated by the teacher, which is built upon the local views of the feature maps and works as a codebook. Instead of using the quantized feature map to generate the target, ReSSL [49] uses different views for the teacher and student based on the

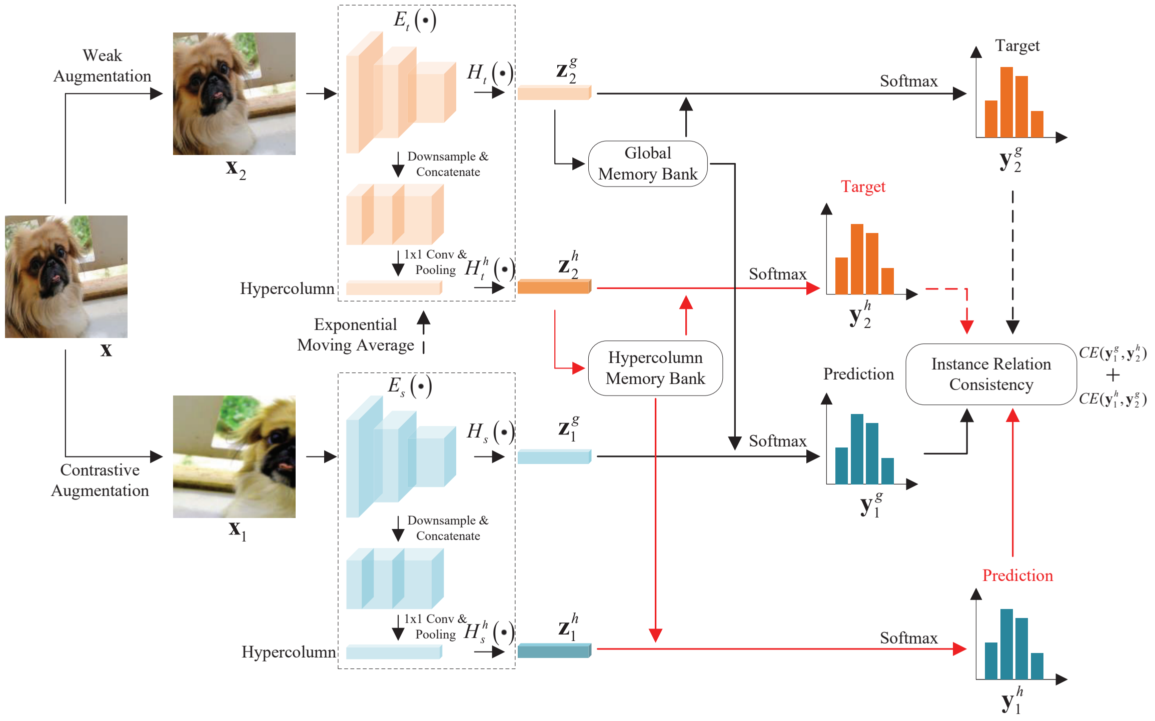

Figure 1. Overview of the proposed CGH framework. We adopt a knowledge distillation framework where the teacher is the exponential moving average of the student. A heavily corrupted view $\mathbf{x}_1$ is fed into the student E_s to obtain both a hypercolumn embedding $\mathbf{z}_1^h$ and a global embedding $\mathbf{z}_1^g$ while a weakly augmented view $\mathbf{x}_2$ is passed to the teacher E_t to obtain a hypercolumn embedding $\mathbf{z}_2^h$ and a global embedding $\mathbf{z}_2^g$. The embeddings are used to measure the similarity relationships between the augmented views $\mathbf{x}_1$, $\mathbf{x}_2$ and the samples in the memory bank – this leads to a similarity distribution. We enforce two instance relations alignments: "*global-hypercolumn alignment*" and "*hypercolumn-global alignment*", which are detailed in the text.

weak-contrastive augmentation strategy. The view through weak augmentation is fed into the teacher to provide a reliable target while the other view (via contrastive augmentation) is passed to the student for prediction. The consistency between different views is enforced. These methods are limited to only learn from global features after the global pooling layer. By contrast, we propose to enforce cross-context learning by using the one of the context (*e.g.*, intermediate layers) as guidance to learn the other (*e.g.*, global features).

3. Methodology

3.1. Overview

An overview of the proposed approach is shown in Fig. 1. The core idea of our scheme is to learn from cross-context information, one context derived from the global features after the global average pooling and one context derived from a hypercolumn that is constructed by the concatenation of intermediate feature maps. To achieve that, we enforce two instance relations alignments: "*global-hypercolumn alignment*" and "*hypercolumn-global alignment*". The global-hypercolumn alignment aims to use the hypercolumn of the teacher E_t to generate a target similarity distribution to guide the learning of the similarity distribution based on the global feature of the student E_s while hypercolumn-global alignment uses the global feature of the teacher to create a target distribution for guiding the learning of the similarity distribution based on the hypercolumn for the student. The effectiveness of the cross-context learning is analysed in Sec. 4.6. We provide the training cost analysis in the supplementary material.

3.2. Cross-context learning

Given an image $\mathbf{x}$, we generate a weakly augmented view $\mathbf{x}_2$ through weak augmentation for the teacher and a heavily augmented view $\mathbf{x}_1$ through contrastive augmentation for the student as in [15, 49]. Compared with the weak augmentation, the contrastive augmentation is more aggressive and generates heavily corrupted views. The student is trained to adapt to the heavy disturbance and noise introduced by the contrastive augmentation to learn robust representations. By contrast, the teacher generates a stable target based on the less aggressive weak augmentation.

We then proceed to generate the contexts of global feature and hypercolumn for the teacher and the student separately, as shown in Fig. 1. First, $\mathbf{x}_2$ is passed to the teacher encoder E_t to produce the "***global feature context***" (after the global average pooling) $\mathbf{h}_2^g = E_t(\mathbf{x}_2)$. Then $\mathbf{h}_2^g$ is transformed by a global projector H_t to produce a low-dimensional global embedding by $\mathbf{z}_2^g = H_t(\mathbf{h}_2^g)$ as in [8, 49]. As for the hypercolumn of the teacher, let $E_t^l(\mathbf{x}_2) \in \mathbb{R}^{c_l \times h_l \times w_l}$ be the intermediate feature maps of the l-th convolutional block, $l \in \{0, \ldots, L\}$, where c_l denotes the number of channels, h_l is the height and w_l is the width. The intermediate feature maps $\{E_t^l(\mathbf{x}_2)\}$, which are downsampled to the same spatial size as the output of the last convolutional block $E_t^L(\mathbf{x}_2)$ to reduce GPU memory consumption, are concatenated first and then mapped to a d-dimensional latent space through a 1×1 convolution followed by average pooling to obtain the "***hypercolumn context***" $\mathbf{h}_2^h \in \mathbb{R}^d$. $\mathbf{h}_2^h$ is transformed by another projector H_t^h to obtain the hypercolumn embedding by $\mathbf{z}_2^h = H_t^h(\mathbf{h}_2^h)$. Thus the contexts of the global feature $\mathbf{h}_2^g$ and hypercolumnn $\mathbf{h}_2^h$ are obtained for the teacher. Likewise, for the student, we produce the global feature context $\mathbf{h}_1^g = E_s(\mathbf{x}_1)$, hypercolumnn context $\mathbf{h}_1^h$ and the corresponding embeddings $\mathbf{z}_1^g = H_s(\mathbf{h}_1^g)$ and $\mathbf{z}_1^h = H_s^h(\mathbf{h}_1^h)$ for the heavily corrupted view $\mathbf{x}_1$.

Next we measure the similarity relationships between the augmented views ($\mathbf{x}_1$ and $\mathbf{x}_2$) and the samples in the memory bank. Following [8, 49], the embeddings are used to maintain two queue-based memory banks separately: a global memory bank $\mathcal{Q}$ based on $\mathbf{z}_2^g$ and a hypercolumn memory bank $\mathcal{Q}^h$ based on $\mathbf{z}_2^h$. To guide the learning of the global feature context $\mathbf{h}_1^g$ for the student, we use the similarity relationships between $\mathbf{h}_2^h$ and the embeddings $\hat{\mathbf{z}}_i^h$ in the hypercolumn memory bank $\mathcal{Q}^h$ as the target. The relationships are measured using the cosine similarity between $\mathbf{z}_2^h$ and $\hat{\mathbf{z}}_i^h$. We normalize the similarities with a softmax operation and produce a target probabilistic distribution $\mathbf{y}_2^h$ for the teacher:

$$\mathbf{y}_2^h[i] = \frac{\exp\left(\mathrm{sim}(\mathbf{z}_2^h, \hat{\mathbf{z}}_i^h)/\tau_h\right)}{\sum_{k=1}^{M} \exp\left(\mathrm{sim}(\mathbf{z}_2^h, \hat{\mathbf{z}}_k^h)/\tau_h\right)}, \tag{1}$$

where $\mathbf{y}_2^h[i]$ is the i-th element of the target similarity distribution generated by hypercolumn context $\mathbf{h}_2^h$, $\hat{\mathbf{z}}_i^h$ is the i-th embedding in the hypercolumn memory bank $\mathcal{Q}^h$, τ_h is the temperature parameter for the hypercolumn context, M is the size of the memory bank and $\mathrm{sim}(\mathbf{u}, \mathbf{v}) = \frac{\mathbf{u}^\top \mathbf{v}}{\|\mathbf{u}\|_2 \|\mathbf{v}\|_2}$ denotes the cosine similarity between the vectors $\mathbf{u}$ and $\mathbf{v}$. Similarly, the predicted distribution from the student is expressed as follows:

$$\mathbf{y}_1^g[i] = \frac{\exp\left(\mathrm{sim}(\mathbf{z}_1^g, \hat{\mathbf{z}}_i/\tau_s)\right)}{\sum_{k=1}^{M} \exp\left(\mathrm{sim}(\mathbf{z}_1^g, \hat{\mathbf{z}}_k)/\tau_s\right)}, \tag{2}$$

where $\mathbf{y}_1^g[i]$ is the i-th element of the predicted similarity distribution generated by global feature context $\mathbf{h}_1^g$, $\hat{\mathbf{z}}_i$ is the i-th embedding in the memory bank $\mathcal{Q}$ and τ_s is the temperature for global feature context of the student. The global-hypercolumn alignment predicts the hypercolumn based similarity distribution $\mathbf{y}_2^h$ from the global feature based distribution $\mathbf{y}_1^g$ by minimizing the cross-entropy loss:

$$\mathcal{L}_{\text{gh}} = CE(\mathbf{y}_1^g, \mathbf{y}_2^h), \tag{3}$$

where $CE(\mathbf{y}_1, \mathbf{y}_2) = -\sum_{k=1}^{M} \mathbf{y}_2[k] \log \mathbf{y}_1[k]$.

Similarly, for hypercolumn-global alignment, we guide the learning of the hypercolumn context $\mathbf{h}_1^h$ for the student using the global feature context $\mathbf{h}_2^g$ as target. The distributions generated by $\mathbf{h}_1^h$ and $\mathbf{h}_2^g$ are obtained as follows:

$$\begin{aligned} \mathbf{y}_1^h[i] &= \frac{\exp\left(\mathrm{sim}(\mathbf{z}_1^h, \hat{\mathbf{z}}_i^h/\tau_h)\right)}{\sum_{k=1}^{M} \exp\left(\mathrm{sim}(\mathbf{z}_1^h, \hat{\mathbf{z}}_k^h)/\tau_h\right)}, \\ \mathbf{y}_2^g[i] &= \frac{\exp\left(\mathrm{sim}(\mathbf{z}_2^g, \hat{\mathbf{z}}_i)/\tau_t\right)}{\sum_{k=1}^{M} \exp\left(\mathrm{sim}(\mathbf{z}_2^g, \hat{\mathbf{z}}_k)/\tau_t\right)}, \end{aligned} \tag{4}$$

where τ_t is the temperature for the global feature context of the teacher. The objective for hypercolumn-global alignment is expressed as:

$$\mathcal{L}_{\text{hg}} = CE(\mathbf{y}_1^h, \mathbf{y}_2^g). \tag{5}$$

Altogether, we enforce the cross-context learning between the global feature context and hypercolumn context with the following objective:

$$\mathcal{L} = \mathcal{L}_{\text{gh}} + \mathcal{L}_{\text{hg}} = CE(\mathbf{y}_1^g, \mathbf{y}_2^h) + CE(\mathbf{y}_1^h, \mathbf{y}_2^g). \tag{6}$$

3.3. Momentum update

The teacher is updated by the exponential moving average of the student:

$$\begin{aligned} E_t &\leftarrow mE_t + (1-m)E_s, \\ H_t &\leftarrow mH_t + (1-m)H_s, \\ H_t^h &\leftarrow mH_t^h + (1-m)H_s^h, \end{aligned} \tag{7}$$

where m is the momentum coefficient, which is set to 0.999 in all experiments following [8, 49].

3.4. Architecture

Following the common settings in self-supervised representation learning with Siamese structure [8, 16], we use ResNet as the online encoder and its momentum-updated version as the momentum encoder. In our framework, the momentum encoder is used as the teacher and the online encoder is used as the student. As in [8, 49], a two-layer MLP is adopted as the projector H_s for transforming the global feature from the global average pooling layer. Additionally, we adopt another two-layer MLP, which has the same architecture as H_s, as the projector H_s^h for transforming the hypercolumn. Both projectors consist of two linear layers with a ReLU non-linear activation in between. Following ReSSL [49], the hidden and output dimension of both projectors are set to 4096 and 512, respectively. When transforming the concatenation of feature maps to generate the hypercolumn vector, we use a 1×1 convolutional layer followed by Batch Normalization (BN) [26], ReLU activation and global average pooling. In our experiments, we use the outputs of the four convolutional blocks of ResNet as intermediate feature maps.

4. Experiments

In this section, we perform performance evaluation on widely used self-supervised learning benchmarks, including classification dataset ImageNet-1k [11] (also known as IN-1K) and detection datasets (i.e, PASCAL VOC [13] and COCO [35]). The visualization results are provided in the supplementary material.

4.1. Experimental setups

4.1.1 Implementation details

We adopt the same encoder backbone for all methods. The experiments on non-ImageNet datasets adopt ResNet-18 while the experiments on IN-1K use ResNet-50. For contrastive augmentation, we use the same strategy as in contrastive learning [8]. For weak augmentation, we use random resized crop and random horizontal flip, which is also the practice in ReSSL.

The teacher temperature and student temperature for global feature are set to $\tau_t = 0.04$ and $\tau_s = 0.1$ respectively, following ReSSL. The hypercolumn temperature is set to $\tau_h = 0.08$. The outputs of the third and fourth convolutional block are used for generating the hypercolumn. For a fair comparison, the other hyper-parameters are kept the same as ReSSL in all experiments. In the revised version of ReSSL [50], an additional predictor is used to further improve performance (denoted as **ReSSL-pred**). We also report our results with a predictor (denoted as **CGH-pred**) using the same pre-training details as discussed above. Note that **2x backprop** methods update the encoder parameters twice at each training step using the two augmented views, which means more samples are used within the same epochs and much higher training cost than **1x backprop** methods [25, 49] like ours.

4.1.2 Training details

By default, the pre-training is performed on the training set of IN-1K with 2 NVIDIA A100 GPUs. In ablation studies, we perform the pre-training on Tiny-ImageNet [32] and STL-10 [10] for 400 epochs. The training recipes are detailed as follows.

Tiny-ImageNet/STL-10. Following ReSSL, we pre-train for 400 epochs, using the SGD optimizer with 0.06 learning rate, 5e-4 weight decay, and 0.9 momentum. The batch size is set to 256. Following the linear evaluation protocol in [30], we train the classifier for 100 epochs with a batch size of 256, 3.0 learning rate, no weight decay, 0.9 momentum and cosine learning rate decay.

IN-1K. Following ReSSL, we pre-train the model for 200 epochs, using the SGD optimizer with 0.05 learning rate, 1e-4 weight decay, and 0.9 momentum. The batch size is set to 256. As in ReSSL, for linear evaluation, we use 0.3 learning rate, no weight decay, 0.9 momentum and cosine learning rate decay.

4.2. Linear classification and KNN evaluation

In this section, following the linear evaluation protocol [6, 8], we evaluate the learned representations by learning a linear classifier on top of the frozen pre-trained encoder for classification task. The encoder is pre-trained on the training set of the dataset first and then the linear classifier is trained on the training set with labels. The classification accuracy on the validation set is reported. For KNN evaluation, we follow the protocol in [17, 43] by evaluating the learned encoder with K-nearest neighbor (KNN) classifier using several nearest neighbor settings $\{10, 20, 100, 200\}$ and reporting the highest accuracy.

The linear and KNN classification results on IN-1K using 200 pre-training epochs are provided in Tab. 1. The proposed method outperforms MoCo-v2/ReSSL by 3.0%/1.2% on linear classification and $7.0\%/1.6\%$ on KNN classification, respectively. The consistent improvement compared with the baselines shows the effectiveness of the proposed cross-context learning strategy.

To further demonstrate our performance, we provide the classification results on IN-1K with multi-crop strategy [23] and longer pre-training epochs in Tab. 2. As we can see, the proposed CGH outperforms previous state-of-the-art methods. Note that our method also outperforms strong baselines that learn from multi-level signals (intermediate features), such as OBoW [15] and CsMl [45]. **The difference between our CGH and these multi-level methods are discussed in the supplementary material**.

4.3. Semi-supervised classification

We report the semi-supervised learning results by fine-tuning the self-supervised pre-trained ResNet-50 using 1% and 10% labelled data in IN-1K. We follow the semi-supervised protocol of [6,49] and report the results in Tab. 3. We use SGD optimizer with batch size of 256, weight decay of 0, and momentum of 0.9 for fine-tuning. For 1% setting, we train for 50 epochs using initial learning rate of 0.5 and 0.0001 for the classification head and feature extractor backbone, respectively, which are decayed by a factor of 0.1

Table 1. **Linear and KNN evaluation results on IN-1K with ResNet-50 backbone**. All methods are evaluated with the single-crop setting. Top-1 and Top-5 validation accuracy are reported. †: our reproduction using the official codes. $*$: results cited from [9].

Method	Backprop	Epochs	Batch Size	Linear Acc.	KNN Acc.
Supervised	1x	100	256	76.5	-
Asymmetric loss.					
MoCo-v2 [8]	1x	200	256	67.5	55.9
PCL-v2 [33]	1x	200	256	67.6	58.1
HCSC [17]	1x	200	256	69.2	60.7
OBoW [15]†	1x	200	256	69.5	57.2
ReSSL [49]†	1x	200	256	69.3	61.3
ReSSL-pred [50]	1x	200	1024	72.0	-
CGH	1x	200	256	**70.5**	**62.9**
CGH-pred	1x	200	256	**72.3**	**65.8**
Symmetric loss. 2× FLOPS					
SimCLR [6]$*$	2x	200	4096	68.3	-
SwAV [3]$*$	2x	200	4096	69.1	-
SimSiam [9]$*$	2x	200	256	70.0	-
BYOL [16]$*$	2x	200	4096	70.6	-
NNCLR [12]	2x	200	4096	70.7	-

Table 2. Linear evaluation on IN-1K with multi-crop strategy [3, 23] and different pre-training epochs.

Method	Backprop	Multi-Crop	Epochs	Batch Size	Linear Acc.
CMSF [37]	1x	✓	200	256	74.4
OBoW [15]	1x	✓	200	256	73.8
ReSSL [49]	1x	✓	200	256	74.7
CGH-pred	1x	✓	200	256	**75.7**
SwAV [3]	2x	✓	800	4096	75.3
HCSC [17]	1x	✓	800	256	74.2
CsMl [45]	2x	✓	300	1024	75.3
DINO [4]	1x	✓	800	4096	75.3
NNCLR [12]	2x	×	1000	4096	75.4
MAST [24]	2x	×	1000	2048	75.8
CGH-pred	1x	✓	400	256	**76.0**

after 30 and 40 epochs. In the 10% setting, we fine-tune for 50 epochs and set the initial learning rate to 0.2 and 0.0002 for the classification head and feature extractor backbone, respectively, which are decayed by a factor of 0.1 at the 30-th and 40-th epoch. Our method outperforms the other methods significantly with 200 pre-training epochs. Moreover, we also report the results with multi-crop in the last section of Tab. 3. In this case, our method achieves better performance than ReSSL on 1% split and comparable results on 10% split. Furthermore, our method outperforms 2x backprop methods with more pre-training epochs.

4.4. Transfer learning

We evaluate the transfer learning performance of the learned representations on the object detection and instance segmentation task. We fine-tune the model pre-trained on IN-1K on two widely used benchmarks PASCAL VOC [13] and COCO [35]. The same protocol and setups as MoCo-v2 are adopted. For PASCAL VOC object detection, we adopt Faster R-CNN [40] as the detector backbone, which is fine-tuned on training and validation splits of VOC 2007 and VOC 2012 and then tested on test set of VOC 2007; for COCO detection and segmentation, we use the Mask R-

Table 3. **IN-1K semi-supervised classification using ResNet-50 pre-trained on IN-1K**. **Multi** denotes the results with multi-crop. Top-1 and Top-5 validation accuracy are reported. †: our reproduction using the official codes. $*$: results cited from [25].

Method	Epochs	Batch Size	1% Labels		10% Labels	
			Top-1	Top-5	Top-1	Top-5
Asymmetric loss.						
MoCo-v2 [8]$*$	200	256	43.8	72.3	61.9	84.6
HCSC [17]	200	256	48.0	75.6	64.3	86.0
ReSSL [49]†	200	256	51.1	77.3	65.0	87.1
CGH	200	256	**53.2**	**78.9**	**66.4**	**88.0**
Symmetric loss. 2× FLOPS						
SimCLR [6]	1000	4096	48.3	75.5	65.6	87.8
SwAV [3]	800	4096	53.9	78.5	70.2	89.9
BYOL [16]	1000	4096	53.2	78.4	68.8	89.0
Multi-crop						
ReSSL (Multi) [49]	200	256	57.9	-	**70.4**	-
CGH (Multi)	200	256	**58.4**	**82.4**	70.3	**90.3**

Table 4. **Transfer learning on PASCAL VOC object detection**. All models are pre-trained for 200 epochs on IN-1K using ResNet-50 as the encoder. ResNet-50-C4 is used as the fine-tuning backbone. The bounding-box detection score ($\mathrm{AP}^{\mathrm{bb}}$) is reported. †: our reproduction using the official codes. $*$: results cited from [9].

Method	$\mathrm{AP}^{\mathrm{bb}}$	$\mathrm{AP}^{\mathrm{bb}}_{50}$	$\mathrm{AP}^{\mathrm{bb}}_{75}$
Asymmetric loss.			
MoCo-v2 [8]	57.0	82.4	63.6
ReSSL [49]†	56.1	82.2	62.5
CGH	56.8	**82.6**	63.4
CGH (Multi)	**57.1**	**82.6**	**63.8**
Symmetric loss. 2× FLOPS			
SimCLR [6]$*$	55.5	81.8	61.4
SwAV [3]$*$	55.4	81.5	61.4
SimSiam [9]$*$	56.4	82.0	62.8
BYOL [16]$*$	55.3	81.4	61.1

CNN [20] backbone, which is trained on the training set and then evaluated on the validation set. The results on PASCAL VOC are reported in Tab. 4 while the performance on COCO can be found in the supplementary material. As we can see, the proposed method achieves competitive performance compared with the state-of-the-art methods, which demonstrates the generality of the learned representations.

4.5. Ablation studies

4.5.1 Comparison of different context variants

In contrast to global-context learning, we leverage the similarity relationships of one context (*e.g.*, hypercolumn) as a supervisory signal for the other context (*e.g.*, global feature). Alternatively, in addition to the consistency of the global features used in global-context learning framework, we could incorporate the context of hypercolumn by enforcing the consistency of the hypercolumns between the teacher and the student, which is termed as same-context. We compare global-context, same-context and cross-context in Tab. 5. Note that global-context is identical to ReSSL here. We find that by incorporating the con-

Table 5. Comparison of different context variants.

Method	Tiny-ImageNet	STL-10
CGH (global-context)	48.9	90.7
CGH (same-context)	51.6	91.0
CGH (cross-context)	**53.8**	**92.0**

Table 6. Effect of hypercolumn temperature on Tiny-ImageNet.

τ_h	0.02	0.04	0.06	0.08	0.1
Acc.	52.1	52.8	53.1	**53.8**	51.8

Table 7. Effect of combinations of intermediate layers.

Layers				Tiny-ImageNet	STL-10
L1	L2	L3	L4		
-	-	-	✓	48.9	90.7
✓	-	-	✓	53.6	91.8
-	✓	-	✓	54.1	92.2
-	-	✓	✓	53.8	92.0
✓	✓	✓	✓	**54.4**	**92.3**

text from intermediate layers, both same-context and cross-context outperform the global-context baseline. Moreover, cross-context achieves the best results. **This suggests that cross-context provides a superior supervisory signal compared with global-context learning because of the use of the other contexts for supervision**.

4.5.2 Hypercolumn temperature

We use a temperature to control the smoothness of the generated similarity distribution. Therefore, the temperature is an important hyper-parameter in our framework. In order to evaluate the effect of the temperature, we evaluate the values of τ_h from set $\{0.02, 0.04, 0.06, 0.08, 0.1\}$. As shown in Tab. 6, we observe an inverted U-shaped trend on the performance when we increase τ_h.

Note that when $\tau_h \rightarrow 0$, the distribution from the hypercolumn becomes extremely sharp. If it is used for target generation, the target turns into one-hot distribution where the goal is to match the query with the most similar sample from the memory bank instead of capturing the instance relations. In other words, the framework degrades to NNCLR [12], except NNCLR doesn't use hypercolumn or memory bank for generating candidate samples. By contrast, when $\tau_h \rightarrow 0.1$, the distribution becomes flat and fails to focus on similar samples. Therefore, the performance tends to be better when τ_h is within $[0.06, 0.08]$.

4.5.3 Intermediate layers for hypercolumn

It is interesting to explore the effectiveness and sensitivity of different combinations of the intermediate layers. We use one of the layers from layer1 to layer3, along with the layer4 to generate hypercolumn and analyze the effect. The results are provided in Tab. 7. We have the following observations: 1) The proposed CGH achieves the best result by using all four layers for hypercolumn generation. 2) Hypercolumn based on layer2 and layer4 achieves the second best result, which suggests that layer2 provides a better balance between low-level and high-level semantics compared with layer1 and layer3. 3) Regardless of different combinations of the intermediate layers, all variants outperform the baseline ReSSL (first row), which shows our method is robust to the choice of the intermediate layers. Note that when we only use the fourth convolutional block for hypercolumn generation, our model will be identical to ReSSL – this is the first row of Tab. 7.

4.6. Effectiveness of hypercolumn context

In this section, we demonstrate the benefits of using hypercolumn representations for learning global representations from the perspective of (soft-) selection of positive samples, following the protocol in [17]. To do so, we note that learning the student under the guidance of the teacher can be viewed as training with soft pseudo-labels provided by the teacher. By thresholding the similarity distribution of the teacher, the predicted positives and negatives in the memory bank are obtained. Since the labels of the dataset are publicly available, the ground-truth for the positives and negatives in the memory bank can also be obtained (the sample is positive if it belongs to the same class as the input). Therefore, we can calculate the recall and precision which indicate the true positive and false positive selected by the teacher's similarity distribution. In Fig. 2, we provide the plots of recall and precision on IN-1K during pre-training based on the hypercolumn distribution from the teacher, and the corresponding plots of ReSSL based on global-context distribution. It's shown that our method has considerably better recall than ReSSL for the duration of the training while maintaining similar precision levels (right hand side plot). In summary, the results show that the proposed scheme can find more correct positive samples corresponding to the same class as the input (true positives), and maintain a low false positive rate at the same time.

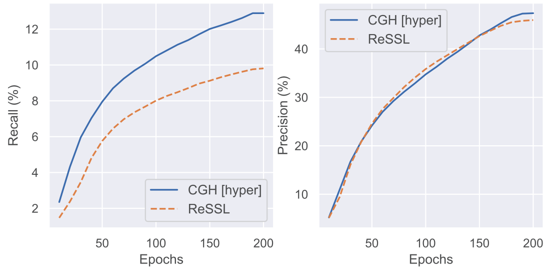

Figure 2. Performance of positive sample selection on IN-1K.

5. Conclusion

In order to solve the class collision problem in contrastive learning, inspired by cross-modality learning [1, 38], we present a novel framework based on knowledge distillation, cross-context learning between global and hypercolumn features (CGH) that learns representations by capturing cross-context information from the context of global features and hypercolumns. The cross-context learning strategy allows the model to identify more similar samples (true positives) in the memory bank and keep low false positives. The extensive experiments on classification and downstream tasks demonstrate the effectiveness and generality of our method.

Acknowledgement

This work was supported by the EU H2020 AI4Media No.951911 project. We thank James Oldfield for his helpful comments.

References

[1] Humam Alwassel, Dhruv Mahajan, Bruno Korbar, Lorenzo Torresani, Bernard Ghanem, and Du Tran. Self-supervised learning by cross-modal audio-video clustering. In *Advances in Neural Information Processing Systems*, volume 33, 2020. 1, 8

[2] 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

[3] Mathilde Caron, Ishan Misra, Julien Mairal, Priya Goyal, Piotr Bojanowski, and Armand Joulin. Unsupervised learning of visual features by contrasting cluster assignments. In *Proceedings of Advances in Neural Information Processing Systems (NeurIPS)*, 2020. 2, 6, 7

[4] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jegou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 9630–9640, 2021. 6

[5] Mayee Chen, Daniel Y Fu, Avanika Narayan, Michael Zhang, Zhao Song, Kayvon Fatahalian, and Christopher Re. Perfectly balanced: Improving transfer and robustness of supervised contrastive learning. In Kamalika Chaudhuri, Stefanie Jegelka, Le Song, Csaba Szepesvari, Gang Niu, and Sivan Sabato, editors, *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pages 3090–3122. PMLR, 17–23 Jul 2022. 1

[6] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In Hal Daumé III and Aarti Singh, editors, *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 1597–1607. PMLR, 13–18 Jul 2020. 2, 5, 6, 7

[7] Ting Chen, Simon Kornblith, Kevin Swersky, Mohammad Norouzi, and Geoffrey E Hinton. Big self-supervised models are strong semi-supervised learners. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 22243–22255. Curran Associates, Inc., 2020. 2

[8] Xinlei Chen, Haoqi Fan, Ross Girshick, and Kaiming He. Improved baselines with momentum contrastive learning. *arXiv preprint arXiv:2003.04297*, 2020. 2, 4, 5, 6, 7

[9] Xinlei Chen and Kaiming He. Exploring simple siamese representation learning. In *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 15745–15753, 2021. 2, 6, 7

[10] Adam Coates, Andrew Ng, and Honglak Lee. An analysis of single-layer networks in unsupervised feature learning. In Geoffrey Gordon, David Dunson, and Miroslav Dudík, editors, *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*, volume 15 of *Proceedings of Machine Learning Research*, pages 215–223, Fort Lauderdale, FL, USA, 11–13 Apr 2011. PMLR. 5

[11] J. Deng, W. Dong, R. Socher, L. Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 248–255, 2009. 5

[12] Debidatta Dwibedi, Yusuf Aytar, Jonathan Tompson, Pierre Sermanet, and Andrew Zisserman. With a little help from my friends: Nearest-neighbor contrastive learning of visual representations. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9588–9597, 2021. 1, 2, 6, 8

[13] Mark Everingham, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. The pascal visual object classes (voc) challenge. *International Journal of Computer Vision*, 88(2):303–338, 2010. 5, 6

[14] Chen Feng and Ioannis Patras. Adaptive soft contrastive learning. In *2022 26th International Conference on Pattern Recognition (ICPR)*, pages 2721–2727. IEEE, 2022. 1

[15] Spyros Gidaris, Andrei Bursuc, Gilles Puy, Nikos Komodakis, Matthieu Cord, and Patrick Pérez. Obow: Online bag-of-visual-words generation for self-supervised learning. In *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6826–6836, 2021. 1, 2, 3, 5, 6

[16] Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Guo, Mohammad Gheshlaghi Azar, Bilal Piot, koray kavukcuoglu, Remi Munos, and Michal Valko. Bootstrap your own latent - a new approach to self-supervised learning. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 21271–21284. Curran Associates, Inc., 2020. 1, 2, 4, 6, 7

[17] Yuanfan Guo, Minghao Xu, Jiawen Li, Bingbing Ni, Xuanyu Zhu, Zhenbang Sun, and Yi Xu. Hcsc: Hierarchical contrastive selective coding. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9696–9705, 2022. 1, 2, 5, 6, 7, 8

[18] Bharath Hariharan, Pablo Arbeláez, Ross Girshick, and Jitendra Malik. Hypercolumns for object segmentation and fine-grained localization. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 447–456, 2015. 2

[19] K. He, H. Fan, Y. Wu, S. Xie, and R. Girshick. Momentum contrast for unsupervised visual representation learning. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9726–9735, 2020. 1, 2

[20] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 2980–2988, 2017. 7

[21] Olivier Henaff. Data-efficient image recognition with contrastive predictive coding. In *International Conference on Machine Learning*, pages 4182–4192. PMLR, 2020. 2

[22] Devon Hjelm, Alex Fedorov, Samuel Lavoie-Marchildon, Karan Grewal, Philip Bachman, Adam Trischler, and Yoshua Bengio. Learning deep representations by mutual information estimation and maximization. In *ICLR 2019*. ICLR, April 2019. 1

[23] Qianjiang Hu, Xiao Wang, Wei Hu, and Guo-Jun Qi. Adco: Adversarial contrast for efficient learning of unsupervised representations from self-trained negative adversaries. In *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1074–1083, 2021. 5, 6

[24] Chen Huang, Hanlin Goh, Jiatao Gu, and Joshua M. Susskind. MAST: Masked augmentation subspace training for generalizable self-supervised priors. In *The Eleventh International Conference on Learning Representations*, 2023. 6

[25] Lang Huang, Shan You, Mingkai Zheng, Fei Wang, Chen Qian, and Toshihiko Yamasaki. Learning where to learn in cross-view self-supervised learning. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 14431–14440, 2022. 5, 7

[26] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In Francis Bach and David Blei, editors, *Proceedings of the 32nd International Conference on Machine Learning*, volume 37 of *Proceedings of Machine Learning Research*, pages 448–456, Lille, France, 07–09 Jul 2015. PMLR. 5

[27] X. Ji, A. Vedaldi, and J. Henriques. Invariant information clustering for unsupervised image classification and segmentation. In *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 9864–9873, 2019. 1

[28] Prannay Khosla, Piotr Teterwak, Chen Wang, Aaron Sarna, Yonglong Tian, Phillip Isola, Aaron Maschinot, Ce Liu, and Dilip Krishnan. Supervised contrastive learning. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 18661–18673. Curran Associates, Inc., 2020. 1

[29] Kyungyul Kim, ByeongMoon Ji, Doyoung Yoon, and Sangheum Hwang. Self-knowledge distillation with progressive refinement of targets. In *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 6547–6556, 2021. 2

[30] Sungnyun Kim, Gihun Lee, Sangmin Bae, and Se-Young Yun. Mixco: Mix-up contrastive learning for visual representation. *arXiv preprint arXiv:2010.06300*, 2020. 5

[31] Soroush Abbasi Koohpayegani, Ajinkya Tejankar, and Hamed Pirsiavash. Mean shift for self-supervised learning. In *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 10306–10315, 2021. 2

[32] Ya Le and X. Yang. Tiny imagenet visual recognition challenge. *CS 231N*, 2015. 5

[33] Junnan Li, Pan Zhou, Caiming Xiong, and Steven Hoi. Prototypical contrastive learning of unsupervised representations. In *International Conference on Learning Representations*, 2021. 1, 2, 6

[34] Tianhong Li, Lijie Fan, Yuan Yuan, Hao He, Yonglong Tian, Rogerio Feris, Piotr Indyk, and Dina Katabi. Addressing feature suppression in unsupervised visual representations. *arXiv e-prints*, pages arXiv–2012, 2020. 2

[35] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. Microsoft coco: Common objects in context. In David Fleet, Tomas Pajdla, Bernt Schiele, and Tinne Tuytelaars, editors, *Computer Vision – ECCV 2014*, pages 740–755, Cham, 2014. Springer International Publishing. 5, 6

[36] S. Lloyd. Least squares quantization in pcm. *IEEE Transactions on Information Theory*, 28(2):129–137, 1982. 2

[37] K. L. Navaneet, Soroush Abbasi Koohpayegani, Ajinkya Tejankar, Kossar Pourahmadi, Akshayvarun Subramanya, and Hamed Pirsiavash. Constrained mean shift using distant yet related neighbors for representation learning. In Shai Avidan, Gabriel Brostow, Moustapha Cissé, Giovanni Maria Farinella, and Tal Hassner, editors, *Computer Vision – ECCV 2022*, pages 23–41, Cham, 2022. Springer Nature Switzerland. 2, 6

[38] Xiaokang Peng, Yake Wei, Andong Deng, Dong Wang, and Di Hu. Balanced multimodal learning via on-the-fly gradient modulation. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8228–8237, 2022. 1, 8

[39] Alexandre Ramé, Rémy Sun, and Matthieu Cord. Mixmo: Mixing multiple inputs for multiple outputs via deep subnetworks. In *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 803–813, 2021. 2

[40] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In C. Cortes, N. Lawrence, D. Lee, M. Sugiyama, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 28. Curran Associates, Inc., 2015. 6

[41] Joshua David Robinson, Ching-Yao Chuang, Suvrit Sra, and Stefanie Jegelka. Contrastive learning with hard negative samples. In *International Conference on Learning Representations*, 2021. 2

[42] Xinlong Wang, Rufeng Zhang, Chunhua Shen, Tao Kong, and Lei Li. Dense contrastive learning for self-supervised visual pre-training. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3024–3033, 2021. 1

[43] Zhirong Wu, Yuanjun Xiong, Stella X. Yu, and Dahua Lin. Unsupervised feature learning via non-parametric instance discrimination. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3733–3742, 2018. 5

[44] Zhenda Xie, Yutong Lin, Zheng Zhang, Yue Cao, Stephen Lin, and Han Hu. Propagate yourself: Exploring pixel-level consistency for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16684–16693, 2021. 1

[45] Haohang Xu, Xiaopeng Zhang, Hao Li, Lingxi Xie, Wenrui Dai, Hongkai Xiong, and Qi Tian. Seed the views: Hierarchical semantic alignment for contrastive representation learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(3):3753–3767, 2023. 5, 6

[46] M. Ye, X. Zhang, P. C. Yuen, and S. Chang. Unsupervised embedding learning via invariant and spreading instance feature. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6203–6212, 2019. 1

[47] Linfeng Zhang, Jiebo Song, Anni Gao, Jingwei Chen, Chenglong Bao, and Kaisheng Ma. Be your own teacher: Improve the performance of convolutional neural networks via self distillation. In *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 3712–3721, 2019. 2

[48] Nanxuan Zhao, Zhirong Wu, Rynson WH Lau, and Stephen Lin. What makes instance discrimination good for transfer learning? *arXiv preprint arXiv:2006.06606*, 2020. 1

[49] Mingkai Zheng, Shan You, Fei Wang, Chen Qian, Changshui Zhang, Xiaogang Wang, and Chang Xu. Ressl: Relational self-supervised learning with weak augmentation. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan, editors, *Advances in Neural Information Processing Systems*, volume 34, pages 2543–2555. Curran Associates, Inc., 2021. 1, 2, 3, 4, 5, 6, 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. 5, 6