ADACLR: ADAPTIVE CONTRASTIVE LEARNING OF REPRESENTATION BY NEAREST POSITIVE EXPANSION

Anonymous authors
Paper under double-blind review

ABSTRACT

Despite their success in perception over the last decade, deep neural networks are also known ravenous to labeled data for training, which limits their applicability to real-world problems. Hence self-supervised learning has recently attracted intensive attention. Contrastive learning has been one of the dominant approaches for effective feature extraction and has also achieved state-of-the-art performance. In this paper, we first theoretically show that these methods cannot fully take advantage of training samples in the sense of nearest positive samples mining. Then we propose a new contrastive method called AdaCLR^{pre} (adaptive self-supervised contrastive learning representations), which can more effectively (supported by our proof) explore the samples in a way of being closer to supervised contrastive learning. We thoroughly evaluate the quality of the learned representation on ImageNet for pretraining based version (AdaCLR^{pre}). The results of accuracy show AdaCLR^{pre} outperforms state-of-the-art contrastive-based models by 3.0% with extra 100 epochs.

1 INTRODUCTION

Unsupervised learning (Hadsell et al., 2006b; Goodfellow et al., 2014; Maji et al., 2013) and self-supervised learning (El-Yaniv & Pechyony, 2009; Chen et al., 2020a; He et al., 2020; Gidaris et al., 2018)) have spanned a line of heated research. Most mainstream approaches fall into one of the three classes: i) generative, ii) contrastive and iii) pretext tasks based methods. Generative based methods (Goodfellow et al., 2014; Maji et al., 2013; Zhang et al., 2016) mainly use pixel-level reconstruction to learn feature extractors. However, since feature extraction can be regarded as dimensionality reduction, it is not necessary to record the information of each pixel. In contrast, more attempts are to learn feature extractors by providing pretext tasks (Doersch et al., 2015; Hadsell et al., 2006b; Oord et al., 2018), which typically involve artificially generating some transformed images e.g. by rotating, or adding artifacts, allowing the training of predictive models to these tasks for feature extraction. Contrastive based methods (Chen et al., 2020a; He et al., 2020; Chen et al., 2020c) recently emerge which have achieved promising accuracy, with the idea dating back to (Hadsell et al., 2006a). The idea is that the same images generated by different augmentations still retains the same semantic information. Therefore, it takes such images as positive samples and other samples as negative ones, and the objective is often based on the Noise Contrastive Estimation (NCE) model e.g. InfoNCE (Oord et al., 2018). Contrastive learning often requires a very large batch size, that is, sufficient negative samples are provided in a batch. In this paper, we show by both empirical results and theoretical proof that, there is still much space to improve existing methods by more effectively explore the training samples.

Departure from the classic cross-entropy for supervised classification (Baum & Wilczek, 1988; Levin & Fleisher, 1988; Rumelhart et al., 1986), which introduces the KL-divergence between two discrete distributions: the label distribution (a discrete distribution of one-hot encoding) and the empirical distribution of the logits, the recent work (Khosla et al., 2020) firstly abandons cross-entropy loss, and instead adopts contrastive loss for supervised classification. By the natural formulation of contrastive loss that pulls the representations of samples from the same class closer together, rather than forcing them to be pulled towards a specific target as done in cross-entropy, it achieves state-of-the-art performance for supervised learning. Back to the unsupervised setting as the focus of this paper, we note that self-supervised contrastive learning (Chen et al., 2020a; He et al., 2020; Oord et al., 2018) typically treats all the images augmented from the same sample as positive and contrastive loss is used for feature learning rather than cross-entropy. Thus, the gap between supervised and unsupervsied can be defined that supervised contrastive learning can take use of the supervisory information which indicates the samples belong to one category or not.

Fortunately, these supervisory information can be obtained via adaptively learning (Blum & Mitchell, 1998; Dasgupta et al., 2002). This paper presents a solution to this problem, which fills the gap between self-supervised contrastive learning and supervised contrastive learning. In a nutshell, the main contributions include:

1) We propose an adaptive self-supervised learning technique to mine nearest positive samples (mainly composed by hard positive pairs). The resulting approach is termed AdaCLR and the experimental results show that AdaCLR^{pre} can notably outperform existing contrastive based self-supervised methods.

2) We theoretically show that the state-of-the-art self-supervised contrastive learning methods (Chen et al., 2020a) are still unable to fully explore the samples, while our proposed AdaCLR^{pre} is closer to supervised contrastive learning.

3) We thoroughly evaluate the quality of the learnt representation on ImageNet. AdaCLR^{pre} can outperform state-of-the-art contrastive model in accuracy by 3.0% with extra 100 epochs.

2 Related work

Unsupervised learning has spanned a few representative subareas. We review recent pretext task based and contrastive learning techniques. We also discuss boosting and semi-supervised models.

Pretext tasks. Pretext tasks based methods (Alwassel et al., 2019; Zhang et al., 2016; Jenni & Favaro, 2018; Dosovitskiy et al., 2014; Caron et al., 2018) in general involve rendering new images by certain rules for transformation, and representation learning is fulfilled by prediction of the enforced transformation. RotNet (Gidaris et al., 2018) proposes making pseudo labels by rotating images. The work (Doersch et al., 2015) proposes training an encoder i.e. feature extractor by predicting the relative position of image patches. The work (Noroozi & Favaro, 2016) views solving jigsaw puzzles as pretext task. There are more self-supervised tasks are as diverse as in-painting (Pathak et al., 2016), instance counting (Noroozi et al., 2017) and colorization (Lai & Xie, 2019). BYOL (Grill et al., 2020) adopts teacher-student framework to learn one encoder from the other encoder, which means the final encoder requires training in multiple steps.

Contrastive learning. The pioneering work (Gutmann & Hyvärinen, 2010) formally defines Noise-Contrastive Estimation (NCE), to distinguish two different distributions. Recently, contrastive based methods have shown their efficacy for self-supervised learning (Chen et al., 2020a; He et al., 2020; Oord et al., 2018) and supervised learning (Khosla et al., 2020). Specifically, the method CPC (Oord et al., 2018) proposes the InfoNCE loss, whereby positive pairs and negative pairs are constructed from sequential setting data. MoCo (He et al., 2020) adopts memory bank strategy and uses momentum-based technique to upgrade encoders, whereby the InfoNCE loss is also adopted. In detail, they construct two distributions to distinguish by InfoNCE: Gaussian Noise and the distribution of original data. Notably, SimCLR (Chen et al., 2020a) proposes a simple yet efficient framework, which directly augments original data with stronger augmentation functions. In their framework, InfoNCE aims to distinguish the distribution of augmented samples from others. Inspired by SimCLR, Moco v2 (Chen et al., 2020c) adopts the same augmentation functions with SimCLR. In contrast, there is little work (Khosla et al., 2020) on supervised contrastive learning. Almost all existing contrastive-based self-supervised methods treat only images augmented from the same sample as positive pairs, and they can only mine hard positives by random augmentation. The result of the work (Khosla et al., 2020) indicates when expand the positive pairs, the performance can greatly improve. In detail, they use the label information to construct positives. Inspired by this, we design a new approach to expand positive pairs adaptively without using label information.

Boosting and semi-supervised learning. The work (Noroozi et al., 2018) brings the idea of boosting to self-supervised learning, whereby an encoder is first trained through pretext tasks. Then, extracted features are clustered by k-means. Finally, pseudo-labels are extracted from the clustering results to fine-tune the encoder. There are also semi-supervised methods when a small portion of labeled data is available, such as co-training (Nigam & Ghani, 2000; Kumar & Daumé, 2011), instance weighting (Jiang & Zhai, 2007; Ting, 2002). The work (Zhai et al., 2019) combines self-supervised learning and semi-supervised learning, taking consideration of balancing pretext tasks' loss and classification loss. Inspired by the idea of boosting and semi-supervised learning, we label the top-K most similar pairs as positives, and then boost the encoder with these pseudo-labeled data.

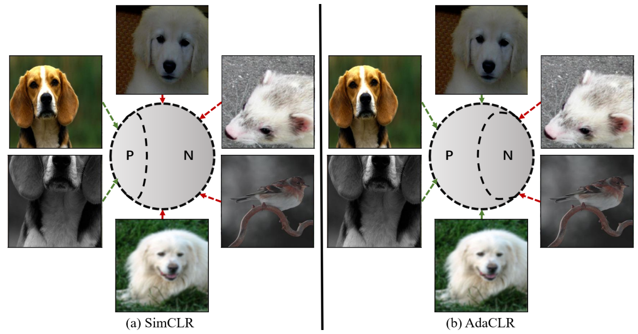

Figure 1: SimCLR (left) regards only data augmented from the same image as positive samples, while the proposed AdaCLR (right) automatically mines these positive samples.

3 THE PROPOSED METHODOLOGY

We detail the method of AdaCLR^{pre} in this section. We also theoretically analyze its convergence rate and generalization capability.

3.1 PRELIMINARIES AND BRIEF ON CONTRASTIVE LEARNING

Given an unlabeled dataset $\mathcal{D} = \{\mathbf{x}\}_{i=1}^{N}$ and a set of augmentation functions $g(\cdot, \boldsymbol{\theta})$, where θ is random seed, self-supervised learning aims to train an encoder function $\mathcal{F}(\cdot)$ (He et al., 2016) for feature extraction without supervision.

We introduce the general pipeline (He et al., 2020) for contrastive learning as follows. First, one mini-batch data $\mathcal{B} = \{\mathbf{x}\}_{i=1}^{b}$ from $\mathcal{D}$ is sampled where b is batch size. Taking sampled data as input of augmentation function, we can then get two mini-batch data termed as $\{\mathbf{x}^1\}_{i=1}^{b} = g(\mathbf{x}, \boldsymbol{\theta}_1)$, $\{\mathbf{x}^2\}_{i=1}^{b} = g(\mathbf{x}, \boldsymbol{\theta}_2)$. The defined encoder can extract information from augmented images, which can be written as $\{\mathbf{h}^1\} = \mathcal{F}(\{\mathbf{x}^1\}_{i=1}^{b})$ and $\{\mathbf{h}^2\} = \mathcal{F}(\{\mathbf{x}^2\}_{i=1}^{b})$. Then a two-layer neural network projection head that maps representations space to the contrastive space is adopted, which can be written as: $\{\mathbf{z}\} = \mathbf{W}_2(\sigma(\mathbf{W}_1 * \mathbf{h}))$, where σ is the activation function and $\mathbf{W}$ is linear layer. The above pipeline can be summarized as:

$$\mathbf{z} = \mathbf{W}_2 * \sigma(\mathbf{W}_1 * \mathcal{F}(g(\mathbf{x}, \boldsymbol{\theta}))). \tag{1}$$

For SimCLR and Moco, only the samples augmented by the same image are recognized as positives, while the others as negatives. Therefore, the negative set can be written as $\mathcal{N} = \{(\mathbf{z}_i^1, \mathbf{z}_j^2) \bigcup (\mathbf{z}_i^1, \mathbf{z}_j^1) \bigcup (\mathbf{z}_i^2, \mathbf{z}_j^2) | (i \neq j)\}$. While each sample only belongs to one positive pair. Thus, the positive pairs can be dentoted by $\mathcal{P} = \{(\mathbf{z}_i^1, \mathbf{z}_i^2) | 0 \leq i \leq b\}$, and the objective becomes:

$$\mathcal{L}_{info} = -\mathbb{E}\left[\log \frac{\exp(\mathbf{z}_i^1 \cdot \mathbf{z}_i^2 / \tau)}{\exp(\mathbf{z}_i^1 \cdot \mathbf{z}_i^2 / \tau) + \sum_{\mathbf{z}_i, \mathbf{z}_j \in \mathcal{N}} \exp(\mathbf{z}_i \cdot \mathbf{z}_j / \tau)}\right], \tag{2}$$

where $\mathbf{z}_i \cdot \mathbf{z}_j$ denotes the dot product of two vectors and τ the temperature parameter (Wu et al., 2018). This loss has been widely used in (Chen et al., 2020a; He et al., 2020; Oord et al., 2018).

3.2 OBJECTIVE FUNCTION

In line with the simple protocol in SimCLR (Chen et al., 2020a), in our approach we first train an encoder using the InfoNCE loss $\mathcal{L}_{info}$ (see Eq. 2). Then the second stage aims to mine nearest positive pairs automatically. For each image, we regard top-K closest vectors as its positives. For example, the positive pairs of image index of 1 can be written as:

$$\mathcal{P}_{1K} = \{(\mathbf{z}_i, \mathbf{z}_j) | (i, j) \in topK(\mathcal{S}_{1n}) | (1 \leq n \leq 2 * b - 1)\}, \tag{3}$$

where $\mathcal{S} \in \mathbb{R}^{2b\times(2b-1)}$ is the similarity matrix, as can be obtained by concatenating two mini-batch and then calculating dot product. Thus, the top-K positive pairs as denoted by $\mathcal{P}_{ada}$ is given by:

$$\mathcal{P}_{ada} = \{(\mathbf{z}_i, \mathbf{z}_j)|(\mathbf{z}_i, \mathbf{z}_j) \in \mathcal{P}_{iK}|i \in \{1, 2, 3 \cdots, 2 * b\}\}, \tag{4}$$

Denote $\mathcal{P}_{same} = \{(\mathbf{z}_i^1, \mathbf{z}_i^2)\}$, then the final positive pairs are the union: $\mathcal{P}_{topK} = \mathcal{P}_{ada} \bigcup \mathcal{P}_{same}$: given a weak encoder, the pair generated by the same image may not belong to $\mathcal{P}_{ada}$.

The negative set can be written as $\mathcal{N}_{topK} = \{(\mathbf{z}_i, \mathbf{z}_j)|(\mathbf{z}_i, \mathbf{z}_j) \notin \mathcal{P}_{topK}\}$. Then the objective is:

$$\mathcal{L}_{ada} = -\mathbb{E}\{\log \frac{\sum_{(\mathbf{z}_i,\mathbf{z}_j)\in\mathcal{P}_{topK}} \exp(\mathbf{z}_i \cdot \mathbf{z}_j/\tau)}{\sum_{(\mathbf{z}_i,\mathbf{z}_j)\in\mathcal{P}_{topK}} \exp(\mathbf{z}_i \cdot \mathbf{z}_j/\tau) + \sum_{(\mathbf{z}_i,\mathbf{z}_j)\in\mathcal{N}_{topK}} \exp(\mathbf{z}_i \cdot \mathbf{z}_j/\tau)}\}, \tag{5}$$

where $\mathcal{P}_{ada}$ can mine neatest positives automatically and $\mathcal{P}_{same}$ restricts lower bound of $\mathcal{L}_{ada}$. Finally, by reducing the error rate of pseudo labels, it is more close to supervised contrastive learning.

Note that AdaCLR^{pre} involves two training stages. In the first stage, $\mathcal{L}_{info}$ is adopted and in the second stage, the objective switches to $\mathcal{L}_{ada}$.

3.3 THEORETICAL ANALYSIS

Impact of batch size and categories size. Given a set of sampled mini-batch data $\{\mathcal{X}\}_{i=1}^b$. Denote $C(b, n)$ as combination function, as written as $C(b, n) = n!/b! * (n - b)!$. Suppose there are c categories, and there is p probability which can make at least δ percent categories with m samples:

$$p(\delta, i) = \frac{C(\lceil c \cdot \delta\rceil + i, b) \prod_{j=0}^{b} \{(\lceil c \cdot \delta\rceil + i) \cdot C(m, \lceil c \cdot \delta\rceil \cdot + i - j)\} \cdot C(\lceil c \cdot \delta\rceil, c)}{c^n}, \tag{6}$$

where $\lceil\cdot\rceil$ is the round up function and $p = \sum_{i=0}^{b-\lceil c*\delta\rceil} p(\delta, i)$. When the batch size is large enough, the performance of SimCLR won't improve with the batch size, since there are too many positives that are recognized as negatives. Specifically, when batch size is larger than c, there must be at least one positive which is mis-recognized into negative pair, with an adverse impact on the encoder.

Impact of directly training and boosting training. Note AdaCLR^{pre} involves two stages for training, and the first stage can give some prior knowledge to the second stage. Note that the label in $\mathcal{P}_{same}$ is strictly clean without noise, while $\mathcal{P}_{ada}$ can be recognized as unlabeled. In this sense, the training of AdaCLR^{pre} is similar to the setting of semi-supervised learning.

Theorem 1 ***(Generalization and Convergence rate of AdaCLR^{pre}).*** *Suppose we are given a mini-batch training sample of size l, after the first stage training, encoder function f can bring λl positive pair in $\mathcal{P}_{ada}$. Let $\mathcal{X}_{out}$ be the set of the full dataset. For any constant γ, with probability at least $1 - \delta$, the generalization error of AdaCLR^{pre} is upper bounded by:*

$$err_{out} \leq err_{in} + \frac{R_{(k+1)l}(\mathcal{X}_{out})}{\gamma} + cQ\sqrt{min((1+\lambda)l, (k-\lambda)l)} + \sqrt{\frac{SQ}{2} \ln \frac{1}{\delta}}, \tag{7}$$

where $Q = 1/(l + \lambda l) + 1/(kl - \lambda l)$ and $S = \frac{(1+k)l}{((1+k)l-1/(2))(1-1/max(1+\lambda,k-\lambda)l)}$. Moreover, the convergence rate of AdaCLR^{pre} is in order of:

$$\mathcal{O}(AdaCLR^{pre}) \approx \mathcal{O}(1/\sqrt{min(1+\lambda, k-\lambda)l}), \tag{8}$$

Proof *AdaCLR^{pre} selects the top-k closest data samples as positives, which can be recognized as a setting with $(1 + \lambda) * l$ labeled data and $(k - \lambda) * l$ unlabeled data for co-training, which satisfies the setting of Eq. 11 in (El-Yaniv & Pechyony, 2009). Then we can get:*

$$err_{out} \leq err_{in} + R_{m+u}(\mathcal{X}) + B_{max}cQ\sqrt{\min(m, u)} + B\sqrt{\frac{SQ}{2} \ln \frac{1}{\delta}}, \tag{9}$$

where B_{max} and B are two constants by letting $B = B_2 - B_1$ and $B_{max} = max(|B_1|, |B_2|)$. Take $B_1 = 0$, $B_2 = 1$, $m = (1 + \lambda) * l$ and $u = (k - \lambda) * l$ to Eq. 9. We can obtain Eq. 7.

When l is large enough, it is easy to get $\mathcal{S} \to 1$. Thus, the convergence rates of the term on the right of Eq. 8 are in order of $\mathcal{O}(cQ/\sqrt{\min(1 + \lambda, k - \lambda)l})$. Then, we can complete the proof.

Algorithm 1 Training procedure of AdaCLR^{pre}. $\mathbf{z} \in \mathbb{R}^{b\times d}$ is the semantic feature in contrastive space for d is the feature dimension size.

Input: number of positive pairs K, mini batch $\{\mathbf{x}\}_{i=1}^{b}$, augmentation function g, max epochs $maxiter$. $\mathcal{S} \in \mathbb{R}^{2b\times(2b-1)}$ is the similarity matrix. $\mathcal{M} \in \mathbb{R}^{2b\times 2b}$ is the mask matrix. $T(\cdot)$ is the transpose function;
Initialization: encoder function $\mathcal{F}$, two non-linear transformation head $(\mathbf{W}_1, \mathbf{W}_2, \sigma)$, $epoch = 0$, random seed list θ, mask matrix $\mathcal{M}$;
Output: trained encoder function $\mathcal{F}$;
Pre-train encoder by runnning the existing model SimCLR: $\mathcal{F} \leftarrow SimCLR(\mathcal{F})$;
while $epoch < maxiter$ **do**
  Obtain two-mini batch augmented data in Eq. 1: $\mathbf{x}^1 \leftarrow g(\mathbf{x}, \boldsymbol{\theta}_1), \mathbf{x}^2 \leftarrow g(\mathbf{x}, \boldsymbol{\theta}_2)$;
  Encoded feature in Eq. 1: $\mathbf{h}^1 \leftarrow \mathcal{F}(\mathbf{x}^1), \mathbf{h}^2 \leftarrow \mathcal{F}(\mathbf{x}^2)$;
  Two non-linear layer transformation in Eq. 1: $\mathbf{z}^1 \leftarrow \mathbf{W}_2\sigma(\mathbf{W}_1\mathbf{h}^1), \mathbf{z}^2 \leftarrow \mathbf{W}_2\sigma(\mathbf{W}_1\mathbf{h}^2)$;
  Calculate cosine similarity matrix: $\mathcal{S} \leftarrow cat(\mathbf{z}^1, \mathbf{z}^2, dim=1) \times T(cat(\mathbf{z}^1, \mathbf{z}^2, dim=1))$;
  Update $\mathcal{S}$ by masking the diagonal values of matrix. $\mathcal{S} \leftarrow S(M)$;
  Construct adaptive positive pairs in Eq. 3 and Eq. 4. $\mathcal{P}_{ada} \leftarrow \{(\mathbf{z}_i, \mathbf{z}_j)|(\mathbf{z}_i, \mathbf{z}_j) \in \mathcal{P}_{iK}|i \in \{1, 2, 3\cdots, 2*N\}\}$;
  Construct original positive pair: $\mathcal{P}_{same} \leftarrow \{\mathbf{z}_i^1, \mathbf{z}_i^2|0 \leq i \leq 2N-1\}$;
  Construct total positive pairs and negative pairs: $\mathcal{P}_{boost} \leftarrow \mathcal{P}_{ada} \bigcup \mathcal{P}_{same}$ and $\mathcal{N}_{boost} \leftarrow \{(\mathbf{z}_i, \mathbf{z}_j)|(\mathbf{z}_i, \mathbf{z}_j) \notin \mathcal{P}_{boost}\}$;
  Update encoder in Eq. 5: $\mathcal{L}_{ada} \leftarrow -\mathbb{E}\{\log \frac{\exp(\mathcal{P}_{boost}/\tau)}{\exp(\mathcal{P}_{boost}/\tau)+\exp(\mathcal{N}_{boost}/\tau)}\}, \mathcal{F} \leftarrow \mathcal{F} - \frac{\partial \mathcal{L}_{ada}}{\partial \mathcal{F}}$;
end while
return $\mathcal{F}$;

4 Experiments

4.1 Experiments setup

Data augmentation. To ensure the fairness of the experiments, we adopt the same augmentation function as in (Chen et al., 2020a). In detail, we utilize random crop (resize and random flip), color distortion, gray scale and random Gaussian blur. For training, we randomly crop and resize the raw images to 224×224, and for inference, we resize to 256×256.

Learning rate and batch size. In the first training stage of AdaCLR^{pre} which is the same to SimCLR, we adopt the same protocol of SimCLR, i.e. using LARs optimizer (You et al., 2017) and set learning rate as $0.3 \times BatchSize/256$, and the weight decay is set as 10^{-6}. While in the second stage of AdaCLR^{pre}, we set learning rate as $0.15 \times BatchSize/256$ with LARs optimizer (You et al., 2019).

Baseline and dataset. We mainly conduct experiments on ResNet-50(1x) (He et al., 2016), in addition with some extensive experiments also on ResNet(2x, 4x). The used dataset ImageNet (Deng et al., 2009) contains about 1.28 million images in 1,000 classes. Note the dataset is well-balanced in its class distribution, and its images generally contain iconic view of objects.

Dimension for evaluation. To keep the same configuration with the compared method SimCLR, the dimension of encoder output is set 2048, and the projection of non-linear transformation's output is set as 256. We evaluate the model by a linear layer as the output of the network.

4.2 Experiment results

Overall comparison with state-of-the-art. Table 1 shows the linear evaluation of top-1 and top-5 accuracy on ImageNet with different models. In the first stage, the initialized encoder is trained with $\mathcal{L}_{info}$ in 1,000 epochs, then we re-train the encoder with $\mathcal{L}_{ada}$ using 100 epochs.

We compare our AdaCLR^{pre} with two kinds of supervised models. One is ResNet-50 on ImageNet with standard cross-entropy loss, and the other one is trained with supervised contrastive loss. To ensure the fairness of data, we use the same augmentation of AdaCLR^{pre}. Previous explorations have shown that the model with stronger augmentation and larger epochs is not better than normal aug-

Table 1: Results on ImageNet with ResNet-50 as backbone. The Top-1 and Top-5 accuracy are under linear evaluation i.e. by a linear classifier for final classification (the same as in SimCLR).

Method	Backbone	Parameters(M)	Top-1	Top-5	Label
Sup_cross_entropy (He et al., 2016)	ResNet-50	24	78.3	93.9	✓
Sup_contrastive (Khosla et al., 2020)	ResNet-50	24	78.8	92.3	✓
Moco (He et al., 2020)	ResNet-50	24	60.6	-	×
Moco (He et al., 2020)	ResNet-50 (4x)	375	68.9	-	×
CPC v2 (Hénaff et al., 2019)	ResNet-50	24	63.8	85.3	×
CPC v2 (Hénaff et al., 2019)	ResNet-161	305	71.5	90.1	×
RotNet (Gidaris et al., 2018)	ResNet (4x)	87	55.4	-	×
BYOL (Grill et al., 2020)	ResNet-50	24	74.3	91.1	×
SimCLR (Chen et al., 2020a)	ResNet-50	24	69.3	89.0	×
Moco v2 (Chen et al., 2020c)	ResNet-50	24	71.1	-	×
SimCLR v2 (Chen et al., 2020b)	ResNet-50	35	77.5	93.4	✓
AdaCLR^{pre}	ResNet-50	24	72.3	91.5	×
SimCLR (Chen et al., 2020a)	ResNet-50 (2x)	94	74.2	92.0	×
SimCLR (Chen et al., 2020a)	ResNet-50 (4x)	375	76.1	93.2	×
AdaCLR^{pre}	ResNet-50 (2x)	94	75.9	92.7	×
AdaCLR^{pre}	ResNet-50 (4x)	375	**77.3**	93.7	×

Table 2: Results of different K and batch sizes. When $K = 0$, AdaCLR^{pre} degenerates to SimCLR.

Batch size / K	0	1	2	3	4	5	6	7	8	9	10
1024	67.1	68.6	69.4	69.9	70.1	**70.8**	70.4	69.8	69.2	68.8	68.1
2048	68.3	68.8	69.6	70.4	71.2	**71.7**	71.6	70.3	70.1	69.6	69.4
4096	69.4	69.6	69.7	70.7	71.7	72.0	**72.3**	71.2	70.6	70.3	69.9

Table 3: Results with different batch sizes. It involves all different ways of augmentation.

Model / Batch size	32	64	128	256	512	1024	2048	4096
AdaCLR^{pre} ($K = 1$)	60.3	61.1	63.8	66.2	**67.8**	68.6	68.8	69.6
AdaCLR^{pre} ($K = 5$)	58.7	60.3	61.4	65.1	66.9	**70.8**	**71.7**	**72.0**
SimCLR (1,000 epochs)	62.4	63.8	65.1	66.4	67.1	67.9	68.2	69.3
SimCLR (1,100 epochs)	62.5	64.0	65.1	66.5	67.2	68.0	68.3	69.4
SimCLR (1,600 epochs)	**62.8**	**64.4**	**65.3**	**66.9**	67.6	68.1	68.5	69.8

mentation and smaller epochs for cross-entropy loss. Thus, we just train supervised cross-entropy loss with 100 epochs. For supervised contrastive learning, the original work set batch size as 8192, due to the limitation of GPUs, we just test it on 4096, and the Top-1 accuracy score is 78.3$\%$. Since the AdaCLR^{pre} is a boosting-like method. The amount of parameters is the same with SimCLR. SimCLR v2 adopts deeper projection head and use 10$\%$ labeled data.

Ablation study on batch size and K values. In Section 3.3 we have analyzed that the setting K should match with mini-batch size. As in our AdaCLR^{pre} the label information is unavailable, it is unknown which images in a mini-batch belong to one category, the batch size can greatly impact the final accuracy.

For comparing different K in our AdaCLR^{pre}, at its first training stage, we train an encoder with 1,000 epochs and batch size 4096, which achieves 69.3% accuracy. In the second stage, we train the pre-trained encoder with extra 100 epochs by different batch sizes and K values.

We compare in Table 2 with different settings of batch sizes and K values. When the batch size is 1024 and K is 10, the accuracy decreases and we conjecture this may be due to the reason that AdaCLR^{pre} regard images belonging to different classes as the positives. While when the batch size is 4096 and K is set as 5, the accuracy has been significantly improved.

Table 4: Results with different augmentations functions. Note R-Color, R-Gray, R-Flip means the removal of color, gray scale, and horizontal flip, respectively. Here the batch size is set 4096.

Model / Augmentation	Color	Color & Blur	R-Color	R-Gray	R-Flip	Total
AdaCLR^{pre} ($K=1$)	47.7	53.9	55.8	68.3	69.1	69.6
AdaCLR^{pre} ($K=5$)	**51.9**	**59.7**	**61.7**	**69.1**	**71.3**	**72.0**
SimCLR (1,000 epochs)	41.8	47.6	51.3	65.7	67.3	69.3
SimCLR (1,100 epochs)	42.2	47.9	51.6	65.8	67.4	69.4
SimCLR (1,600 epochs)	42.8	48.6	52.8	66.1	67.7	69.8

4.3 Further Ablation Study

The size of mini-batch and that of the augmentation domain can greatly influence the accuracy. We compare AdaCLR and SimCLR with different settings of augmentation and batch size.

Batch size. For contrastive methods, the main way that batch size affects the performance is whether it can provide enough negative pairs. For AdaCLR, we conjecture that the batch size will influence both negative and positive pairs. In detail, with smaller batch size, AdaCLR can recognize negative pairs as positives with larger probability. Thus, AdaCLR^{pre} is more sensitive than SimCLR to batch size. To verify this, we first train SimCLR with $\mathcal{L}_{info}$ by 1,000 epochs, and the batch size is set 32 to 4096. Then, we re-train the encoder with $\mathcal{L}_{ada}$ of the same batch size by 100 epochs.

Data augmentations. It is interesting that when expanding the domain of augmentations, the performance can greatly improve. Stronger augmentation functions can cause larger area of augmentation domain and each point in this area can provide a little contribution. Then, the hard positives can be sampled with higher probability. In other words, the hard positives play an important role in contrastive learning. AdaCLR^{pre} can mine hard positive samples automatically, since the positive pairs can not only be generated by the same images but also different images in one category. Thus, AdaCLR^{pre} is less sensitive than SimCLR to data augmentation. Same with the experiments in batch size, we evaluate them with a few augmentation ways: $\{$*Crop only, Crop and blur, remove color, remove gray-scale and remove horizontal flip*$\}$.

Table 3 and Table 4 show the accuracy with different batch sizes and augmentations functions. When the batch size is larger than 512, the performance of AdaCLR^{pre} improves after 100 epochs of training. We conjecture that some samples augmented from different categories should also belong to the positives, and our methods can exactly mine such positives. Besides, the proposed adaptive algorithm can be recognized as a way to expand the distribution of positive pairs. Thus, AdaCLR is more robust than SimCLR.

5 Conclusion

In this paper, we have proposed a novel self-supervised method called AdaCLR with its two variants with pretraining based on SimCLR: AdaCLR^{pre}. It is difficult to mine hard positives for existing contrastive self-supervised models, while our model provides an adaptive technique. We theoretically show that AdaCLR can combine the properties of semi-supervised learning and self-supervised learning, leading to the improved generalization ability. Moreover, we further theoretically show why the convergence rate of AdaCLR^{pre}. Finally, we evaluate our approaches on ImageNet and the experimental results are consistent with our theoretical analysis.

References

Humam Alwassel, Dhruv Mahajan, Lorenzo Torresani, Bernard Ghanem, and Du Tran. Self-supervised learning by cross-modal audio-video clustering. *arXiv preprint arXiv:1911.12667*, 2019.

Eric B Baum and Frank Wilczek. Supervised learning of probability distributions by neural networks. In *NIPS*, pp. 52–61, 1988.

Avrim Blum and Tom Mitchell. Combining labeled and unlabeled data with co-training. In *COLT*, pp. 92–100, 1998.

Mathilde Caron, Piotr Bojanowski, Armand Joulin, and Matthijs Douze. Deep clustering for unsupervised learning of visual features. In *ECCV*, pp. 132–149, 2018.

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. *arXiv preprint arXiv:2002.05709*, 2020a.

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*, 2020b.

Xinlei Chen, Haoqi Fan, Ross Girshick, and Kaiming He. Improved baselines with momentum contrastive learning. *arXiv preprint arXiv:2003.04297*, 2020c.

Sanjoy Dasgupta, Michael L Littman, and David A McAllester. Pac generalization bounds for co-training. In *NIPS*, pp. 375–382, 2002.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *CVPR*, pp. 248–255, 2009.

Carl Doersch, Abhinav Gupta, and Alexei A Efros. Unsupervised visual representation learning by context prediction. In *Proceedings of the IEEE international conference on computer vision*, pp. 1422–1430, 2015.

Alexey Dosovitskiy, Jost Tobias Springenberg, Martin Riedmiller, and Thomas Brox. Discriminative unsupervised feature learning with convolutional neural networks. In *NIPS*, pp. 766–774, 2014.

Ran El-Yaniv and Dmitry Pechyony. Transductive rademacher complexity and its applications. *Journal of Artificial Intelligence Research*, 35:193–234, 2009.

Spyros Gidaris, Praveer Singh, and Nikos Komodakis. Unsupervised representation learning by predicting image rotations. *arXiv preprint arXiv:1803.07728*, 2018.

Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *NIPS*, pp. 2672–2680, 2014.

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, et al. Bootstrap your own latent: A new approach to self-supervised learning. *arXiv preprint arXiv:2006.07733*, 2020.

Michael Gutmann and Aapo Hyvärinen. Noise-contrastive estimation: A new estimation principle for unnormalized statistical models. In *AISTATS*, pp. 297–304, 2010.

Raia Hadsell, Sumit Chopra, and Yann LeCun. Dimensionality reduction by learning an invariant mapping. In *CVPR*, 2006a.

Raia Hadsell, Sumit Chopra, and Yann LeCun. Dimensionality reduction by learning an invariant mapping. In *CVPR*, volume 2, pp. 1735–1742, 2006b.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, pp. 770–778, 2016.

Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *CVPR*, pp. 9729–9738, 2020.

Olivier J Hénaff, Aravind Srinivas, Jeffrey De Fauw, Ali Razavi, Carl Doersch, SM Eslami, and Aaron van den Oord. Data-efficient image recognition with contrastive predictive coding. *arXiv preprint arXiv:1905.09272*, 2019.

Simon Jenni and Paolo Favaro. Self-supervised feature learning by learning to spot artifacts. In *CVPR*, pp. 2733–2742, 2018.

Jing Jiang and Chengxiang Zhai. Instance weighting for domain adaptation in nlp. In *ACL*, pp. 264–271, 2007.

Prannay Khosla, Piotr Teterwak, Chen Wang, Aaron Sarna, Yonglong Tian, Phillip Isola, Aaron Maschinot, Ce Liu, and Dilip Krishnan. Supervised contrastive learning. *arXiv preprint arXiv:2004.11362*, 2020.

Abhishek Kumar and Hal Daumé. A co-training approach for multi-view spectral clustering. In *ICML*, pp. 393–400, 2011.

Zihang Lai and Weidi Xie. Self-supervised learning for video correspondence flow. *arXiv preprint arXiv:1905.00875*, 2019.

Esther Levin and Michael Fleisher. Accelerated learning in layered neural networks. *Complex Systems*, 2(625-640):3, 1988.

Subhransu Maji, Esa Rahtu, Juho Kannala, Matthew Blaschko, and Andrea Vedaldi. Fine-grained visual classification of aircraft. *arXiv preprint arXiv:1306.5151*, 2013.

Kamal Nigam and Rayid Ghani. Analyzing the effectiveness and applicability of co-training. In *CIKM*, pp. 86–93, 2000.

Mehdi Noroozi and Paolo Favaro. Unsupervised learning of visual representations by solving jigsaw puzzles. In *ECCV*, pp. 69–84, 2016.

Mehdi Noroozi, Hamed Pirsiavash, and Paolo Favaro. Representation learning by learning to count. In *ICCV*, pp. 5898–5906, 2017.

Mehdi Noroozi, Ananth Vinjimoor, Paolo Favaro, and Hamed Pirsiavash. Boosting self-supervised learning via knowledge transfer. In *CVPR*, pp. 9359–9367, 2018.

Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018.

Deepak Pathak, Philipp Krahenbuhl, Jeff Donahue, Trevor Darrell, and Alexei A Efros. Context encoders: Feature learning by inpainting. In *CVPR*, pp. 2536–2544, 2016.

David E Rumelhart, Geoffrey E Hinton, and Ronald J Williams. Learning representations by back-propagating errors. *Nature*, 323(6088):533–536, 1986.

Kai Ming Ting. An instance-weighting method to induce cost-sensitive trees. *IEEE Transactions on Knowledge and Data Engineering*, 14(3):659–665, 2002.

Zhirong Wu, Yuanjun Xiong, Stella X Yu, and Dahua Lin. Unsupervised feature learning via non-parametric instance discrimination. In *CVPR*, pp. 3733–3742, 2018.

Yang You, Igor Gitman, and Boris Ginsburg. Large batch training of convolutional networks. *arXiv preprint arXiv:1708.03888*, 2017.

Yang You, Jing Li, Sashank Reddi, Jonathan Hseu, Sanjiv Kumar, Srinadh Bhojanapalli, Xiaodan Song, James Demmel, Kurt Keutzer, and Cho-Jui Hsieh. Large batch optimization for deep learning: Training bert in 76 minutes. *arXiv preprint arXiv:1904.00962*, 2019.

Xiaohua Zhai, Avital Oliver, Alexander Kolesnikov, and Lucas Beyer. S4l: Self-supervised semi-supervised learning. In *ICCV*, pp. 1476–1485, 2019.

Richard Zhang, Phillip Isola, and Alexei A Efros. Colorful image colorization. In *ECCV*, pp. 649–666, 2016.