AdCo: Adversarial Contrast for Efficient Learning of Unsupervised Representations from Self-Trained Negative Adversaries

Qianjiang Hu[1,*], Xiao Wang[2*], Wei Hu[1†], Guo-Jun Qi[3†]
[1]Peking University, [2]Purdue University
[3]Laboratory for MAchine Perception and LEarning (MAPLE)
http://maple-lab.net/

Abstract

Contrastive learning relies on constructing a collection of negative examples that are sufficiently hard to discriminate against positive queries when their representations are self-trained. Existing contrastive learning methods either maintain a queue of negative samples over minibatches while only a small portion of them are updated in an iteration, or only use the other examples from the current minibatch as negatives. They could not closely track the change of the learned representation over iterations by updating the entire queue as a whole, or discard the useful information from the past minibatches. Alternatively, we present to directly learn a set of negative adversaries playing against the self-trained representation. Two players, the representation network and negative adversaries, are alternately updated to obtain the most challenging negative examples against which the representation of positive queries will be trained to discriminate. We further show that the negative adversaries are updated towards a weighted combination of positive queries by maximizing the adversarial contrastive loss, thereby allowing them to closely track the change of representations over time. Experiment results demonstrate the proposed Adversarial Contrastive (AdCo) model not only achieves superior performances (a top-1 accuracy of 73.2% over 200 epochs and 75.7% over 800 epochs with linear evaluation on ImageNet), but also can be pre-trained more efficiently with much shorter GPU time and fewer epochs. The source code is available at https://github.com/maple-research-lab/AdCo.

1. Introduction

Learning visual representations in an unsupervised fashion [1, 31, 10, 23, 24, 39] has attracted many attentions as it greatly reduces the cost of collecting a large volume of labeled data to train deep networks. Significant progresses have also been made to reduce the performance gap with the fully supervised models. Among them are a family of contrastive learning methods [19, 7, 38, 21, 29, 3, 41, 36, 20] that self-train deep networks by distinguishing the representation of positive queries from their negative counterparts. Depending on how the negative samples are constructed, two large types of contrastive learning approaches have been proposed in literature [19, 7]. These negative samples play a critical role in contrastive learning since the success in self-training deep networks relies on how positive queries can be effectively distinguished from negative examples.

Specifically, one type of contrastive learning methods explicitly maintains a queue of negative examples from the past minibatches. For example, the Momentum Contrast (MoCo) [19] iteratively updates an underlying queue with the representations from the current minibatch in a First-In-First-Out (FIFO) fashion. However, only a small portion of oldest negative samples in the queue would be updated, which could not continuously track the rapid change of the feature representations over iterations. Even worse, the momentum update of key encoders, which is necessary to stabilize the negative queue in MoCo, could further slow down the track of the representations. Consequently, this would inefficiently train the representation network, since partially updated negatives may not cover all critically challenging samples thus far that ought to be distinguished from positive queries to train the network.

Alternatively, another type of contrastive learning methods [7] abandons the use of such a separate queue of negative examples. Instead, all negative examples come from the current minibatch, and a positive query would be retrieved by distinguishing it from the other examples in the minibatch. However, it discards the negative examples from the past minibatches, and often requires a much larger size of minibatch so that a sufficient number of negative samples are available to train the representation network by discriminating against positive queries. This incurs heavy memory

*Q. Hu and X. Wang made an equal contribution to performing experiments while being mentored by G.-J. Qi.

†Corresponding author: W. Hu and G.-J. Qi (guojunq@gmail.com). G.-J. Qi conceived the idea, formulated the method, and wrote the paper.

and computing burden to train over each minibatch for this type of contrastive learning.

In this paper, we are motivated to address the aforementioned drawbacks, and the objective is twofold. First, we wish to construct a set of negative examples that can continuously track the change of the learned representation rather than updating only a small portion of them. In particular, it will update negative samples *as a whole* by making them sufficiently challenging to train a representation network more efficiently with fewer epochs. On the other hand, it will retain the discriminative information from the past iterations without depending on a much larger size of minibatch to train the network [7]. We will show that in the proposed model, the negative examples are directly *trainable* so that they can be integrated as a part of the underlying network and trained end-to-end together with the representation. Thus, the trainable negatives are analogous to the additional network component in other self-supervised models without involving negative examples, such as the prediction MLP of the BYOL [17] that also needs to be trained end-to-end. More discussions on *whether we still need negative examples* can be found in Appendix C.

Particularly, we will present an Adversarial Contrast (AdCo) model consisting of two adversarial players. One is a backbone representation network that encodes the representation of input samples. The other is a collection of negative adversaries that are used to discriminate against positive queries over a minibatch. Two players are alternately updated. With a fixed set of negative adversaries, the network backbone is trained by minimizing the contrastive loss of mistakenly assigning positive queries to negative samples as in the conventional contrastive learning. On the other hand, the negative adversaries are updated by *maximizing* the contrastive loss, which pushes the negative samples to closely track the positive queries over the current minibatch. This results in a minimax problem to find an equilibrium as its saddle point solution. Although there is no theoretical guarantee of convergence, iterative gradient updates to the network backbone and the negative adversaries work well in practice, which has been observed in many other adversarial methods [16, 32, 35, 40]. We will also show that the derivative of the contrastive loss wrt the negative adversaries reveals how they are updated towards a weighted combination of positive queries, and gives us an insight into how the AdCo focuses on low-density queries compared with an alternative model.

The experiment results not only demonstrate the AdCo has the superior performance on downstream tasks, but also verify that it can train the unsupervised networks with fewer epochs by updating the negative adversaries more efficiently. For example, with merely 10 epochs of pretraining, the AdCo has a top-1 accuracy of 44.4%, which is almost 5% higher than that of the MoCo v2 pretrained over the same number of epochs with the linear evaluation on the ResNet-50 backbone on ImageNet. It also greatly outperforms the MoCHi [22] by 4.1% in top-1 accuracy over 800 epochs that enhances the MoCo v2 with mixed hard negatives, showing its effectiveness in constructing more challenging negative adversaries in a principled fashion to pretrain the representation network.

Moreover, the AdCo achieves a record top-1 accuracy of 75.7% over 800 epochs compared with the state-of-the-art BYOL (74.3%) and SWAV (75.3%) models pretrained for $800 \sim 1,000$ epochs. This is obtained with the same or even smaller amount of GPU time than the two top-performing models. Indeed, the AdCo is computationally efficient with a negligible cost of updating negative adversaries, making it an attractive paradigm of contrast model having higher accuracies over fewer epochs with no increase in the computing cost.

The remainder of the paper is organized as follows. We will review the related works in Section 2, and present the proposed approach in Section 3. By comparing the proposed AdCo with an alternative form of adversarial contrastive loss, we will reveal how AdCo focuses on low-density queries whose representations have not been well captured in Section 4. We will conduct experiments in Section 5 to demonstrate its superior performance in multiple tasks. Finally, we will conclude in Section 6.

2. Related Works

In this section, we review the related works on unsupervised representation learning. In particular, we will review two large families of unsupervised learning methods: the contrastive learning that is directly related with the proposed Adversarial Contrastive Learning (AdCo), and the alternative approach that instead aims to learns the transformation equivariant features without labeled data. We will see that while the former explores the *inter-instance discrimination* to self-supervise the training of deep networks, the latter leverages *intra-instance variation* to learn transformation-equivariant representations.

2.1. Contrastive Learning

Originally, contrastive learning [29, 20] was proposed to learn unsupervised representations by maximizing the mutual information between the learned representation and a particular context [3, 41, 36]. It usually uses the context of the same instance to learn representations by discriminating between positive queries and a collection of negative examples in an embedding feature space [38, 18, 7]. Among them is the instance discrimination that has been used as a pretext task by distinguishing augmented samples from each other in a minimatch [7], over a memory bank [38], or a dynamic queue [18] with a momentum update. The retrieval of a given query is usually performed by matching it against an augmentation of the same sample from a separate collection of negative examples. These negative examples play a critical role to challenge the self-trained representation so that it can become gradually more discriminative over iterations to distinguish the presented queries from those hard negative examples. Recently, Grill et al. [17] propose to train deep networks by predicting the representation of an image from that of an different augmented view of the same image. Kalantidis et al. [22] present a feature-level sample mixing strategies to generate harder meaningful negatives to further improve network pretraining. However, this differs from the proposed AdCo that aims to update negative examples in a more principle way through an adversarial self-training mechanism.

2.2. Other Unsupervised Methods

An alternative approach to unsupervised representation learning is based on transformation prediction [39, 15]. On contrary to contrastive learning that focuses on inter-instance discrimination, it attempts to learn the representations equivarying to various transformations on 2D plenary images and 3D cloud points [34, 31, 13]. These transformations are applied to augment training examples and learn their representations from which the transformations can be predicted from the captured visual structures. It focuses on modeling the *intra-instance variations* from which transformation-equivariant representations can be leveraged on downstream tasks such as image classification [39, 31, 15], object detection [15, 31], semantic segmentations [15, 30] and 3D cloud points [13]. This category of methods can be viewed as orthogonal to contrastive learning approaches that are based on *inter-instance discrimination*. More comprehensive review of recent advances on unsupervised methods can be found in [33].

3. The Proposed Approach

In this section, we will first briefly review the preliminary work on contrastive learning and discuss its drawbacks that motivate the proposed approach in Section 3.1. Then, we will present the proposed method in Section 3.2 to jointly train the representation network and negative adversaries. The insight into the proposed approach can be better revealed from the derivative of the adversarial contrastive loss leading to the update rule for the negative samples in Section 3.3.

3.1. Preliminaries and Motivations

We begin by briefly revisiting contrastive learning and its variants, and discuss their limitations that motivate the proposed work. In a typical contrastive learning method, we seek to learn an unsupervised representation by minimizing a contrastive loss $\mathcal{L}$. Specifically, in a minibatch $\mathcal{B}$ of N samples, consider a given query $\mathbf{x}_i$ and the embedding $\mathbf{q}_i$ of its augmentation through a backbone network with the parametric weights θ. The contrastive learning aims to train the network θ such that the query q can be distinguished from a set $\mathcal{N} = \{\mathbf{n}_k | k = 1, \cdots, K\}$ of the representations of negative samples. Formally, the following contrastive loss is presented in InfoNCE [29] in a soft-max form,

$$\mathcal{L}(\theta) = \frac{-1}{N} \sum_{i=1}^{N} \log \frac{\exp(\mathbf{q}_i^\intercal \mathbf{q}_i'/\tau)}{\exp(\mathbf{q}_i^\intercal \mathbf{q}_i'/\tau) + \sum_{k=1}^{K} \exp(\mathbf{q}_i^\intercal \mathbf{n}_k/\tau)} \quad (1)$$

where $\mathbf{q}_i'$ is the embedding of another augmentation of the same instance $\mathbf{x}_i$, which is considered as the positive example for the query $\mathbf{q}_i$; and τ is a positive value of temperature. Then the network can be updated in each minibatch by minimizing this loss over θ. Note that the representations of both queries and negative samples are ℓ_2 normalized such that the dot product results in a cosine similarity between them.

There are two main categories of methods about how to construct the set of negative samples $\mathcal{N}$ in contrast to positive queries. These negative samples play a critical role in self-training unsupervised representations, and deserve a careful investigation. The first category of contrastive learning approaches maintain a queue of negative representations that are iteratively updated in a First-In-First-Out (FIFO) fashion over minibatches [18], while the alternative type of approaches only adopt the other samples as negatives from the current minibatch [7].

However, both types suffer some drawbacks. For the first type of approaches, in each iteration over a minibatch, only a small portion of negative representations in the queue are updated, and this results in an under-represented set that fails to cover the most critical negative samples to train the representation network, since not all of them have been updated timely to track the change of unsupervised representations. On the other hand, the other type completely abandons the negative samples from the past minibatches, which forces them to adopt a much larger size of minibatch to construct a sufficient number of negative representations. Moreover, only using the other samples from the current minibatch as negatives discards rich information from the past.

To address these drawbacks, we propose to *actively* train a set of negative examples as a whole in an adversarial fashion, in contrast to passively queueing the negative samples over minibatches or simply using the other samples from the current minibatch as negatives. Its advantages are twofold. First, the entire set rather than only a small portion of negative representations will be updated as a whole. This enables the learned negative adversaries to not only closely track the change of the learned representation but also keep the accumulative information from the past minibatches.

Moreover, the set of negative samples will act as adversaries to the representation network θ by *maximizing* the contrastive loss, which will result in a minimax problem to self-train the network. As known in literature, a selfsupervisory objective ought to be sufficiently challenging to prevent the learned representation from overfitting to the objective or finding a bypassing solution that could be trivial to downstream tasks. In this sense, the adversarially learned negative set can improve the generalizability of the learned representation as an explicit adversary to challenge the updated network by engaging the most challenging negative samples over epochs.

3.2. Adversarial Training of Representation Networks and Negative Samples

In this section, we formally present the proposed Adversarial Contrastive Learning (AdCo) of unsupervised representations. It consists of two mutually interacted players: a deep representation network θ embedding input examples into feature vectors, and a set $\mathcal{N}$ of negative adversaries closely tracking the change of the learned representation. While negative adversaries contain the most critical examples that are supposed to be confused with given queries to challenge the representation network, the network is self-trained to continuously improve its discrimination ability to distinguish positive queries from those challenging negative adversaries. Eventually, we hope an equilibrium can be reached where the learned representation can achieve a maximized performance.

Formally, this results in the following minimax problem to train both players jointly with the adversarial contrastive loss $\mathcal{L}$ as in (1),

$$\theta^\star, \mathcal{N}^\star = \arg\min_{\theta}\max_{\mathcal{N}} \mathcal{L}(\theta, \mathcal{N}) \tag{2}$$

where the embedding $\mathbf{q}_i$ of each query in the current minibatch are a function of network weights θ, and thus we optimize over θ through them; the negative adversaries in $\mathcal{N}$ are treated as free variables directly, which are unit-norm vectors subject to ℓ_2-normalization.

As in many existing adversarial training algorithms, it is hard to find a saddle point $(\theta^\star, \mathcal{N}^\star)$ solution to the above minimax problem. Usually, a pair of gradient descent and ascent are applied to update them, respectively

$$\theta \longleftarrow \theta - \eta_\theta \frac{\partial \mathcal{L}(\theta, \mathcal{N})}{\partial \theta} \tag{3}$$

$$\mathbf{n}_k \longleftarrow \mathbf{n}_k + \eta_{\mathcal{N}} \frac{\partial \mathcal{L}(\theta, \mathcal{N})}{\partial \mathbf{n}_k} \text{[1]} \tag{4}$$

for $k = 1, \cdots, K$, where η_θ and $\eta_{\mathcal{N}}$ are the positive learning rates for updating the network and negative adversaries. Although no theory guarantees the convergence to the saddle point, it works well in our experiments by alternately updating θ and $\mathcal{N}$.

[1]Strictly speaking, the update of $\mathbf{n}_k$ ought to be performed by taking the derivative of $\mathcal{L}$ wrt the one prior to ℓ_2-normalized. However, to ease the exposition later, we simply adopt the derivative wrt the normalized embedding without the loss of generality.

Before we take an insight into the updated $\mathbf{n}_k$'s, an intuitive explanation can be given by noticing that the cosine similarities between the negative samples $\mathbf{n}_k$'s and the queries $\mathbf{q}_i$ in the denominator of (1) are maximized when $\mathcal{L}(\theta, \mathcal{N})$ is maximized for the adversarial training of $\mathcal{N}$. In other words, this tends to push the negative samples closer towards the queries from the current minibatch, thus resulting in challenging negatives closely tracking the change of the updated network.

3.3. Derivatives of The AdCo Loss

Let us look into the update rule (4) for the negative samples $\mathbf{n}_k, k = 1, \cdots, K$. It is not hard to show that the derivative of the adversarial loss in updating a negative sample $\mathbf{n}_k$ is

$$\frac{\partial \mathcal{L}}{\partial \mathbf{n}_k} = \frac{1}{N\tau} \sum_{i=1}^{N} \frac{\exp(\mathbf{q}_i^\intercal \mathbf{n}_k/\tau) \cdot \mathbf{q}_i}{\exp(\mathbf{q}_i^\intercal \mathbf{q}_i'/\tau) + \sum_{k=1}^{K} \exp(\mathbf{q}_i^\intercal \mathbf{n}_k/\tau)} \tag{5}$$

where the first factor can be viewed as the conditional probability of assigning the query $\mathbf{q}_i$ to a negative sample $\mathbf{n}_k$,

$$p(\mathbf{n}_k|\mathbf{q}_i) \triangleq \frac{\exp(\mathbf{q}_i^\intercal \mathbf{n}_k/\tau)}{\exp(\mathbf{q}_i^\intercal \mathbf{q}_i'/\tau) + \sum_{k=1}^{K} \exp(\mathbf{q}_i^\intercal \mathbf{n}_k/\tau)}$$

This is a valid probability because it is always nonnegative and the sum of all such conditional probabilities is one, i.e.,

$$p(\mathbf{q}_i'|\mathbf{q}_i) + \sum_{k=1}^{K} p(\mathbf{n}_k|\mathbf{q}_i) = 1.$$

Now the derivative (5) can be rewritten as

$$\frac{\partial \mathcal{L}}{\partial \mathbf{n}_k} = \frac{1}{N\tau} \sum_{i=1}^{N} p(\mathbf{n}_k|\mathbf{q}_i)\, \mathbf{q}_i. \tag{6}$$

This reveals the physical meaning by updating each negative sample $\mathbf{n}_k$ along the direction given by a $p(\mathbf{n}_k|\mathbf{q}_i)$-weighted combination of all the queries $\mathbf{q}_i$ over the current minibatch. The more likely a negative sample is assigned to a query, the closer the sample is pushed towards the query. This will force the negative samples to continuously track the queries that are most difficult to distinguish, and thus give rise to a challenging set of adversarial negative samples to more critically train the underlying representation. Moreover, this adversarial update also tends to cover low-density queries that have not been well captured by the current set of negative samples, which we will discuss in the next section.

4. Further Discussions

In this section, first we will present an alternative of the proposed AdCo in Section 4.1. Then, based on it, we will review the derivative of the AdCo loss in Section 4.2, and show that it seeks to adapt negative adversaries to low-density queries, thereby allowing more efficient adversarial training of the representation network.

4.1. An Alternative of AdCo

An alternative adversarial loss can be motivated from (6). One can adopt another form of conditional probability as the weighting factor by constructing a new adversarial loss $\mathcal{J}(\theta,\mathcal{N})$ to train the negative adversaries. In particular, we may consider the conditional probability $p(\mathbf{q}_i|\mathbf{n}_k)$ in place of $p(\mathbf{n}_k|\mathbf{q}_i)$ used in (6). This turns out to be a more natural choice since the derivative becomes the conditional expectation of the query $\mathbf{q}_i$'s, i.e.,

$$\frac{\partial \mathcal{J}(\theta,\mathcal{N})}{\partial \mathbf{n}_k} = \sum_{i=1}^{N} p(\mathbf{q}_i|\mathbf{n}_k)\ \mathbf{q}_i \triangleq \mathbb{E}_{p(\mathbf{q}_i|\mathbf{n}_k)}\left[\mathbf{q}_i|\mathbf{n}_k\right]. \quad (7)$$

In this case, it is not hard to verify that the corresponding adversarial loss $\mathcal{J}$ satisfying the above derivative relation can be defined as

$$\max_{\mathcal{N}} \mathcal{J}(\theta,\mathcal{N}) \triangleq -\tau \cdot \sum_{k=1}^{K} \log \frac{1}{\sum_{i=1}^{N} \exp(\mathbf{q}_i^\intercal \mathbf{n}_k/\tau)}$$

with the following conditional probability of assigning the negative sample $\mathbf{n}_k$ to a target query $\mathbf{q}_i$

$$p(\mathbf{q}_i|\mathbf{n}_k) = \frac{\exp(\mathbf{q}_i^\intercal \mathbf{n}_k/\tau)}{\sum_{i=1}^{N} \exp(\mathbf{q}_i^\intercal \mathbf{n}_k/\tau)}$$

Similarly, it is not hard to show that this is a valid conditional probability by verifying it is nonnegative and satisfies its sum being one. Then, the negative samples are updated by gradient ascent on $\mathcal{J}(\theta,\mathbf{N})$ over $\mathbf{n}_k$'s.

4.2. Adapting to Low-Density Queries

Although the derivative of this adversarial loss $\mathcal{J}$ embodies a more explicit meaning as in (7), our preliminary experiments with this loss to update negative samples perform worse than that of the AdCo in (2). This probably is due to the fact that the loss $\mathcal{J}$ does not directly adverse against the contrastive loss $\mathcal{L}$ used to train the embedding network θ. Thus, the generated negative samples may not provide sufficiently hard negative samples as direct adversaries to the contrastive loss (2) used to train the representation network.

Indeed, by applying the Bayesian rule to $p(\mathbf{n}_k|\mathbf{q}_i)$, the derivative in (6) can be rewritten as

$$\frac{\partial \mathcal{L}}{\partial \mathbf{n}_k} \propto \sum_{i=1}^{N} \frac{p(\mathbf{q}_i|\mathbf{n}_k)p(\mathbf{n}_k)}{p(\mathbf{q}_i)}\ \mathbf{q}_i = p(\mathbf{n}_k)\mathbb{E}_{p(\mathbf{q}_i|\mathbf{n}_k)}\left[\widetilde{\mathbf{q}}_i|\mathbf{n}_k\right]$$

where $\widetilde{\mathbf{q}}_i \triangleq \frac{\mathbf{q}_i}{p(\mathbf{q}_i)}$ is the embedded query $\mathbf{q}_i$ normalized by $p(\mathbf{q}_i)$, and we put $p(\mathbf{n}_k)$ outside the conditional expectation by viewing $\mathbf{n}_k$ as a constant in the condition.

It is not hard to see that for a low-density query $\mathbf{q}_i$ with a smaller value of $p(\mathbf{q}_i)$, it has a larger value of the normalized query $\widetilde{\mathbf{q}}_i$. Thus, the resultant derivatives will push negative adversaries closer to such low-density queries that have not been well represented by the negative examples [2]. This will enable more efficient training of the representation network with the negative adversaries updated to cover these under-represented queries.

For this reason, we will still use the AdCo loss (2) to jointly train the negative adversaries and the network in our experiments. However, the loss (7) derived from the conditional expectation provides an insight into some future direction to alterative forms of adversarial loss to train negative samples. While this gives us an alternative explanation of the proposed method, it could have some theoretical value that deserves further study in future.

5. Experiments

In this section, we conduct experiments on the proposed AdCo model and compare it with the other existing unsupervised methods.

5.1. Training Details

For the sake of fair and direct comparison with the existing models [18, 7], we adopt the ResNet-50 as the backbone for unsupervised pretraining on ImageNet. The output feature map from the top ResNet-50 block is average-pooled and projects to a 128-D feature vector through two-layer MLP (2048-D hidden lyaer with the ReLU) [7]. The resultant vector is ℓ_2 normalized to calculate the cosine similarity. We apply the same augmentation proposed in the SimCLR [7] and adopted by the other methods [18, 8] to augment the images over each minibatch. Although it was reported in literature that carefully tuning the augmentation may improve the performance, we do not adopt it for a fair comparison with the other unsupervised methods while avoiding over-tuning the image augmentation strategy on the dataset.

For the unsupervised pretraining on ImageNet, we use the SGD optimizer ([4]) with an initial learning rate of 0.03 and 3.0 for updating the backbone network and the negative adversaries, respectively, where a weight decay of 10^{-4} and a momentum of 0.9 are also applied. Cosine scheduler ([27]) is used to gradually decay the learning rate. We also choose a lower temperate ($\tau = 0.02$) to update the negative adversaries than that ($\tau = 0.12$) used for updating the backbone network. This makes the updated negative adversaries sharper in distinguishing themselves from the positive images, and thus they will be nontrivially challenging examples in training the network.

The batch size is set to 256. When multiple GPU servers are used, the batch size will be multiplied by the num-

[2]From $p(\mathbf{q}_i) = \sum_{k=1}^{K} p(\mathbf{q}_i|\mathbf{n}_k)p(\mathbf{n}_k)$, a low-density query $\mathbf{q}_i$ should have a small value of $p(\mathbf{q}_i|\mathbf{n}_k)$ for high-density negative adversaries $\mathbf{n}_k$'s, which means the query should be far apart from the dense negative examples in the embedding space.

Table 1: Top-1 accuracy under the linear evaluation on ImageNet with the ResNet-50 backbone. The table compares the methods over 200 epochs of pretraining.

Method	Batch size	Top-1
InstDisc [38]	256	58.5
LocalAgg [41]	128	58.8
MoCo [19]	256	60.8
SimCLR [7]	256	61.9
CPC v2 [20]	512	63.8
PCL v2 [25]	256	67.6
MoCo v2 [8]	256	67.5
MoCHi [22]	512	68.0
PIC [5]	512	67.6
SWAV* [6]	256	72.7
AdCo	256	**68.6**
AdCo*	256	**73.2**

* with multi-crop augmentations.

Table 2: Running time (in hours) for different methods. The last column shows the total GPU time per epoch.

Method	Epoch	GPU	(GPU·Time)/Epoch
MoCo v2 [8]	200	8×V100	2.12
BYOL [17]	1000	512×TPU	4.10
SWAV* [6]	200	64×V100	4.06
AdCo	200	8×V100	2.26
AdCo*	200	8×V100	4.39

* with multi-crop augmentations.

ber of servers by convention. The number of negative adversaries is set to $65,536$, which is the same as the queue length of negative examples in MoCo [18]. All negative adversaries are also ℓ_2-normalized in each iteration to have an unit norm after they are updated by the SGD optimizer. The backbone network is first initialized, and its output feature vectors over randomly drawn training images are used to initialize the set of negative examples. Once the initialization is done, both the representation network and negative adversaries will be alternately updated by the AdCo.

5.2. Experiment Results

After the backbone network is pretrained, the resultant network will be evaluated on several downstream tasks.

5.2.1 Linear Classification on ImageNet

First, we compare the proposed AdCo with the other methods in terms of top-1 accuracy. A linear fully connected classifier is fine-tuned on top of the frozen 2048-D feature vector out of the pretrained ResNet-50 backbone. The linear layer is trained for 100 epochs, with a learning rate of 10 with the cosine learning rate decay and a batch size of 256.

Table 3: Top-1 accuracy under the linear evaluation on ImageNet with the ResNet-50 backbone. The table compares the methods with more epochs of network pretraining.

Method	Epoch	Batch size	Top-1
Supervised	-	-	76.5
BigBiGAN[11]	-	2048	56.6
SeLa[2]	400	256	61.5
PIRL[28]	800	1024	63.6
CMC[36]	240	128	66.2
SimCLR[7]	800	4096	69.3
PIC[5]	1600	512	70.8
MoCo v2[8]	800	256	71.1
MoCHi [22]	800	512	68.7
BYOL **[17]	1000	4096	74.3
SWAV*[6]	800	4096	75.3
AdCo	800	256	**72.8**
AdCo*	800	1024	**75.7**

* with multi-crop augmentations.
** fine-tuned with a batch size of 1024 over 80 epochs.

Table 1 reports experiment results with the backbone network pretrained for 200 epochs. We note that the state-of-the-art SWAV model has applied multi-crop augmentations to pretrain the model. Thus for the sake of a fair comparison, in addition to the single-crop model, we also report the result by applying five crops (224x224, 192x192, 160x160, 128x128, and 96x96) over minibatch images during the pretraining. The result shows the proposed AdCo achieves the best performance among the compared models. It not only outperforms the SOTA SWAV model in the top-1 accuracy over 200 epochs, but also has a computing time on par with the top-performing contrast models in terms of GPU·Time per epoch as shown in Table 2. It is not surprising since the computing overhead for updating the negative adversaries is negligible compared with that for updating the backbone network.

Note that some methods such as the BYOL [17] and the SimCLR [7] used symmetric losses by swapping pairs of augmented images to improve the performance. For a fair comparison, we also symmetrize the AdCo loss, and show in Appendix A that it obtains a competitive top-1 accuracy of 70.6% over 200 epochs of pre-training based only on single-crop augmentation with much shorter GPU time and a smaller batch size. We refer the readers to the appendix for more details. In Appendix C, we also seek the answer to an emerging question – *do we still need negative samples to pre-train deep networks while the BYOL does not?* .

We also compare the results over more epochs in Table 3.

While the AdCo* has the running time on par with the other top-2 best models (SWAV and BYOL) during the pretraining (see Table 2), it has achieved a record top-1 accuracy of 75.7% with linear evaluation on the ImageNet after 800 epochs of pretraining. Even without multi-crop augmentations, the AdCo also outperforms the top-performing MoCo v2 [8] (72.8% vs. 71.1%) and the variant MoCHi (72.8% vs. 68.7%) [22] that enhances the MoCo v2 with mixed hard negative examples. This demonstrates that the AdCo is much more effective in constructing challenging negative examples to pre-train the representation network than not only the typical contrast model like the MoCo v2 but also its variant explicitly mining hard samples.

5.2.2 Transfer Learning and Object Detection

We also evaluate the ResNet-50 pretrained on ImageNet for two downstream tasks – cross-dataset transfer learning and object detection. For the transfer learning on the VOC dataset [12], we keep the pretrained ResNet-50 backbone frozen, fine-tune the linear classifier on VOC07trainval, and test on the VOC07test. We also train a linear classifier with the pretrained ResNet-50 on the Places dataset by following the previous evaluation protocol in literature [7, 8, 6].

For object detection, we adopt the same protocol in [19] to fine-tune the pretrained ResNet-50 backbone based on detectron2 [37] for a direct comparison with the other methods. On the VOC dataset [12], the detection network is fine-tuned with VOC07+12 trainval dataset and tested on VOC07 test set. On the COCO dataset [26], the detection network is fine-tuned on the train2017 dataset with 118k images, and is evaluated on the val2017 dataset.

Table 4 shows the proposed AdCo achieves much competitive results on both tasks, suggesting the AdCo has better generalizability to the downstream tasks than many compared methods. For example, it has achieved a much higher accuracy of 93.1% for the VOC07 classification task than the SWAV (88.9%) based on the linear evaluation on the pretrained network, even without multi-crop augmentations. Moreover, the COCO dataset is well known for its challenging small object detection task measured by AP_S, and the result shows that the AdCo can significantly improve the AP_S by $3.3\%-3.6\%$ no matter if multi-crop augmentations are applied. This is a striking result as the SWAV with the multi-crop augmentations is even worse than the MoCo v2. This implies that the negative adversaries constructed by the AdCo may correspond to small objects, which in turn pushes the associated network to learn the representation highly discriminative in recognizing these challenging objects.

5.3. Analysis and Visualization of Results

First, we perform a comparative study of how the top-1 accuracy of AdCo and MoCo v2 changes over epochs. As the state-of-the-art contrast model, the MoCo v2 also maintains a set of negative examples that play a critical role to train the representation network. As illustrated in Figure 1, we plot the curve of their top-1 accuracies on ImageNet with a linear evaluation on the pretrained backbone under various numbers of epochs. The same number of $K = 65,536$ negative examples are used in both models to ensure a fair comparison.

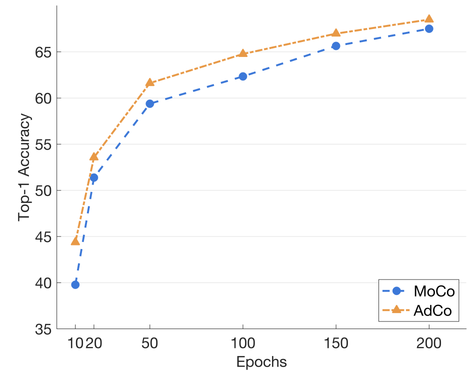


Figure 1: Top-1 accuracy of AdCo and MoCo V2 on ImageNet fine-tuned with a single fully connected layer upon the backbone ResNet-50 networks pretrained over up to 200 epochs. For a fair comparison, the AdCo maintains the same size ($K = 65,536$) of negative samples as in MoCo v2 so that both models are directly compared under various numbers of epochs. With an extremely small number of 10 epochs, the AdCo performs much better than the MoCo v2 by more than 5% in top-1 accuracy. As the epoch increases, the AdCo usually achieves the performance comparable to that of the MoCo v2 with about $30 \sim 50$ fewer epochs. This shows the AdCo is more efficient in constructing a more critical collection of negative adversaries to improve network pretraining.

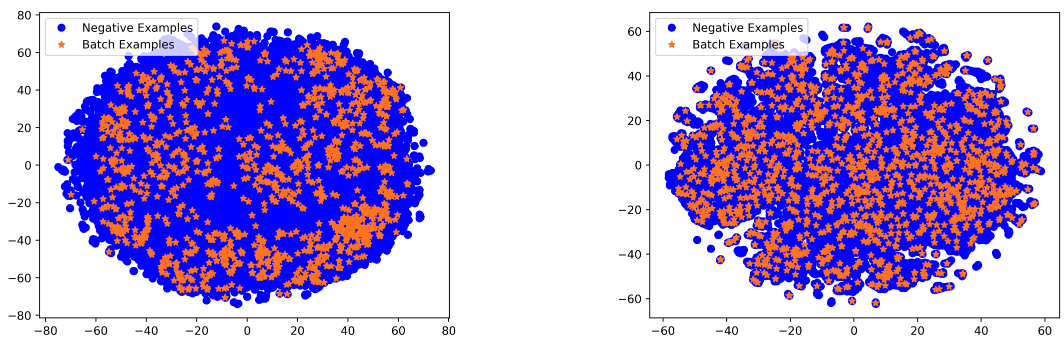
With an extreme small number of epochs, the result shows that the AdCo greatly outperforms the MoCo v2 by a significant margin. With 10 epochs the top-1 accuracy of the AdCo is 5% higher than that of the MoCo v2. With more epochs, the AdCo can reach the same level of accuracy with about $30 \sim 50$ fewer epochs than the MoCo v2. This shows the AdCo can serve as an efficient paradigm of contrast model with fewer epochs of pretraining that does not increasing the computing cost as shown in Table 2.

We also compare the obtained negative examples by the AdCo and MoCo v2 by plotting t-SNE visualization in Figure 2. We note that the MoCo v2 has more outliers of negative examples than the AdCo. These negative outliers form many small clusters that are isolated from most of batch examples in the learned representation space. Thus, they have little contributions to the contrastive training of the representation network as it is much easier to distinguish these negative outliers from positive batch examples. On the contrary, with fewer negative outliers, the AdCo can more

Table 4: Transfer learning results on cross-dataset classification and object detection tasks.

		Classification		Object Detection		
		VOC07	Places205	VOC07+12	COCO	
Method	Epoch	mAP	Top-1	AP_{50}	AP	AP_S
Supervised	-	87.5	53.2	81.3	40.8	20.1
NPID++ [38]	200	76.6	46.4	79.1	-	-
MoCo [19]	800	79.8	46.9	81.5	-	-
PIRL [28]	800	81.1	49.8	80.7	-	-
PCLv2 [25]	200	85.4	50.3	78.5	-	-
BoWNet [14]	280	79.3	51.1	81.3	-	-
SimCLR [7]	800	86.4	53.3	-	-	-
MoCo v2 [8]	800	87.1	52.9	82.5	**42.0**	20.8
SWAV* [6]	800	88.9	**56.7**	82.6	42.1	19.7
AdCo	800	**93.1**	**53.7**	**83.1**	41.8	**24.1**
AdCo*	800	**92.4**	56.0	**83.0**	**42.2**	**24.5**

* with multi-crop augmentations.



(a) AdCo

(b) MoCo v2

Figure 2: t-SNE visualization of negative representations obtained by the AdCo and the MoCo in the 2D representation plane, alongside positive examples from the most recent six minibatches after 200 epochs of pretraininig on the ImageNet. The figure shows that the AdCo has fewer outliers of negative examples than the MoCo v2, and thus more closely tracks the representation of positive samples over epochs.

closely track the representation of batch examples over iterations, and thus efficiently train the representation network with more challenging negative adversaries.

More experiment results on the impact of various model designations and hyperparameters are presented in the appendix of this paper.

6. Conclusions

This paper presents an Adversarial Contrast (AdCo) model by learning challenging negative adversaries that can be used to criticize and further improve the representation of deep networks. Compared with existing methods accumulating negative examples over the past minibatches and the other queries from the current minibatch, these negative adversaries in the AdCo are obtained by *maximizing* the adversarial contrastive loss of mis-assigning each positive query to negative samples. This updates the negative examples as a whole to closely track the changing representation, thus making it more challenging to distinguish them from positive queries. Consequently, the representation network must be efficiently updated to produce more discriminate representation. By analyzing the derivative of the adversarial objective, we show that each negative adversary is pushed towards a weighted combination of positive queries. Experiment results on multiple downstream tasks demonstrate its superior performances and the efficiency in pretraining the representation network.

References

[1] Pulkit Agrawal, Joao Carreira, and Jitendra Malik. Learning to see by moving. In *Proceedings of the IEEE international conference on computer vision*, pages 37–45, 2015. 1

[2] Yuki Markus Asano, Christian Rupprecht, and Andrea Vedaldi. Self-labelling via simultaneous clustering and representation learning. *arXiv preprint arXiv:1911.05371*, 2019. 6

[3] Philip Bachman, R Devon Hjelm, and William Buchwalter. Learning representations by maximizing mutual information across views. In *Advances in Neural Information Processing Systems*, pages 15509–15519, 2019. 1, 2

[4] Léon Bottou. Large-scale machine learning with stochastic gradient descent. In *Proceedings of COMPSTAT'2010*, pages 177–186. Springer, 2010. 5

[5] Yue Cao, Zhenda Xie, Bin Liu, Yutong Lin, Zheng Zhang, and Han Hu. Parametric instance classification for unsupervised visual feature learning. *Advances in Neural Information Processing Systems*, 33, 2020. 6

[6] Mathilde Caron, Ishan Misra, Julien Mairal, Priya Goyal, Piotr Bojanowski, and Armand Joulin. Unsupervised learning of visual features by contrasting cluster assignments. *arXiv preprint arXiv:2006.09882*, 2020. 6, 7, 8

[7] 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, 5, 6, 7, 8, 11, 12

[8] Xinlei Chen, Haoqi Fan, Ross Girshick, and Kaiming He. Improved baselines with momentum contrastive learning. *arXiv preprint arXiv:2003.04297*, 2020. 5, 6, 7, 8, 12

[9] Xinlei Chen and Kaiming He. Exploring simple siamese representation learning. *arXiv preprint arXiv:2011.10566*, 2020. 12

[10] 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*, pages 1422–1430, 2015. 1

[11] Jeff Donahue and Karen Simonyan. Large scale adversarial representation learning. In *Advances in Neural Information Processing Systems*, pages 10542–10552, 2019. 6

[12] 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. 7

[13] Xiang Gao, Wei Hu, and Guo-Jun Qi. Graphter: Unsupervised learning of graph transformation equivariant representations via auto-encoding node-wise transformations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7163–7172, 2020. 3

[14] Spyros Gidaris, Andrei Bursuc, Nikos Komodakis, Patrick Pérez, and Matthieu Cord. Learning representations by predicting bags of visual words. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6928–6938, 2020. 8

[15] Spyros Gidaris, Praveer Singh, and Nikos Komodakis. Unsupervised representation learning by predicting image rotations. *arXiv preprint arXiv:1803.07728*, 2018. 3

[16] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in neural information processing systems*, pages 2672–2680, 2014. 2

[17] 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. 2, 3, 6, 11, 12

[18] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. *arXiv preprint arXiv:1911.05722*, 2019. 2, 3, 5, 6

[19] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9729–9738, 2020. 1, 6, 7, 8

[20] 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. 1, 2, 6

[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. 1

[22] Yannis Kalantidis, Mert Bulent Sariyildiz, Noe Pion, Philippe Weinzaepfel, and Diane Larlus. Hard negative mixing for contrastive learning. *arXiv preprint arXiv:2010.01028*, 2020. 2, 3, 6, 7

[23] Dahun Kim, Donghyeon Cho, Donggeun Yoo, and In So Kweon. Learning image representations by completing damaged jigsaw puzzles. In *2018 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 793–802. IEEE, 2018. 1

[24] Gustav Larsson, Michael Maire, and Gregory Shakhnarovich. Learning representations for automatic colorization. In *European conference on computer vision*, pages 577–593. Springer, 2016. 1

[25] Junnan Li, Pan Zhou, Caiming Xiong, Richard Socher, and Steven CH Hoi. Prototypical contrastive learning of unsupervised representations. *arXiv preprint arXiv:2005.04966*, 2020. 6, 8

[26] 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 *European conference on computer vision*, pages 740–755. Springer, 2014. 7

[27] Ilya Loshchilov and Frank Hutter. Sgdr: Stochastic gradient descent with warm restarts. *arXiv preprint arXiv:1608.03983*, 2016. 5

[28] 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*, pages 6707–6717, 2020. 6, 8

[29] Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018. 1, 2, 3

[30] Guo-Jun Qi. Hierarchically gated deep networks for semantic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2267–2275, 2016. 3

[31] Guo-Jun Qi. Learning generalized transformation equivariant representations via autoencoding transformations. *arXiv preprint arXiv:1906.08628*, 2019. 1, 3

[32] Guo-Jun Qi. Loss-sensitive generative adversarial networks on lipschitz densities. *International Journal of Computer Vision*, 128(5):1118–1140, 2020. 2

[33] Guo-Jun Qi and Jiebo Luo. Small data challenges in big data era: A survey of recent progress on unsupervised and semi-supervised methods. *arXiv preprint arXiv:1903.11260*, 2019. 3

[34] Guo-Jun Qi, Liheng Zhang, Chang Wen Chen, and Qi Tian. Avt: Unsupervised learning of transformation equivariant representations by autoencoding variational transformations. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 8130–8139, 2019. 3

[35] Guo-Jun Qi, Liheng Zhang, Hao Hu, Marzieh Edraki, Jingdong Wang, and Xian-Sheng Hua. Global versus localized generative adversarial nets. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1517–1525, 2018. 2

[36] Yonglong Tian, Dilip Krishnan, and Phillip Isola. Contrastive multiview coding. *arXiv preprint arXiv:1906.05849*, 2019. 1, 2, 6

[37] Yuxin Wu, Alexander Kirillov, Francisco Massa, Wan-Yen Lo, and Ross Girshick. Detectron2. `https://github.com/facebookresearch/detectron2`, 2019. 7

[38] Zhirong Wu, Yuanjun Xiong, Stella X Yu, and Dahua Lin. Unsupervised feature learning via non-parametric instance discrimination. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3733–3742, 2018. 1, 2, 6, 8

[39] Liheng Zhang, Guo-Jun Qi, Liqiang Wang, and Jiebo Luo. Aet vs. aed: Unsupervised representation learning by autoencoding transformations rather than data. *arXiv preprint arXiv:1901.04596*, 2019. 1, 3

[40] Yiru Zhao, Zhongming Jin, Guo-jun Qi, Hongtao Lu, and Xian-sheng Hua. An adversarial approach to hard triplet generation. In *Proceedings of the European conference on computer vision (ECCV)*, pages 501–517, 2018. 2

[41] Chengxu Zhuang, Alex Lin Zhai, and Daniel Yamins. Local aggregation for unsupervised learning of visual embeddings. In *Proceedings of the IEEE conference on Computer Vision*, 2019. 1, 2, 6