Train a One-Million-Way Instance Classifier for Unsupervised Visual Representation Learning

Yu Liu, Lianghua Huang, Pan Pan, Bin Wang, Yinghui Xu, Rong Jin
Machine Intelligence Technology Lab, Alibaba Group
{ly103369, xuangen.hlh, panpan.pp, ganfu.wb, renji.xyh, jinrong.jr}@alibaba-inc.com

Abstract

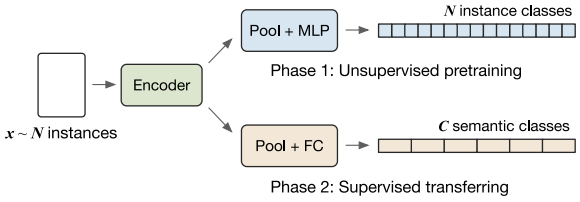
This paper presents a simple unsupervised visual representation learning method with a pretext task of discriminating all images in a dataset using a parametric, instance-level classifier. The overall framework is a replica of a supervised classification model, where *semantic classes* (e.g., *dog, bird,* and *ship*) are replaced by *instance IDs*. However, scaling up the classification task from thousands of *semantic labels* to millions of *instance labels* brings specific challenges including 1) the large-scale softmax computation; 2) the slow convergence due to the infrequent visiting of instance samples; and 3) the massive number of negative classes that can be noisy. This work presents several novel techniques to handle these difficulties. First, we introduce a hybrid parallel training framework to make large-scale training feasible. Second, we present a raw-feature initialization mechanism for classification weights, which we assume offers *a contrastive prior* for instance discrimination and can clearly speed up converge in our experiments. Finally, we propose to smooth the labels of a few hardest classes to avoid optimizing over very similar negative pairs. While being conceptually simple, our framework achieves competitive or superior performance compared to state-of-the-art unsupervised approaches, i.e., SimCLR, MoCoV2, and PIC under ImageNet linear evaluation protocol and on several downstream visual tasks, verifying that full instance classification is a strong pretraining technique for many semantic visual tasks.

Introduction

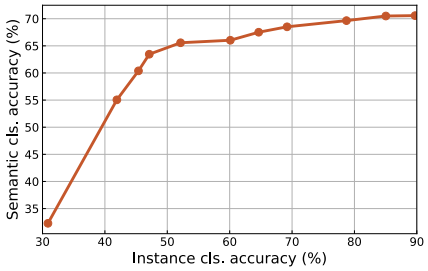
Unsupervised visual representation learning has recently shown encouraging progress (He et al. 2020; Chen et al. 2020a). Methods using *instance discrimination* as a pretext task (Tian, Krishnan, and Isola 2019; He et al. 2020; Chen et al. 2020a) have demonstrated competitive or even superior performance compared to supervised counterparts under ImageNet (Deng et al. 2009) linear evaluation protocol and on many downstream visual tasks. This shows the potential of unsupervised representation learning methods since they can utilize almost unlimited data without manual labels.

To solve the *instance discrimination* task, usually a dual-branch structure is used, where two transformed views of a same image are encouraged to get close, while transformed views from different images are expected to get far apart (He et al. 2020; Chen et al. 2020a,c). These methods often rely on specialized designs such as memory bank (Wu et al. 2018a), momentum encoder (He et al. 2020; Chen et al. 2020c), large batch size (Chen et al. 2020a,b), or shuffled batch normalization (BN) (He et al. 2020; Chen et al. 2020c) to compensate for the lack of negative samples or handle the information leakage issue (i.e., samples on a same GPU tend to get closer due to shared BN statistics).

Unlike dual-branch approaches, one-branch scheme (e.g., parametric instance-level classification) usually avoids the information leakage issue, and can potentially explore a larger set of negative samples. ExemplarCNN (Dosovitskiy et al. 2014) and PIC (Cao et al. 2020) are of this category.



(a) Outline of our representation learning framework.



(b) Correlation between instance and semantic classification.

Figure 1: (a) An overview of our unsupervised visual representation learning framework. Without manual labels, we simply train an instance-level classifier that tries to *distinguish all images in a dataset* to learn discriminative representations that can be well transferred to supervised tasks. (b) The relationship between instance (unsupervised) and semantic (supervised) classification accuracies. We observe a strong positive correlation between them in our experiments.

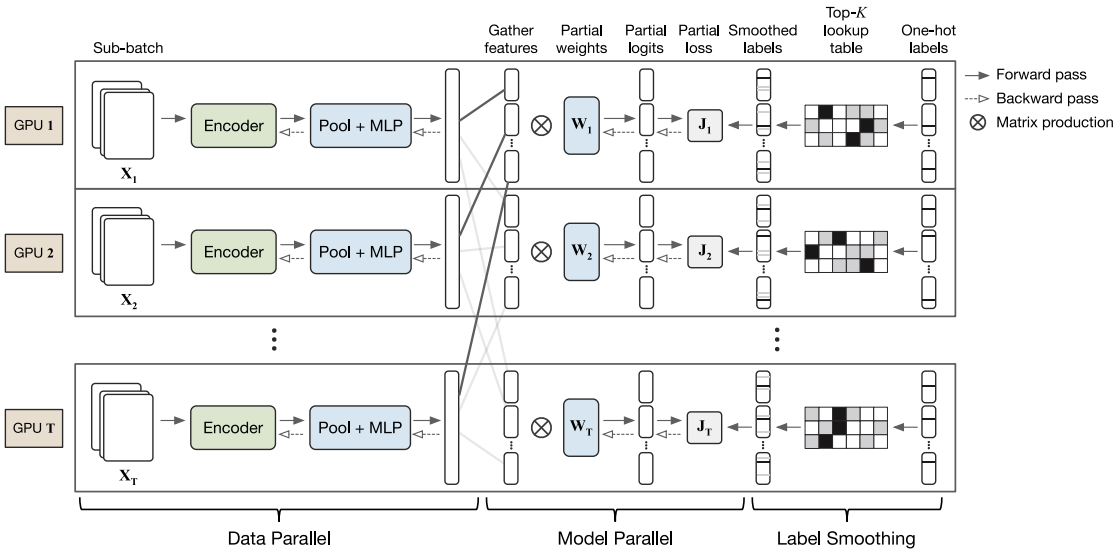


Figure 2: An outline of our distributed hybrid parallel (DHP) training process on T GPU nodes. *Data parallel:* Following *data parallel* mechanism, we copy the encoding and MLP layers to all nodes, each processing a subset of minibatch data. *Model parallel:* Following *model parallel* mechanism, we evenly divide the classification weights to different nodes, and distribute the computation of classification scores (forward pass) and weight/feature gradients (backward pass) to different GPUs. *Label smoothing:* We smooth labels of the top-K hardest negative classes for each instance to avoid optimizing over noisy pairs.

Nevertheless, due to the high GPU computation and memory overhead of large-scale instance-level classification, these methods are either tested on small datasets (Dosovitskiy et al. 2014) or rely on negative class sampling (Cao et al. 2020) to make training feasible.

This work summarizes typical challenges of using one-branch instance discrimination for unsupervised representation learning, including 1) the large-scale classification, 2) the slow convergence due to the infrequent instance access, and 3) a large number of negative classes that can be noisy, and proposes novel techniques to handle them. First, we introduce a hybrid parallel training framework to make large-scale classifier training feasible. It relies on model parallelism that divides classification weights to different GPUs, and evenly distribute the softmax computation (both in forward and backward passes) to them. Figure 2 shows an overview of our distributed training process. This training framework can theoretically support up to 100-million-way classification using 256 GPUs (Song et al. 2020), which far exceeds the number of 1.28 million ImageNet-1K instances, indicating the scalability of our method.

Second, instance classification faces the slow convergence problem due to the extremely infrequent visiting of instance samples (Cao et al. 2020). In this work, we tackle this problem by introducing *a contrastive prior* to the instance classifier. Specifically, with a randomly initialized network, we fix all but BN layers of it and run an inference epoch to extract all instance features; then we directly assign these features to classification weights as an initialization. The intuition is two-fold. On the one hand, we presume that *running BNs* may offer a contrastive prior in the output instance features, since in each iteration the features will subtract a weighted average of other instance features extracted in previous iterations. On the other hand, initializing classification weights as instance features in essence converts the classification task to a pair-wise instance comparison task, providing a warm start for convergence.

Finally, regarding the massive number of negative instance classes that significantly raises the risk of optimizing over very similar negative pairs, we propose to smooth the labels of the top-K hardest negative classes to make training easier. Specifically, we compute cosine similarities between instance proxies – their corresponding classification weights – and find the negative classes with the top-K highest similarities for each instance. The labels of these classes are smoothed by a factor of α (i.e., from $y_- = 0$ to $y_- = \alpha/K$). The right part of Figure 2 shows the smoothing process. Note these top-K indices are computed once per training epoch, which is very efficient and adds only minimal computational overhead for the training process.

We evaluate our method under ImageNet linear evaluation protocol and on several downstream tasks related to detection or fine-grained classification. Despite its simplicity, our method shows competitive results on these tasks. For example, our method achieves a top-1 accuracy of 71.4% under ImageNet linear evaluation protocol, outperforming all other instance discrimination based methods (Chen et al. 2020a,c; Cao et al. 2020). We also obtain a semi-supervised accuracy of 81.8% on ImageNet-1K when only 1% of labels are provided, surpassing SimCLR (Chen et al. 2020a) by around 4.7%. We hope our full instance classification framework can serve as a simple and strong baseline for the unsupervised representation learning community and beyond.

Related Work

Unsupervised Visual Representation Learning Unsupervised visual representation learning aims to learn discriminative representation from visual data where no manual labels are available. Usually a pretext task is utilized to determine the quality of the learned representation and to iteratively optimize the parameters. Representative pretext tasks include transformation prediction (Gidaris, Singh, and Komodakis 2018; Zhang et al. 2019), in-painting (Pathak et al. 2016), spatial or temporal patch order prediction (Doersch, Gupta, and Efros 2015; Noroozi et al. 2018), colorization (Zhang, Isola, and Efros 2016), clustering (Caron et al. 2018; Zhuang, Zhai, and Yamins 2019a; Asano, Rupprecht, and Vedaldi 2020), data generation (Jenni and Favaro 2018; Donahue and Simonyan 2019; Donahue, Krähenbühl, and Darrell 2016), geometry (Dosovitskiy et al. 2015), and a combination of multiple pretext tasks (Doersch and Zisserman 2017; Feng, Xu, and Tao 2019).

Contrastive Visual Representation Learning More recently, contrastive representation learning methods (Hénaff et al. 2019; He et al. 2020) have shown significant performance improvements by using strong data augmentation and proper loss functions (Chen et al. 2020c,a). For these methods, usually a dual-branch structure is employed, where two augmented views of an image are encouraged to get close while augmented views from different images are forced to get far apart. One problem of these methods is the shortage of negative samples. Some methods rely on large batch size (Chen et al. 2020a), memory bank (Wu et al. 2018a), or momentum encoder (He et al. 2020; Chen et al. 2020c) to enlarge the negative pool. Another issue is regarding the information leakage issue (He et al. 2020; Chen et al. 2020c) where features extracted on a same GPU tend to get close due to the shared BN statistics. MoCo (He et al. 2020; Chen et al. 2020c) solves this problem by using shuffled batch normalization (BN), while SimCLR (Chen et al. 2020a) handles the problem with a synchronized global BN.

Instance Discrimination for Representation Learning Unlike the two-branch structure used in contrastive methods, some approaches (Dosovitskiy et al. 2014; Cao et al. 2020) employ a parametric, one-branch structure for instance discrimination, which avoids the information leakage issue. Exemplar-CNN (Dosovitskiy et al. 2014) learns a classifier to discriminate between a set of "surrogate classes", each class represents different transformed patches of a single image. Nevertheless, it shows worse performance than non-parametric approaches (Wu et al. 2018a). PIC (Cao et al. 2020) improves Exemplar-CNN in two ways: 1) it introduces a sliding-window data scheduler to alleviate the infrequent instance visiting problem; 2) it utilizes recent classes sampling to reduce the GPU memory consumption. Despite its effectiveness, it relies on complicated scheduling and optimization processes, and it cannot fully explore the large number of negative instances. This work presents a much simpler instance discrimination method that uses an ordinary data scheduler and optimization process. In addition, it is able to make full usage of the massive number of negative instances in every training iteration.

Methodology

Overall Framework

This work presents an unsupervised representation learning method with a pretext task of classifying all image instances in a dataset. Figure 1 (a) shows the outline of our method. The pipeline is similar to common supervised classification, where *semantic classes* are replaced by *instance IDs*. Inspired by the design improvements used in recent unsupervised frameworks (Chen et al. 2020a), we slightly modify some components, including using stronger data augmentation (i.e., random crop, color jitter, and Gaussian blur), a two-layer MLP projection head, and a cosine softmax loss. The cosine softmax loss is defined as

$$J = -\frac{1}{|I|}\sum_{i\in I}\log\frac{\exp(\cos(\mathbf{w}_i,\mathbf{x}_i)/\tau)}{\sum_{j=1}^{N}\exp(\cos(\mathbf{w}_j,\mathbf{x}_i)/\tau)}, \tag{1}$$

where I denotes the indices of sampled image instances in a minibatch, $\mathbf{x}_i$ is the projected embedding of instance i, $\mathbf{W} = \{\mathbf{w}_1, \mathbf{w}_2, \cdots, \mathbf{w}_N\} \in \mathbb{R}^{D\times N}$ represents the instance classification weights, $\cos(\mathbf{w}_j, \mathbf{x}_i) = (\mathbf{w}_j^{\mathrm{T}}\mathbf{x}_i)/(\|\mathbf{w}_j\|_2 \cdot \|\mathbf{x}_i\|_2)$ denotes the cosine similarity between $\mathbf{w}_j$ and $\mathbf{x}_i$, and τ is a temperature adjusting the scale of cosine similarities.

Nevertheless, there are still challenges for this vanilla instance classification model to learn good representation, including 1) the large-scale instance classes (e.g., 1.28 million instance classes for ImageNet-1K dataset); 2) the extremely infrequent visiting of instance samples; and 3) the massive number of negative classes that makes training difficult. We propose three efficient techniques to improve the representation learning and the scalability of our method:

- *Hybrid parallelism.* To support large-scale instance classification, we rely on hybrid parallelism and evenly distribute the softmax computation (both in forward and backward passes) to different GPUs. Figure 2 shows a schematic of the distributed training process on T GPUs.
- *A contrastive prior.* To improve the convergence, we propose to introduce *a contrastive prior* to the instance classifier. This is simply achieved by initializing classification weights as raw instance features extracted by a *fixed* random network with *running BNs*.
- *Smoothing labels of hardest classes.* the massive number of negative classes raises the risk of optimizing over very similar pairs. We apply label smoothing on the top-K hardest instance classes to alleviate this issue.

Note that the above improvements only bring little or no computational overhead for the training process. Next we will introduce these techniques respectively in details.

Hybrid Parallelism

Training an instance classifier usually requires to learn a large-scale *fc* layer. For example, for ImageNet-1K with approximately 1.28 million images, one needs to optimize a weight matrix of size $\mathbf{W} \in \mathbb{R}^{D\times 1280000}$. For ImageNet-21K, the size further enlarged to $\mathbf{W} \in \mathbb{R}^{D\times 14200000}$. This is often infeasible when using a regular distributed data parallel (DDP) training pipeline. In this work, we introduce a

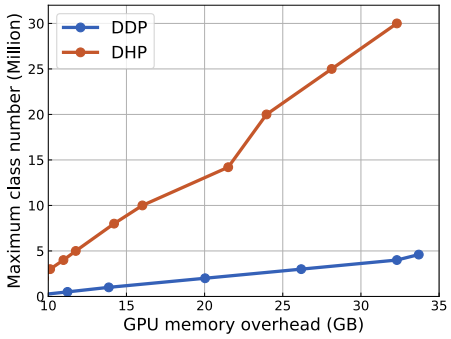


Figure 3: Comparison of the maximum number of classes supported by DDP and DHP training frameworks under different GPU memory constraints.

Dataset	#Instances	DDP		DHP	
		Memory	Time	Memory	Time
ImageNet-1K	1.28M	15.6GB	428s	**9.18GB**	**302s**
ImageNet-21K	14.2M	*OOM*[*]	-	**21.51GB**	**2940s**

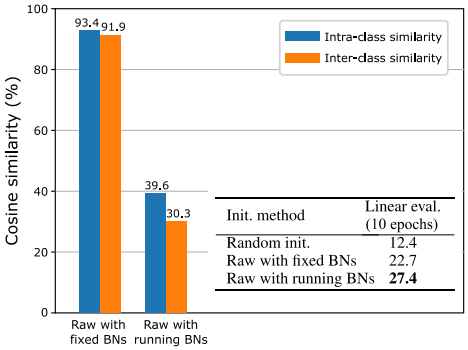
[*] *OOM*: Out-of-memory.

Table 1: Comparison of the GPU memory consumption and the training time per epoch of DDP and DHP training frameworks. Experiments are conducted on ImageNet-1K and ImageNet-21K datasets.

distributed hybrid parallel (DHP) training framework (Song et al. 2020) to make large-scale classification feasible.

Figure 2 summarizes the outline of the distributed hybrid parallel training process on T GPU nodes. For encoding and MLP layers, we follow the *data parallel* pipeline and **copy** them to different GPUs, each processing a subset of mini-batch data; while for the large-scale *fc* layer, we follow the *model parallel* mechanism and **split** the weights evenly to T GPUs. At each training iteration and for each GPU node, we 1) extract features of a subset of minibatch samples; 2) gather features from all other nodes; 3) compute partial cosine logits using local classification weights; 4) compute exponential values of logits and sum over all classes across GPUs to obtain the softmax denominators; 5) compute softmax probabilities and the cross-entropy loss on the subset data; 6) deduce gradients of the local loss with respect to features and weights; 7) gather feature gradients from all GPU node and sum them; 8) run a step of optimization to update parameters of encoding, MLP, and classification layers. The pipeline is repeated to loop through the complete dataset for several epochs to optimize for better representation.

Figure 3 compares the GPU memory overhead of DDP and DHP training frameworks when increasing the (pseudo) class number from 10K to 30M. The experiment is conducted on 64 V100 GPUs with 32GB memory and a total batch size of 4096. DDP reports out-of-memory (OOM) error when the class number reaches 4.7 million, while the DHP training framework can support up to 30 million number of classes, which is $\mathbf{6.4}\times$ of the DDP's limit. We also note that the DHP can benefit from more GPUs to support



Init. method	Linear eval. (10 epochs)
Random init.	12.4
Raw with fixed BNs	22.7
Raw with running BNs	**27.4**

Figure 4: *Table:* comparison of ImageNet linear evaluation accuracies of different weight-initialization methods, evaluated after 10 epochs training. *Bar chart:* comparison of average intra- and inter-instance-class cosine similarities when using different initialization methods.

larger-scale instance classification, but DDP does not bear this scalability. Table 1 compares the training efficiency of DDP and DHP frameworks on ImageNet-1K and ImageNet-21K under the same batch size settings. We show that the DHP training framework not only consumes less GPU memory, but also trains much faster than the DDP counterpart.

A Contrastive Prior

Instance classification faces the slow convergence problem in early epochs due to the infrequent visiting of instance samples (i.e., once access per epoch). A recent work (Cao et al. 2020) handles this infrequent visiting problem by using a sliding-window data scheduler, which samples overlapped batches between adjacent iterations. This increases the positive instance visiting but it also significantly multiplies the time of looping over the complete dataset.

In this work, we handle this problem from a different perspective: we propose to speed up the convergence by introducing a contrastive prior to classification weights. Specifically, before training started, we run an inference epoch using the *fixed* random initial network with *running BNs* to extract all instance features $\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_N\} \in \mathbb{R}^{D\times N}$; then we directly assign them to classification weights $\mathbf{W} = \{\mathbf{w}_1, \mathbf{w}_2, \cdots, \mathbf{w}_N\} \in \mathbb{R}^{D\times N}$ as an initialization. The intuition behind this initialization mechanism is two-fold. First, running BNs can offer a contrastive prior in the output features, since in each inference phase, the features computed after every BN layer will subtract a running average of other instance features extracted in previous iterations. Second, assigning features to weights approximately converts the classification task to a pair-wise metric learning task in early epochs, which is relatively easier to converge and offers a warm start for instance classification.

Figure 4 compares the discriminative ability of different classifier initialization schemes, i.e., random weight initialization (*random init.* for short), raw-feature initialization with fixed BNs (*raw with fixed BNs* for short), and raw-feature initialization with running BNs (*raw with running BNs* for short). We use ImageNet linear evaluation accu-

MLP head	A contrastive prior	Label smoothing	ImageNet Top-1	ImageNet Top-5
	✓	✓	58.6	83.1
✓			67.3	87.7
✓	✓		67.6	88.0
✓	✓	✓	**68.2**	**88.5**

Table 2: Ablation study on the effectiveness of different components in our method.

#Sampled instances	Top-1	Top-5
2^{10}	64.8	86.3
2^{12}	65.3	86.7
2^{14}	65.4	86.7
2^{16}	65.5	86.8
Full	**67.3**	**87.7**

Table 3: Ablation study that compares full instance classification and sampled instance classification.

racy (evaluated after 10 epochs training) as well as average intra- and inter-instance-class similarities as the indicators. As shown in Figure 4, *raw with running BNs* achieves the best linear evaluation accuracy, and clearly outperforms other initialization methods. In addition, *raw with running BNs* obtains a much larger similarity gap ($\sim 9.3\%$) between positive and negative instance pairs than *raw with fixed BNs* ($\sim 1.5\%$), validating the assumption that running BNs may provide a contrastive prior for instance discrimination.

We also note that the instance features extracted by a random network with running BNs are also a robust start for semantic classification. We run an instance retrieval experiment on the *train* set of ImageNet-1K with a randomly initialized ResNet-50 network to extract all image features, and determine whether the searched instance and the query instance are of a same semantic category. We find that a 3% top-1 accuracy can be achieved, which far exceeds the top-1 accuracy of 0.1% of a random guess.

Smoothing Labels of Hardest Classes

A challenge of instance-level classification is that it introduces a very large number of negative classes, significantly raising the risk of optimizing over very similar pairs that can be noisy and make the training hard to converge.

In this work, we handle this problem by applying label smoothing on a few hardest instance classes. Although other techniques (e.g., clustering) are also applicable, we choose label smoothing for its simplicity and efficiency. We notice that the semantically similar instance pairs are relatively stable across the training process. Therefore, we represent each instance i as its corresponding classification weights $\mathbf{w}_i$ (instead of its unstable features $\mathbf{x}_i$), and compute the cosine similarities between $\mathbf{w}_i$ and all other weights $\mathbf{W}_{\bar{i}} = \{\mathbf{w}_1, \cdots, \mathbf{w}_{i-1}, \mathbf{w}_{i+1}, \cdots, \mathbf{w}_N\} \in \mathbb{R}^{D\times(N-1)}$ to find the top-K hardest negative classes $H^i_- = \{c_1, c_2, \cdots, c_K\}$. The

#Epochs	Gaussian random	A contrastive prior
10	12.4	27.4 (**+15.0**)
25	40.8	46.3 (**+5.5**)
50	56.1	58.0 (**+1.9**)
100	62.9	64.1 (**+1.2**)
200	67.3	67.6 (**+0.3**)
400	69.3	69.7 (**+0.4**)

Table 4: Ablation study that compares Gaussian random and a contrastive prior for classifier initialization.

Hard class number K	Smoothing factor α	Top-1	Top-5
no smoothing		67.6	88.0
100	0.1	67.7	88.2
100	0.2	**68.2**	**88.5**
100	0.3	67.5	88.2
50	0.2	68.0	88.5
200	0.2	67.9	88.4

Table 5: Ablation study of label smoothing with different number of hard classes K and smoothing factor α.

label of class $j \in \{1, 2, \cdots, N\}$ is then defined as

$$y^i_j = \begin{cases} 1-\alpha, & j = i, \\ \alpha/K, & j \in H^i_-, \\ 0, & \text{otherwise.} \end{cases} \tag{2}$$

The loss function in Eq. (1) is redefined as

$$J = -\frac{1}{|I|}\sum_{i\in I}\log\frac{\sum_{j=1}^{N} y^i_j \exp(\cos(\mathbf{w}_j, \mathbf{x}_i)/\tau)}{\sum_{j=1}^{N}\exp(\cos(\mathbf{w}_j, \mathbf{x}_i)/\tau)}. \tag{3}$$

The top-K similarities between instance weights are computed once per epoch, which only amounts for a small fraction of training time. The smoothed softmax cross-entropy loss reduces the impact of noisy or very similar negative pairs on the learned representation. This is also verified later in our ablation study, where smoothing labels of several hardest classes improves the transfer performance.

Experiments

Experiment Settings

Training Datasets Unless specified, we use ImageNet-1K to train our unsupervised model for most experiments. ImageNet-1K consists of around 1.28 million images belonging to 1000 classes. We treat every image instance (along with its various transformed views) in the dataset as a unique class, and train a 1.28 million-way instance classifier as a pretext task to learn visual representation.

Evaluation Datasets The learned visual representations are evaluated in three ways: First, under the linear evaluation protocol of ImageNet-1K, we fix the representation model and learn a linear classifier upon it. The *top-1/top-5* classification accuracies are employed to compare different unsupervised methods. Second, we evaluate the semi-supervised

Method	Top-1	Top-5
Supervised	76.3	93.1
Exemplar (Dosovitskiy et al. 2014)	48.6	-
CPC (Oord, Li, and Vinyals 2018)	48.7	73.6
InstDisc. (Wu et al. 2018a)	54.0	-
B.BiGAN (Donahue and Simonyan 2019)	56.0	77.4
NPID++ (Wu et al. 2018b)	59.0	-
L.Agg. (Zhuang, Zhai, and Yamins 2019b)	60.2	-
MoCo (He et al. 2020)	60.6	-
SeLa (Asano, Rupprecht, and Vedaldi 2020)	61.5	84.0
PIRL (Misra and Maaten 2020)	63.6	-
CPCv2 (Hénaff et al. 2019)	63.8	85.3
CMC (Tian, Krishnan, and Isola 2019)	64.1	85.4
SimCLR (Chen et al. 2020a)	69.3	89.0
PIC (Cao et al. 2020)	70.8	90.0
MoCoV2 (Chen et al. 2020c)	71.1	-
Ours	**71.4**	**90.3**

Table 6: State-of-the-art comparison of linear classification accuracy of unsupervised methods on ImageNet-1K.

Method	Label Fraction	
	1%	10%
Supervised	48.4	80.4
Label propagation:		
PseudoLabels (Zhai et al. 2019)	51.6	82.4
VAT+Entropy Min. (Miyato et al. 2018)	47.0	83.4
UDA (Xie et al. 2019)	-	88.5
FixMatch (Sohn et al. 2020)	-	89.1
Representation Learning:		
InstDisc. (Dosovitskiy et al. 2014)	39.2	77.4
PIRL (Misra and Maaten 2020)	57.2	83.8
PCL (Li et al. 2020)	75.6	86.2
SimCLR (Chen et al. 2020a)	75.5	87.8
SimCLRv2 (Chen et al. 2020b)	**82.5**	**89.2**
PIC (Cao et al. 2020)	77.1	88.7
Ours	81.8	**89.2**

Table 7: State-of-the-art comparison of semi-supervised learning accuracy on ImageNet-1K.

learning performance on ImageNet-1K, where methods are required to classify images in the *val* set when only a small fraction (i.e., 1% or 10%) of manual labels are provided in the *train* set. Third, we evaluate the transferring performance by finetuning the representations on several downstream tasks and compute performance gains. In our experiments, downstream tasks include Pascal-VOC object detection (Everingham et al. 2010), iNaturalist18 fine-grained image classification (Van Horn et al. 2018), and many others.

Implementation Details We use ResNet-50 (He et al. 2016) as the backbone in all our experiments. We train our model using the SGD optimizer, where the weight decay and momentum are set to 0.0001 and 0.9, respectively. The initial learning rate (*lr*) is set to 0.48 and decays using the cosine annealing scheduler. In addition, we use 10 epochs of linear *lr* warmup to stabilize training. The minibatch size is 4096 and the feature dimension $D = 128$. We set the temperature in Eq. (1) as $\tau = 0.15$, and the smoothing factor in Eq. (3) as $\alpha = 0.2$. For fair comparison, following practices in recent works (Chen et al. 2020a; Cao et al. 2020), we feed two augmented views per instance for training. All experiments are conducted on 64 V100 GPUs with 32GB memory.

Ablation Study

This section validates several modeling and configuration options for our method. We compare the quality of representations using ImageNet linear protocol evaluated on the *val* set. In each experiment, the linear classifier is trained with a batch size of 2048 and a *lr* of 40 that decays during training under the cosine annealing rule.

Ablation: Effectiveness of Components Table 2 shows the linear evaluation results using different combinations of components in our method, including a two-layer MLP head, a contrastive prior, and label smoothing. Accuracies are measured after 200-epochs training. We find that all the three components bring performance gains, improving the top-1 accuracy of our method from 58.6% to a competitive 68.2%. We also observe that a vanilla instance classification model can already achieve a top-1 accuracy of 67.3%, suggesting that full instance classification is a strong baseline for unsupervised representation learning. The a contrastive prior and label smoothing on the top-K hardest classes further boost the linear evaluation accuracy by around 1%.

Ablation: Full Instance Classification *v.s.* Sampled Instance Classification Table 3 compares linear classification accuracies using representations learned by full instance classification and by sampled instance classification, with sampling sizes ranging from 2^{10} to 2^{16}. Note we remove *a contrastive prior* and *label smoothing* in the experiments and only analyze the impact of class sampling. We observe that full instance classification clearly outperforms sampled instance classification by a margin of 1.8%, verifying the benefits of exploring the complete set of negative instances.

Ablation: A Contrastive Prior *v.s.* Random Initialization Table 4 compares the linear evaluation performance of our method using Gaussian random and a contrastive prior for classifier initialization, with training length increased from 10 to 400 epochs. We observe that a contrastive prior significantly speeds up convergence compared to random initialization, especially in early epochs (i.e., epoch 10, 25, and 50). Besides, the accuracy of a contrastive prior version consistently outperforms random initialization counterpart, showing the robustness of our initialization mechanism.

Ablation: Label Smoothing on Hardest Classes Table 5 shows the impact of label smoothing on representation learning. We vary the considered number K of hardest negative classes, and the smoothing factor α. A *no smoothing* baseline where $K = 0$ and $\alpha = 0$ is also included for comparison. We observe that smoothing labels of a few hardest classes improves linear evaluation performance over non-smoothing baseline in most hyper-parameter settings. The best accuracy can be obtained with $K = 100$ and $\alpha = 0.2$, where a 0.6% gain of top-1 accuracy can be achieved.

Method	AP	AP50	AP75
Supervised	53.5	81.3	58.8
MoCo (He et al. 2020)	55.9	81.5	62.6
MoCoV2 (Chen et al. 2020c)	**57.4**	**82.5**	**64.0**
PIC (Cao et al. 2020)	57.1	82.4	63.4
Ours	57.2	82.2	**64.0**

Table 8: Comparison of transferring performance on PASCAL VOC object detection.

Method	Top-1	Top-5
Scratch	65.4	85.5
Supervised	66.0	85.6
MoCo (He et al. 2020)	65.7	85.7
PIC (Cao et al. 2020)	**66.2**	85.7
Ours	**66.2**	**86.2**

Table 9: Comparison of transferring performance on iNaturalist fine-grained classification.

Comparison with Previous Results

ImageNet Linear Evaluation Table 6 compares our work with previous unsupervised visual representation learning methods under the ImageNet linear evaluation protocol. We follow recent practices (Chen et al. 2020a,c) to train a longer length, i.e., 1000 epochs. The proposed unsupervised learning framework achieves a top-1 accuracy of 71.4% on ImageNet-1K, outperforming SimCLR (+2.1%), PIC (0.6%) and MoCoV2 (+0.3%). The results verify that a simple full instance classification framework can learn very competitive visual representations. The performance gains can partly be attributed to the ability of large-scale full negative instance exploration, which is not supported by previous unsupervised frameworks (Chen et al. 2020a,c; Cao et al. 2020).

Semi-supervised Learning Following (Kolesnikov, Zhai, and Beyer 2019), we sample a 1% or 10% fraction of labeled data from ImageNet, and train a classifier starting from our pretrained model to evaluate the semi-supervised learning performance. For 1% labels, we train the backbone with a *lr* of 0.001 and the classifier with a *lr* of 15. For 10% labels, the *lr*s for the backbone and the classifier are set to 0.001 and 10, respectively (Li et al. 2020). Table 7 compares our work with both *representation learning* based and *label propagation* based methods. We obtain performance comparable to SimCLRv2, which uses three steps training (pretrain, finetune, and distill) and momentum contrast, which are not used in our method. The results suggest the strong discriminative ability of our learned representation.

Transfer Learning To further evaluate the learned representation, we apply the pretrained model to several downstream visual tasks (including detection, fine-grained classification, and many others) to evaluate the transferring performance.

PASCAL VOC Object Detection: Following (He et al. 2020), we use Faster-RCNN (Ren et al. 2015) with ResNet-50 backbone as the object detector. We initialize ResNet-50 with our pretrained weights, and finetune all layers end-to-end on the *trainval07+12* set of the PASCAL VOC dataset (Everingham et al. 2010). We adopt the same experiment settings as MoCoV2 (Chen et al. 2020c). The AP (Average Precision), AP50, and AP75 scores on the *test2007* set are used as indicators. Table 8 shows the results. Our transferring performance is significantly better than supervised pretraining counterpart (+3.7% in AP), and is competitive with state-of-the-art unsupervised learning methods.

iNaturalist fine-grained classification: We finetune the pretrained model end-to-end on the *train* set of iNaturalist 2018 dataset (Van Horn et al. 2018) and evaluate the top-1 and top-5 classification accuracies on the *val* set. Results are shown in Table 9. Our method is closely competitive with the ImageNet supervised pretraining counterpart as well as previous state-of-the-art unsupervised methods. The results indicate the discriminative ability of our pretrained representation in fine-grained classification.

Method	CIFAR10	CIFAR100	SUN397	DTD
Scratch	95.9	80.2	53.6	64.8
Supervised	97.5	**86.4**	**64.3**	74.6
SimCLR	97.7	85.9	63.5	73.2
Ours	**97.8**	**86.2**	**64.2**	**77.6**

Table 10: Transferring performance of different pretrained models on more downstream visual tasks.

More downstream tasks: Table 10 shows transferring results on more downstream tasks, including image classification on CIFAR10, CIFAR100 (Krizhevsky and Hinton 2009), SUN397 (Xiao et al. 2010), and DTD (Cimpoi et al. 2014). To summarize, our method performs competitively with ImageNet supervised pretraining as well as state-of-the-art unsupervised pretraining.

Conclusion

In this work, we present an unsupervised visual representation learning framework where the pretext task is to distinguish all instances in a dataset with a parametric classifier. The task is similar to supervised semantic classification, but with a much larger number of classes (equal to the dataset size) and finer granularity. We first introduce a hybrid parallel training framework to make large-scale instance classification feasible, which significantly reduces the GPU memory overhead and speeds up training in our experiments. Second, we propose to improve the convergence by introducing a contrastive prior to the instance classifier. This is achieved by initializing the classification weights as raw instance features extracted by a fixed random network with running BNs. We show in our experiments that this simple strategy clearly speeds up convergence and improves the transferring performance. Finally, to reduce the impact of noisy negative instance pairs, we propose to smooth the labels of a few hardest classes. Extensive experiments on ImageNet classification, semi-supervised classification, and many downstream tasks show that our simple unsupervised representation learning method performs comparable or superior than state-of-the-art unsupervised methods.

References

Asano, Y. M.; Rupprecht, C.; and Vedaldi, A. 2020. Self-labelling via simultaneous clustering and representation learning. In *ICLR*.

Cao, Y.; Xie, Z.; Liu, B.; Lin, Y.; Zhang, Z.; and Hu, H. 2020. Parametric Instance Classification for Unsupervised Visual Feature Learning. *arXiv* .

Caron, M.; Bojanowski, P.; Joulin, A.; and Douze, M. 2018. Deep clustering for unsupervised learning of visual features. In *ECCV*, 132–149.

Chen, T.; Kornblith, S.; Norouzi, M.; and Hinton, G. 2020a. A Simple Framework for Contrastive Learning of Visual Representations. *arXiv* .

Chen, T.; Kornblith, S.; Swersky, K.; Norouzi, M.; and Hinton, G. 2020b. Big Self-Supervised Models are Strong Semi-Supervised Learners. *arXiv* .

Chen, X.; Fan, H.; Girshick, R.; and He, K. 2020c. Improved baselines with momentum contrastive learning. *arXiv* .

Cimpoi, M.; Maji, S.; Kokkinos, I.; Mohamed, S.; and Vedaldi, A. 2014. Describing Textures in the Wild. In *CVPR*.

Deng, J.; Dong, W.; Socher, R.; Li, L.-J.; Li, K.; and Fei-Fei, L. 2009. Imagenet: A large-scale hierarchical image database. In *CVPR*, 248–255. IEEE.

Doersch, C.; Gupta, A.; and Efros, A. A. 2015. Unsupervised visual representation learning by context prediction. In *ICCV*, 1422–1430.

Doersch, C.; and Zisserman, A. 2017. Multi-task self-supervised visual learning. In *ICCV*, 2051–2060.

Donahue, J.; Krähenbühl, P.; and Darrell, T. 2016. Adversarial feature learning. *arXiv* .

Donahue, J.; and Simonyan, K. 2019. Large scale adversarial representation learning. In *NIPS*, 10542–10552.

Dosovitskiy, A.; Fischer, P.; Springenberg, J. T.; Riedmiller, M.; and Brox, T. 2015. Discriminative unsupervised feature learning with exemplar convolutional neural networks. *TPAMI* 38(9): 1734–1747.

Dosovitskiy, A.; Springenberg, J. T.; Riedmiller, M.; and Brox, T. 2014. Discriminative unsupervised feature learning with convolutional neural networks. In *NIPS*, 766–774.

Everingham, M.; Van Gool, L.; Williams, C. K.; Winn, J.; and Zisserman, A. 2010. The pascal visual object classes (voc) challenge. *International journal of computer vision* 88(2): 303–338.

Feng, Z.; Xu, C.; and Tao, D. 2019. Self-supervised representation learning by rotation feature decoupling. In *CVPR*, 10364–10374.

Gidaris, S.; Singh, P.; and Komodakis, N. 2018. Unsupervised representation learning by predicting image rotations. *arXiv* .

He, K.; Fan, H.; Wu, Y.; Xie, S.; and Girshick, R. 2020. Momentum contrast for unsupervised visual representation learning. In *CVPR*, 9729–9738.

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2016. Deep residual learning for image recognition. In *CVPR*, 770–778.

Hénaff, O. J.; Srinivas, A.; De Fauw, J.; Razavi, A.; Doersch, C.; Eslami, S.; and Oord, A. v. d. 2019. Data-efficient image recognition with contrastive predictive coding. *arXiv* .

Jenni, S.; and Favaro, P. 2018. Self-supervised feature learning by learning to spot artifacts. In *CVPR*, 2733–2742.

Kolesnikov, A.; Zhai, X.; and Beyer, L. 2019. Revisiting self-supervised visual representation learning. In *CVPR*, 1920–1929.

Krizhevsky, A.; and Hinton, G. 2009. Learning Multiple Layers of Features from Tiny Images. *Master's thesis, University of Tront* .

Li, J.; Zhou, P.; Xiong, C.; Socher, R.; and Hoi, S. C. 2020. Prototypical Contrastive Learning of Unsupervised Representations. *arXiv* .

Misra, I.; and Maaten, L. v. d. 2020. Self-supervised learning of pretext-invariant representations. In *CVPR*, 6707–6717.

Miyato, T.; Maeda, S.-i.; Koyama, M.; and Ishii, S. 2018. Virtual adversarial training: a regularization method for supervised and semi-supervised learning. *TPAMI* 41(8): 1979–1993.

Noroozi, M.; Vinjimoor, A.; Favaro, P.; and Pirsiavash, H. 2018. Boosting self-supervised learning via knowledge transfer. In *CVPR*, 9359–9367.

Oord, A. v. d.; Li, Y.; and Vinyals, O. 2018. Representation learning with contrastive predictive coding. *arXiv* .

Pathak, D.; Krahenbuhl, P.; Donahue, J.; Darrell, T.; and Efros, A. A. 2016. Context encoders: Feature learning by inpainting. In *CVPR*, 2536–2544.

Ren, S.; He, K.; Girshick, R.; and Sun, J. 2015. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. In Cortes, C.; Lawrence, N. D.; Lee, D. D.; Sugiyama, M.; and Garnett, R., eds., *NIPS*, 91–99.

Sohn, K.; Berthelot, D.; Li, C.-L.; Zhang, Z.; Carlini, N.; Cubuk, E. D.; Kurakin, A.; Zhang, H.; and Raffel, C. 2020. Fixmatch: Simplifying semi-supervised learning with consistency and confidence. *arXiv* .

Song, L.; Pan, P.; Zhao, K.; Yang, H.; Chen, Y.; Zhang, Y.; Xu, Y.; and Jin, R. 2020. Large-Scale Training System for 100-Million Classification at Alibaba. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 2909–2930.

Tian, Y.; Krishnan, D.; and Isola, P. 2019. Contrastive Multiview Coding. *arXiv* .

Van Horn, G.; Mac Aodha, O.; Song, Y.; Cui, Y.; Sun, C.; Shepard, A.; Adam, H.; Perona, P.; and Belongie, S. 2018. The inaturalist species classification and detection dataset. In *CVPR*, 8769–8778.

Wu, Z.; Xiong, Y.; Yu, S. X.; and Lin, D. 2018a. Unsupervised feature learning via non-parametric instance discrimination. In *CVPR*, 3733–3742.

Wu, Z.; Xiong, Y.; Yu, S. X.; and Lin, D. 2018b. Unsupervised feature learning via non-parametric instance discrimination. In *CVPR*, 3733–3742.

Xiao, J.; Hays, J.; Ehinger, K. A.; Oliva, A.; and Torralba, A. 2010. Sun database: Large-scale scene recognition from abbey to zoo. In *CVPR*, 3485–3492. IEEE.

Xie, Q.; Dai, Z.; Hovy, E.; Luong, M.-T.; and Le, Q. V. 2019. Unsupervised data augmentation for consistency training. *arXiv* .

Zhai, X.; Oliver, A.; Kolesnikov, A.; and Beyer, L. 2019. S4l: Self-supervised semi-supervised learning. In *ICCV*, 1476–1485.

Zhang, L.; Qi, G.-J.; Wang, L.; and Luo, J. 2019. Aet vs. aed: Unsupervised representation learning by auto-encoding transformations rather than data. In *CVPR*, 2547–2555.

Zhang, R.; Isola, P.; and Efros, A. A. 2016. Colorful image colorization. In *ECCV*, 649–666. Springer.

Zhuang, C.; Zhai, A. L.; and Yamins, D. 2019a. Local aggregation for unsupervised learning of visual embeddings. In *ICCV*, 6002–6012.

Zhuang, C.; Zhai, A. L.; and Yamins, D. 2019b. Local aggregation for unsupervised learning of visual embeddings. In *ICCV*, 6002–6012.