Unsupervised Finetuning

8 authors, including:

Suichan Li
University of Science and Technology of China
11 PUBLICATIONS   **56** CITATIONS

Dongdong Chen
University of Science and Technology of China
117 PUBLICATIONS   **1,867** CITATIONS

Yinpeng Chen
Microsoft
86 PUBLICATIONS   **1,537** CITATIONS

Lei Zhang
University College Dublin
1,138 PUBLICATIONS   **37,721** CITATIONS

Some of the authors of this publication are also working on these related projects:

Project https://github.com/zsyOAOA/NIID-MSL View project

Project Mixed Reality Rehab View project

Unsupervised Finetuning

Suichan Li[1,*], Dongdong Chen[2,*,†], Yinpeng Chen[2], Lu Yuan[2], Lei Zhang[2], Qi Chu[1], Bin Liu[1], Nenghai Yu[1],
[1]University of Science and Technology of China   [2]Microsoft Cloud & AI
{lsc1230@mail., qchu@, flowice@, ynh@}ustc.edu.cn, cddlyf@gmail.com,
{ yiche, luyuan, leizhang}@microsoft.com

Abstract

This paper studies "unsupervised finetuning", the symmetrical problem of the well-known "supervised finetuning". Given a pretrained model and small-scale unlabeled target data, unsupervised finetuning is to adapt the representation pretrained from the source domain to the target domain so that better transfer performance can be obtained. This problem is more challenging than the supervised counterpart, as the low data density in the small-scale target data is not friendly for unsupervised learning, leading to the damage of the pretrained representation and poor representation in the target domain. In this paper, we find the source data is crucial when shifting the finetuning paradigm from supervise to unsupervise, and propose two simple and effective strategies to combine source and target data into unsupervised finetuning: "sparse source data replaying", and "data mixing". The motivation of the former strategy is to add a small portion of source data back to occupy their pretrained representation space and help push the target data to reside in a smaller compact space; and the motivation of the latter strategy is to increase the data density and help learn more compact representation. To demonstrate the effectiveness of our proposed "unsupervised finetuning" strategy, we conduct extensive experiments on multiple different target datasets, which show better transfer performance than the naive strategy.

1. Introduction

In recent years, visual recognition has achieved tremendous success [30, 19, 31] from the development of deep neural networks and large-scale labeled data. However, in some real applications, acquiring such large-scale data is so difficult or even possible, let alone the intensive annotation. Therefore, *finetuning from pretraining* [20, 4, 13] has become the dominant training paradigm for such application scenarios, *i.e.*, first pretrain the model on the large-scale

source data, then finetune the pretrained model with few labeled target data, which we call "supervised finetuning".

In this paper, we study *"unsupervised finetuning"*, which shares the similar goal as supervised finetuning that tries to adapt the representation pretrained from the source domain to the target domain, so that better downstream task performance can be achieved, *e.g.*, retrieval, unsupervised clustering, and few-shot transfer via supervised finetuning with few labeled samples. To the best of our knowledge, "unsupervised finetuning" is not well studied in the computer vision field and there is almost no prior work that studies generic unsupervised finetuning, even though unsupervised pretraining is actively studied as an extremely hot topic in recent two years.

At the first glance, implementing unsupervised finetuning is very intuitive, *i.e.*, continue unsupervised learning on the unlabeled target data from the pretrained representation, in a similar way as supervised finetuning. However, this naive way does not work that well, because existing unsupervised learning schemes (*e.g.*, contrastive learning) often require large-scale data to work properly [26]. Otherwise, they will destroy the original pretrained representation structure and struggle in learning a compact representation in the target domain. In the following section, we will analyze the mostly used unsupervised contrastive loss, and hypothesize that this may be pre-determined by its own formulation. Inspired by the analysis, we discover two simple and effective strategies: *sparse source data replaying* and *data mixing*.

Specifically, for data replaying, we add some source data back, and involve both these source data and the unlabeled target data into the contrastive learning process. The main motivation here is using such source data to occupy their pretrained representation space and help push the target data to reside in a smaller compact space. However, to ensure the efficiency of unsupervised finetuning, only a small portion (e.g., 10%) of source data is involved. To further increase the data density and help contrastive learning on a small data scale, data mixing is used to create extra positive samples. In details, given two random images, we create

a mixed image and regard both the original two images as positive samples weighted by the mixing weights. It can help pull similar images closer and learn more compact representation.

To demonstrate the effectiveness of our proposed "unsupervised finetuning" strategy, we conduct extensive experiments on multiple small-scale target datasets with both unsupervised pretrained models and supervised pretrained models. We also evaluate the unsupervised finetuned model on different transfer tasks to evaluate its transfer performance, including unsupervised image retrieval, unsupervised clustering, and few-shot transfer. Experimental results show that the proposed unsupervised finetuning strategy can achieve much better performance than the naive unsupervised finetuning strategy and the original pretrained representation.

To summarize, our contributions are in three aspects:

- To the best of our knowledge, we are the first that systematically analyzed the "unsupervised finetuning" problem by dissecting the widely used contrastive loss.

- Based on the analysis, we discover two simple and effective strategies, *i.e.*, *sparse source data replaying* and *data mixing*, based on which better unsupervised finetuning can be achieved.

- Extensive experiments are conducted and demonstrate the superiority and generalization ability of our unsupervised finetuning strategy across different target datasets, pretraining methods and transfer tasks.

2. Related Works

Supervised pretraining. Benefiting from the powerful learning capacity, deep neural networks revolutionized many computer vision tasks in recent years, such as image recognition [19, 30], object detection [27, 16, 3], and semantic segmentation [6, 23]. However, such deep networks often require a large-scale labeled dataset for training, *e.g.*, ImageNet[10] and JFT-300M [29]. In order to achieve great performance on small datasets, model pretraining [38, 37] has demonstrated to be an effective way, and most previous success comes from supervised pretraining. For example, the ImageNet pretrained models are widely used in a broad range of downstream tasks [27, 33, 14] and can significantly boost the downstream performance. In the recent work BiT [20], Alexander *et al.* demonstrate pretraining big models on big supervised data is extremely powerful and can achieve excellent transfer performance on downstream few-shot tasks.

Unsupervised Pretraining. Despite the great success of supervised pretraining, it is relatively difficult to scale up because annotating a large scale labeled data is extremely costly and time-consuming. Over the past few years, unsupervised or self-supervised pretraining has attracted more and more attentions. Early unsupervised pretraining methods often design some pretext tasks as the learning signal. Examples include recovering the input under corruptions, *e.g.*, denoise auto-encoders[32], cross-channel auto-encoders [36]. Other popular pre-tasks include rotation prediction [15] and patch ordering [11]. Recently, contrastive learning [34, 18, 7] has almost dominated this field. The underlying motivation is the InfoMax principle [24], which aims to maximize the mutual information across different augmentations of the same images. For detailed implementation, it encourages the representation of different augmentations of the same image to be similar and that of different images to be dissimilar. Many recent unsupervised pretraining methods [18, 17, 5] have shown unsupervised pretraining on ImageNet can achieve comparable transfer performance on downstream datasets to the supervised pretraining counterpart. But since unsupervised learning starts *tabula rasa*, it still requires extremely large amounts of unlabeled data to learn a good representation, which is also illustrated in the work [22, 26].

Nevertheless, in many application scenarios, it is difficult or even impossible to collect a large-scale dataset, such as finegrained classification and some other medical related tasks, especially when the semantic category number is small. Inspired by it, we study "unsupervised finetuning", which aims to relax this requirement and studies how to finetune the pretrained model with the small-scale target data in an unsupervised way. Our method is orthogonal to both supervised and unsupervised pretraining works. On the one hand, our unsupervised finetuning can work upon both supervised and unsupervised pretrained models. On the other hand, our unsupervised finetuning is essentially contrastive learning but with two new strategies, so integrating better contrastive learning methods can also improve our unsupervised finetuning method.

Target Domain Transfer. Given the pretrained model, the most widely used transfer strategy is supervised finetuning, i.e., collect some labeled samples in the target domain and finetune the pretrained model in a supervised way with a proper (often smaller) learning rate. In this paper, we study how to finetune the pretrained model with small-scale unlabeled target data, which is useful for many real application scenarios. One typical example is that, in some large-scale retrieval systems, there is only unlabeled target data. Another typical example is to provide better representation to conduct unsupervised clustering. But due to the lack of label supervision, unsupervised finetuning is not a trivial task like supervised finetuning. Using a small number of unlabeled target data, naively unsupervised learning from the pretrained model is not satisfactory. Because it will eas-

ily destroy the original representation structure and degrade to a trivial solution. In the experiment part, to demonstrate the better transfer performance from unsupervised finetuned representation, we will use several typical transfer strategies in the target domain for evaluation, including few-shot transfer, unsupervised retrieval and clustering.

3. Motivation

Given a small scale of unlabeled target data $T = \{x_1^t, ..., x_n^t\}$, there are two naive ways to learn representation from these data. One is directly conducting unsupervised learning on this data from scratch, and another is to conduct unsupervised learning on this data from the pretrained representation that is trained on other source domain. To implement the first solution, we adopt the existing state-of-the-art unsupervised learning method MoCoV2[9] on two example target datasets: Caltech101 [12] and Pet [25], which contain a total of 3680, 3060 images for training respectively. To evaluate the quality of the learned representation, we finetune the pretrained model with a few labeled samples to get the few-shot transfer performance. As the reference, we also show the few-shot transfer performance of the unsupervised pretrained model from the large-scale source dataset, *i.e.*, ImageNet.

As shown in the second and third row of Table 1, we can observe that conducting unsupervised learning directly on the small-scale target dataset cannot learn a good representation, and its few-shot transfer performance is much worse than the unsupervised pretrained model on the large-scale source dataset (ImageNet). But from another perspective, it demonstrates the value of the second baseline, *i.e.*, finetuning upon the pretrained representation from the source domain. However, how to conduct unsupervised finetuning is still unknown. As the simple baseline, we follow a similar strategy as the well studied "supervised finetuning". In other words, we continue unsupervised learning on the target data with a smaller learning rate by using the pretrained model as the initialization. The corresponding transfer performance is shown in the last row of Table 1. It can be seen that, on Caltech101, the transfer performance is slightly improved, but on Pet37, the transfer performance becomes even worse. It means unsupervised finetuning is not as trivial as simple finetuning.

To further understand why unsupervised finetuning is nontrivial, we follow the analysis in the work [8] about the contrastive loss, which represents the generalized contrastive loss in the below form:

$$\mathcal{L}_{\text{contr}} = \mathcal{L}_{\text{align}} + \lambda \mathcal{L}_{\text{distr}} \quad (1)$$

where $\mathcal{L}_{\text{align}}$ is the alignment term that encourages the learned representation of different augmented views of the same image to be as close as possible. And $\mathcal{L}_{\text{distr}}$ is the distribution term that encourages the learned representation to

Model	Pet37			Caltech101		
	2-Shot	4-Shot	8-Shot	2-Shot	4-Shot	8-Shot
Unsup-T	9.08	10.97	12.64	18.69	27.17	38.27
Unsup-S	56.84	68.94	75.10	63.12	76.54	84.45
Unsupft-T	55.94	67.03	72.92	68.48	78.18	82.03

Table 1: Few-shot transfer performance on two small-scale target datasets with different baseline models. Unsup-T and Unsup-S mean the unsupervised pretrained model on the small target dataset T and the large source dataset ImageNet S from scratch respectively. Unsup-T means the naive unsupervised finetuned model on T from the pretrained Unsup-S.

match a prior distribution of high entropy. Following the definition in [7], the standard contrastive loss is defined as:

$$\mathcal{L}_{contr} = -\frac{1}{N} \sum_{i,j \in \mathcal{MB}} \log \frac{\exp(\boldsymbol{z}_i \cdot \boldsymbol{z}_j/\tau)}{\sum_{k=1}^{2N} \mathbb{1}_{[k \neq i]} \exp(\boldsymbol{z}_i \cdot \boldsymbol{z}_k/\tau)} \quad (2)$$

where $\boldsymbol{z}_i, \boldsymbol{z}_j$ are the representations of the two augmented views of the same example. $\text{sim}(\boldsymbol{u}, \boldsymbol{v})$ is the cosine similarity between $\boldsymbol{u}$ and $\boldsymbol{v}$, and τ is the temperature hyperparameter. $\mathcal{MB}$ is the mini-batch that consists of random augmented pairs of images and N is the mini-batch size. In the detailed implementation, all the learned representation $\boldsymbol{z}_i$ will be normalized before passing into the contrastive loss. The above equation can be further rewritten into:

$$\mathcal{L}_{contr} = -\frac{1}{N\tau} \sum_{i,j} \text{sim}(\boldsymbol{z}_i, \boldsymbol{z}_j) + \\ \frac{1}{N} \sum_i \log \sum_{k=1}^{2N} \mathbb{1}_{[k \neq i]} \exp(\text{sim}(\boldsymbol{z}_i, \boldsymbol{z}_k)/\tau) \quad (3)$$

The first and second term correspond to the alignment term $\mathcal{L}_{\text{align}}$ and distribution term in Equation (1) respectively. More interestingly, the second term is related to pairwise potential in a Gaussian kernel and can be minimized with a uniform encoder. It will encourage the learned representation to be uniformly distributed on the hypersphere. Considering $\boldsymbol{z}_i$ often has a high dimension for larger representation capacity, if the training data has a limited scale, the data density is not enough to learn a compact representation because of the $\mathcal{L}_{distr}$ term. In this case, contrastive learning indeed degrades to a trivial solution, *i.e.*, the learned target representation scatters in the whole space sparsely. This explains why the first naive way does not work very well.

Conversely, if the training data has a very large scale and data density, different samples will squeeze each other, thus pushing the similar samples to be closer and learned representation more compact. Therefore, the ImageNet unsupervised pretrained model has a relatively good transfer performance. As for the naive simple unsupervised finetuning

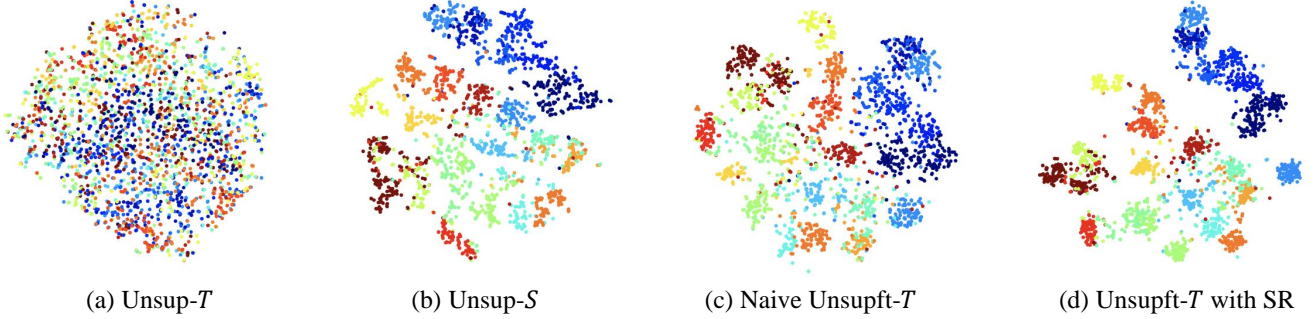| (a) Unsup-T | (b) Unsup-S | (c) Naive Unsupft-T | (d) Unsupft-T with SR |

Figure 1: The feature distribution visualization of the Pet dataset with different models via TSNE. (a) is the unsupervised pretrained model only on the small-scale Pet dataset from scratch, (b) is the unsupervised pretrained model on the large-scale ImageNet dataset, (c) is the naive unsupervised finetuned model on the Pet dataset by initializing with model (b), and (d) is the unsupervised finetuned model with our source data replaying.

baseline, since the source data is not involved in the finetuning stage, there is much freedom for the sparse target data to wander in the whole space. This will not only easily destroy the original space structure but also struggle in learning a more compact representation.

Trying to verify the above analysis, we further visualize the feature distribution on the Pet dataset with different pretrained models via TSNE in Figure 1. The visualization results roughly align with our above analysis and the fewshot transfer performance shown in the Table 1. In details, unsupervised learning directly on the small-scale target data almost learns a uniformly distributed representation, and the naive unsupervised finetuning strategy does not improve the pretrained representation a lot either. Accompanying the visualization, we also evaluate the feature clustering quality on the source ImageNet dataset and target Pet dataset. Before and after the unsupervised finetuning, the clustering accuracy on ImageNet and Pet are $36.85\%, 47.72\%$ and $16.60\%, 47.31\%$ respectively. The significant clustering accuracy drop on the ImageNet also indicates the naive unsupervised finetuning will destroy the pretrained representation structure on the source dataset.

4. Method

Based on the above analysis, we discover two simple and effective strategies: *sparse source data replaying* and *data mixing*. We find both of them can significantly improve unsupervised finetuning for better representation learning by adding more push force from the $\mathcal{L}_{distr}$ term and more pull force from the $\mathcal{L}_{align}$ term respectively. In this paper, we implement unsupervised finetuning on top of the state-of-the-art unsupervised learning framework MoCOV2 [9]. Before introducing the above two strategies, we will first briefly introduce MoCoV2 for better understanding.

In MoCoV2, for each image within the mini-batch, two different augmentations will be conducted. One augmented sample is regarded as query q and the other is regarded as the positive key k^+. All augmented versions of other images are regarded as negative keys $\{k_i^-\}$. Then for each image, the contrastive loss is defined as:

$$\mathcal{L}_q = -\log \frac{\exp(z_q \cdot z_k^+/\tau)}{\exp(z_q \cdot z_k^+/\tau) + \sum_{i=0}^{K-1} \exp(z_q \cdot z_{k_i}^-/\tau)},$$ (4)

where z_q, z_k are the L_2-normalized feature embedding by feeding q, k into two separated encoders: a query encoder and a key encoder. The query encoder and key encoder share the same architecture, *i.e.*, the backbone of the target network followed by an extra MLP-based projection head, but have different weights. The key idea of MoCo is making the weights of the key encoder be the momentum moving average of query encoder.

Sparse Source Data Replaying. To alleviate the sparse target data density issue and avoid destroying the original pretrained representation, we propose to add some source data back and mix them with the target data together for the unsupervised finetuning. The main motivation is from two aspects:

- These source data can help occupy their original feature space, which leaves a smaller remaining space for the target data to reside and avoids their original representation being destroyed.

- The training data density is increased, thus more push force from the source data will be imposed onto the target data via the distribution term L_{distr}.

However, since the source data has a very large scale, involving all of them into the unsupervised finetuning process will significantly hurt the efficiency. Fortunately, as

the source data is often well clustered in the pretraining representation space, we find that only randomly sampling a small portion (10% by default) of source data as the representative samples is good enough, especially when equipped with the following the data mixing strategy. Thus we call this strategy "sparse source data replaying". As shown in Figure 1 (d), with this strategy, a more compact target representation can be learned. The corresponding clustering accuracy on the ImageNet and Pet dataset also increases to $28.14\%, 56.11\%$ respectively, indicating better preserving of the source representation and more compact target representation.

Contrastive learning with Data Mixing To further increase the data density, we propose to use data mixing to create extra positive samples, which link different images with more alignment terms. In our method, we use cutmix[35] as the default instantiation of data mixing, and we empirically observed that other mixing strategies like MixUP can also work well. In details, given two images, we denote one augmented version of them as the querys q_a, q_b, and another augmented version as the positive keys k_a^+, k_b^+, then we create a mixed query sample q_m:

$$q_m \leftarrow \mathbf{M} \odot q_a + (\mathbf{1} - \mathbf{M}) \odot q_b \qquad (5)$$

when $\mathbf{M} \in [0,1]^{W \times H}$ is a binary mixing weight mask, $\mathbf{1}$ is a binary mask filled with ones. $\odot$ means the element-wise multiplication operation and W, H is the width and height of training images. Then the mixed query q_m is regarded as the positive sample both for k_a^+ and k_b^+, thus the contrastive loss with respect to q_m is changed to:

$$\mathcal{L}_{q_m} = -\Big(\lambda\log\frac{\exp(z_{q_m} \cdot z_{k_a}^+/\tau)}{\exp(z_{q_m} \cdot z_{k_a}^+/\tau) + \sum_{i=0}^{K-1}\exp(z_{q_m} \cdot z_{k_i}^-/\tau)}$$
$$+(1-\lambda)\log\frac{\exp(z_{q_m} \cdot z_{k_b}^+/\tau)}{\exp(z_{q_m} \cdot z_{k_b}^+/\tau) + \sum_{i=0}^{K-1}\exp(z_{q_m} \cdot z_{k_i}^-/\tau)}\Big) \qquad (6)$$

where λ is the combination ratio between two images and defined as the area ratio of q_a in $\mathbf{M}$. During training, λ is sampled from the beta distribution Beta(α, α), and $\alpha = 1$ by default. At each training iteration, we randomly selected two training samples from the mini-batch and generate the mixed query with a sampled λ.

To understand why data mixing can help, we can also rewrite the Equation (6) into the format of Equation (1) in a similar way. Then we will get two alignment terms bridged by the common query image q_m. If q_a and q_b are from the source and target domain respectively, the alignment terms will help pull their similar images closer and help encourage the learned representation more compact.

5. Experiments

In this section, we conduct extensive experiments and ablations to demonstrate the effectiveness of unsupervised finetuning in improving the transfer performance. We leverage ImageNet as the large-scale source data for pretraining and multiple public small datasets as target domain for unsupervised finetuning and transfer, including Caltech101 [12], Pet [25], CIFAR100 [21], and Food101 [2]. For these target datasets, we use their original data splits, which have a total of 3060, 3680, 50000, 75750 images for training respectively. Among them, Food101 (fine-grained food dataset) has a relatively large domain gap with ImageNet. To reduce the influence from randomness, each experiment is repeated three trials and reports the averaged result. For comparison, we follow the standard evaluation setting and report the mean class accuracy for Pet, Caltech101 and top-1 accuracy for other datasets. Regarding the pretrained model for unsupervised finetuning, we tried both supervised pretraining and unsupervised pretraining to demonstrate the generalization ability.

Evaluation Metrics. To evaluate the representation quality, we consider three different evaluation metrics: *few-shot transfer*, *retrieval accuracy*, and *clustering accuracy*. The first metric requires label samples while the latter two metrics are purely unsupervised metrics. More specifically, for few-shot transfer, we randomly choose K labeled samples for each category and finetune the learned representation with such labeled samples in a supervised way. For retrieval accuracy, we regard each image in the test set as the query, and retrieve the top-5 similar images from the training set by using the learned representation. Then the ratio of images that have the same label as the query image is defined as the retrieval accuracy. For clustering accuracy, we directly run K-Means clustering on the training set and adopt the BCubed Precision [1] as the clustering accuracy. Due to the space limit, we adopt the few-shot transfer performance as the main evaluation metric. In the ablation study , we will further try to apply our unsupervised finetuned representation to help semi-supervised learning.

Implementation details. For unsupervised pretraining, we use MocoV2[9] and follow its training protocol. The initial learning rate is 0.24 with a cosine scheduler and the batch size is 2,048. All the pretraining models are trained with 800 epochs. For supervised pretraining, we directly use the pretrained models from [20]. For unsupervised finetuning, the initial learning rate is 0.015 with batch size 256. All the models are finetuned with 200 epochs with a cosine learning rate scheduler. For transfer, we finetune the model for 30 epochs and the learning rate for the newly added *FC* layer and pretrained layers are 3.0 and 0.0001 respectively,

	Method	Caltech101	Pet	CIFAR100	Food101	Avg
1-Shot Acc	Unsup-S	45.49	41.23	13.65	10.09	27.62
	+ Naive Unsupft-T	55.65 (+10.16)	40.01 (−1.22)	20.83 (+7.18)	28.36 (+18.27)	36.21 (+8.59)
	+ Our Unsupft-T	74.33 (+28.84)	60.35 (+19.12)	33.75 (+20.10)	48.89 (+38.80)	54.33 (+26.71)
2-Shot Acc	Unsup-S	63.12	56.84	22.95	15.80	39.68
	+ Naive Unsupft-T	68.48 (+5.36)	55.94 (−0.9)	30.15 (+7.20)	39.64 (+23.84)	48.55 (+8.87)
	+ Our Unsupft-T	81.02 (+17.88)	70.29 (+13.45)	44.51 (+21.56)	61.99 (+46.19)	64.45 (+24.77)
4-Shot Acc	Unsup-S	76.54	68.94	35.08	25.49	51.51
	+ Naive Unsupft-T	78.18 (+1.64)	67.03 (−1.91)	39.70 (+4.62)	50.82 (+25.33)	58.93 (+7.42)
	+ Our Unsupft-T	85.35 (+8.81)	76.89 (+7.95)	52.60 (+17.52)	71.14 (+45.65)	71.50 (+19.99)
8-Shot Acc	Unsup-S	84.45	75.10	47.84	35.50	60.72
	+ Naive Unsupft-T	82.03 (−2.42)	72.92 (−2.18)	48.59 (+0.75)	59.58 (+24.08)	65.78 (+5.06)
	+ Our Unsupft-T	87.24 (+2.79)	79.88 (+4.78)	59.68 (+9.14)	76.86 (+41.36)	75.92 (+15.20)
RetrievalAcc	Unsup-S	83.14	69.70	53.25	35.91	60.50
	+ Naive Unsupft-T	84.50 (+1.36)	63.73 (−5.97)	64.27 (+11.02)	61.26 (+25.35)	68.44 (+7.94)
	+ Our Unsupft-T	85.95 (+2.81)	71.67 (+1.97)	68.83 (+15.58)	75.13 (+39.22)	75.40 (+14.90)
ClusterAcc	Unsup-S	56.16	47.72	23.91	11.85	34.91
	+ Naive Unsupft-T	61.33 (+5.17)	47.31 (−0.41)	41.72 (+17.81)	34.18 (+22.33)	46.14 (+11.23)
	+ Our Unsupft-T	69.93 (+13.77)	56.11 (+8.39)	44.92 (+21.01)	51.74 (+39.89)	55.68 (+20.77)

Table 2: Transfer performance comparison on different target datasets. "Unsup-S" denote the few-shot results by using ImageNet unsupervised pretrained model respectively, "Naive Unsupft-T, and "Our Unsup-T" denote the few-shot results by using the naive unsupervised finetuned model and our proposed unsupervised finetuned model from "Unsup-S". It shows our unsupervised finetuning strategy can learn better representation upon the pretrained models and outperforms the naive baseline strategy.

both for few-shot transfer and semi-supervised transfer. The default backbone is ResNet-50 [19].

5.1. Main Results

In our main experiments, we use the ImageNet unsupervised pretrained model as the initialization representation, and apply the naive unsupervised finetuning strategy and our proposed unsupervised finetuning strategy on top of it, respectively. To compare the representation quality, few-shot transfer accuracy, retrieval accuracy and unsupervised clustering accuracy are all evaluated. The detailed comparison results are shown in the Table 2. It can be seen that, by unsupervised finetuning the pretrained representation on the small-scale unlabeled target data, it can significantly improve the transfer performance in the target domain. And compared to the naive unsupervised finetuning strategy, our proposed unsupervised finetuning strategy consistently outperforms the original pretrained model and brings much larger performance gain, while the naive strategy achieves lower transfer performance than the pretrained model on the Pet dataset. By carefully studying the above results, we can draw more interesting conclusions:

- In general, the fewer the labeled samples, the bigger performance gain unsupervised finetuning will bring. This is because finetuning with fewer labeled samples is much easier to overfit, thus having a higher requirement of the initialization representation.

- If the target domain has more unlabeled data (*e.g.*, Food101), unsupervised finetuning will bring better performance and even the naive unsupervised finetuning strategy can get very plausible results.

- When the domain gap between the source dataset and target dataset is small, the performance gain from unsupervised finetuning is relatively smaller. For example, in the ImageNet dataset, there are a lot of animal images (especially "dog"), which potentially have a lot of overlapped categories with the Pet dataset. Therefore, on the Pet dataset, the pretrained representation already has very good transfer performance.

5.2. Ablation Study

In this section, we conduct a series of ablation studies to demonstrate the generalization ability of unsupervised finetuning and the importance of our discovered strategies.

Generalization Ability to Supervised Pretrained Models. Besides applying to the default unsupervised pretrained models, our proposed unsupervised finetuning strategy is also general to supervised pretrained models. In this experiment, we use the supervised pretrained ResNet50 model from BiT [20] as the instantiation and apply unsupervised finetuning in a similar way. The few-shot transfer performance before and after unsupervised finetuning

K	Method	Caltech101	Pet	CIFAR100	Food101	Avg
1-Shot Acc	Sup-S	45.78	50.04	25.73	16.48	38.39
	+ Our Unsupft-T	73.35 $_{(+27.57)}$	61.30 $_{(+11.26)}$	32.29 $_{(+6.56)}$	42.60 $_{(+26.12)}$	56.48 $_{(+18.09)}$
2-Shot Acc	Sup-S	61.07	63.95	37.40	24.99	51.14
	+ Our Unsupft-T	79.12 $_{(+18.05)}$	73.59 $_{(+9.64)}$	45.93 $_{(+8.53)}$	61.70 $_{(+36.71)}$	68.38 $_{(+17.24)}$
4-Shot Acc	Sup-S	73.89	76.08	47.79	35.26	62.41
	+ Our Unsupft-T	84.42 $_{(+10.53)}$	84.41 $_{(+8.33)}$	55.09 $_{(+7.30)}$	67.33 $_{(+32.07)}$	75.89 $_{(+13.48)}$
8-Shot Acc	Sup-S	83.87	82.87	57.46	45.60	70.90
	+ Our Unsupft-T	88.08 $_{(+4.21)}$	85.61 $_{(+3.74)}$	60.54 $_{(+3.08)}$	69.54 $_{(+23.94)}$	79.12 $_{(+8.22)}$

Table 3: Few-shot transfer performance on different target datasets when applying unsupervised finetuning upon the supervised pretrained models, which demonstrates the generalization ability of unsupervised finetuning. Here, "Sup-S" denote the few-shot results by using the original ImageNet supervised pretrained model, "Our Unsup-T" denote the few-shot results by our proposed unsupervised finetuned model from "Sup-S".

K	Method	Caltech101	Pet	CIFAR100	Food101	Avg
1-Shot Acc	Sup-S	53.28	50.67	26.76	17.05	36.94
	+ Our Unsupft-T	74.03 $_{(+20.75)}$	64.02 $_{(+13.35)}$	34.58 $_{(+7.82)}$	42.99 $_{(+25.94)}$	53.91 $_{(+16.97)}$
2-Shot Acc	Sup-S	66.32	65.96	37.20	26.20	48.92
	+ Our Unsupft-T	83.59 $_{(+17.27)}$	79.37 $_{(+13.41)}$	50.20 $_{(+13.00)}$	64.33 $_{(+38.13)}$	69.37 $_{(+20.45)}$
4-Shot Acc	Sup-S	78.08	79.50	49.20	36.38	60.79
	+ Our Unsupft-T	87.63 $_{(+9.55)}$	86.15 $_{(+6.65)}$	58.32 $_{(+9.12)}$	70.02 $_{(+33.64)}$	75.53 $_{(+14.74)}$
8-Shot Acc	Sup-S	86.23	82.69	59.36	45.84	68.53
	+ Our Unsupft-T	89.65 $_{(+3.42)}$	86.32 $_{(+3.63)}$	63.97 $_{(+4.61)}$	71.95 $_{(+26.11)}$	77.97 $_{(+9.44)}$

Table 4: Few-shot transfer performance on the large supervised pretrained model ResNet101. The annotations are the same as Table 3, and the only difference is the larger pretrained model. It demonstrates that our unsupervised finetuning is also general to large pretrained models and brings better transfer performance.

is shown in the Table 3. It can be seen that, our unsupervised finetuning can also boost the transfer performance of supervised pretrained models. Compared to the few-shot transfer performance shown in the Table 2, we can further observe: 1) the original supervised pretrained model has better few-shot transfer performance than the original unsupervised pretrained model. This is because supervised pretrained models are often trained with the cross entropy loss by using supervised labels, and there is not a similar distribution term in the cross entropy loss to encourage uniformity. Therefore, supervised pretrained representation is more compact; 2) by equipping with our unsupervised finetuning, the performance gap between unsupervised and supervised pretrained model becomes much smaller. Note that, we also try the naive unsupervised finetuning strategy upon the supervised pretrained model, but find its transfer performance is very bad, so we do not provide its results in the table.

Generalization Ability to Large Models. Besides the pretraining method, our unsupevised finetuning strategy also generalizes well to large models. In the Table 4, we take the supervised pretrained ResNet101 model as an example and evaluate its transfer performance. It shows that,

even though the large model itself can produce better transfer performance than the default ResNet50 model, it can still improve its performance in a similar way to the small model by using the unsupervised finenetuned model as initialization.

Application in Semi-supervised Learning. The essence of unsupervised finetuning is utilizing the unlabeled target data to get a better representation. Similarly, in semi-supervised learning, it also aims to leverage the unlabeled target data as the extra learning guidance to achieve better performance. In this experiment, we are curious about whether semi-supervised learning can still benefit from unsupervised finetuned representation under the same data setting. To answer this question, we leverage the state-of-the-art semi-supervised learning method FixMatch [28], and use the original unsupervised pretrained representation and unsupervised finetuned representation as the initialization respectively.

Surprisingly, as shown in the Table 5, semi-supervised learning also benefits from better initialization models from unsupervised finetuning, but the overall performance gain is smaller. In our understanding, many semi-supervised methods including FixMatch use the pseudo labels of the un-

K	Method	Caltech101	Pet	CIFAR100	Food101	Avg
Semi-2 Acc	Unsup-S	79.04	54.52	41.43	18.23	48.31
	+ Our Unsupft-T	82.84 (+3.80)	68.58 (+14.06)	56.35 (+14.92)	70.33 (+52.10)	69.53 (+21.22)
Semi-4 Acc	Unsup-S	86.07	68.18	61.54	34.67	62.61
	+ Our Unsupft-T	86.87 (+0.8)	73.25 (+5.07)	66.18 (+4.64)	79.35 (+44.68)	76.41 (+13.80)
Semi-8 Acc	Unsup-S	90.12	75.39	70.59	59.06	73.79
	+ Our Unsupft-T	90.27 (+0.15)	79.18 (+3.79)	71.65 (+1.06)	82.35 (+23.29)	80.86 (+7.07)

Table 5: Semi-supervised learning results by using different models as initialization. "Semi-K" means K labeled samples together with the whole unlabeled data are used for semi-supervised learning. Other annotations are similar as Table 2. It shows that, even though semi-supervised learning also leverage the unlabeled data, our unsupervised finetuned model can provide better initialization for it and boost the final performance.

Dataset	Unsupft	SDR	Mix	$K = 2$	$K = 4$	$K = 8$
Caltech101	–			63.12	76.54	84.45
	✓			68.48	78.18	82.03
	✓	✓		75.28	82.99	85.15
	✓	✓	✓	81.02	85.35	87.24
Pet	–			56.84	68.94	75.10
	✓			55.94	67.03	72.92
	✓	✓		62.81	71.35	75.14
	✓	✓	✓	70.29	76.89	79.88
CIFAR100	–			22.95	35.08	47.84
	✓			30.15	39.70	48.59
	✓	✓		35.60	45.01	52.62
	✓	✓	✓	44.51	52.60	59.68
Food101	–			15.84	25.49	35.50
	✓			39.64	50.82	59.58
	✓	✓		38.87	48.97	56.85
	✓	✓	✓	61.99	71.14	76.86

Table 6: Ablation study of the two key strategies: sparse source data replaying (SDR) and data mixing (Mix). "–" means baseline without unsupervised finetuning, "✓" means having the corresponding strategy. The second row of each dataset only with "Unsupft" checked denotes the naived unsupervised finetuning. It shows that both source data replaying and data mixing can help learn better representations.

labeled data as the extra learning constraints, so providing better initialization will help generate more precise pseudo labels, subsequently improving the final performance. Conversely, if the initialization representation is not good, the pseudo labels will contain more errors and the gain from the unlabeled data will be very marginal. For example, compared to the few-shot transfer, when only 2 labeled samples exist for each category on the Food101 dataset, semi-supervised learning initialized from the ImageNet unsupervised pretrained model can only bring about 2% performance gain. By contrast, by using our unsupervised finetuned model as initialization, the extra performance gain from semi-supervised learning is 8.3%.

Importance of Source Data Replaying and Data Mixing.
Sparse source data replaying and data mixing are the two keys in our unsupervised finetuning strategy. To quantitatively demonstrate their importance, we conduct the ablation studies by gradually removing each of them. As shown in the Table 6, on all the datasets except Food101, source data replaying can boost the transfer performance by a large margin. For example, on the Caltech101 and Pet dataset, using source data replaying will boost the 2-shot transfer performance by 6.8%, 6.9% respectively. The reason why source data replaying cannot help Food101 should be because Food101 already has a larger number of images for each category, *i.e.*, 750 images per category. As for data mixing, it consistently brings better performance on all the datasets. Especially on the Food101 dataset, the 2-shot transfer performance is significantly boosted by 23.12%.

6. Conclusion

Unsupervised learning has achieved great progress in recent years and become one of the most active and hot research topic. However, almost all existing works focus on the large-scale unsupervised pretraining, *i.e.*, conduct unsupervised learning on a large-scale dataset. But in many real application scenarios, collecting such a large-scale unlabeled dataset is difficult or impossible. In this paper, we study "unsupervised finetuning", which aims to relax this requirement by unsupervised learning on a small-scale unlabel dataset on top of the pretrained representation. To the best of our knowledge, there is no prior work that focuses on this problem in the computer vision field yet. According to our analysis, we find unsupervised learning on the small-scale unlabeled data either from scratch or from the pretrained representation does not work well. After analyzing the widely used contrastive loss, we propose two simple and effective strategies, "sparse source data replaying" and "data mixing". Through extensive experiments, we demonstrate the proposed unsupervised finetuning strategy can achieve better performance than the naive unsupervised finetuning strategy. We hope our study will inspire more research in this direction.

References

[1] Enrique Amigó, Julio Gonzalo, Javier Artiles, and Felisa Verdejo. A comparison of extrinsic clustering evaluation metrics based on formal constraints. *Information retrieval*, 12(4):461–486, 2009. 5

[2] Lukas Bossard, Matthieu Guillaumin, and Luc Van Gool. Food-101–mining discriminative components with random forests. In *European conference on computer vision*, pages 446–461. Springer, 2014. 5

[3] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *European Conference on Computer Vision*, pages 213–229. Springer, 2020. 2

[4] Mathilde Caron, Piotr Bojanowski, Julien Mairal, and Armand Joulin. Unsupervised pre-training of image features on non-curated data. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2959–2968, 2019. 1

[5] Mathilde Caron, Ishan Misra, Julien Mairal, Priya Goyal, Piotr Bojanowski, and Armand Joulin. Unsupervised learning of visual features by contrasting cluster assignments. 2020. 2

[6] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L Yuille. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *IEEE transactions on pattern analysis and machine intelligence*, 40(4):834–848, 2017. 2

[7] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, pages 1597–1607. PMLR, 2020. 2, 3

[8] Ting Chen and Lala Li. Intriguing properties of contrastive losses. *arXiv preprint arXiv:2011.02803*, 2020. 3

[9] Xinlei Chen, Haoqi Fan, Ross Girshick, and Kaiming He. Improved baselines with momentum contrastive learning. *arXiv preprint arXiv:2003.04297*, 2020. 3, 4, 5

[10] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009. 2

[11] 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. 2

[12] Li Fei-Fei, Rob Fergus, and Pietro Perona. Learning generative visual models from few training examples: An incremental bayesian approach tested on 101 object categories. In *2004 conference on computer vision and pattern recognition workshop*, pages 178–178. IEEE, 2004. 3, 5

[13] Dengpan Fu, Dongdong Chen, Jianmin Bao, Hao Yang, Lu Yuan, Lei Zhang, Houqiang Li, and Dong Chen. Unsupervised pre-training for person re-identification. *arXiv preprint arXiv:2012.03753*, 2020. 1

[14] Dengpan Fu, Bo Xin, Jingdong Wang, Dongdong Chen, Jianmin Bao, Gang Hua, and Houqiang Li. Improving person re-identification with iterative impression aggregation. *IEEE Transactions on Image Processing*, 2020. 2

[15] Spyros Gidaris, Praveer Singh, and Nikos Komodakis. Unsupervised representation learning by predicting image rotations. *arXiv preprint arXiv:1803.07728*, 2018. 2

[16] Ross Girshick. Fast r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 1440–1448, 2015. 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

[18] 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. 2

[19] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 1, 2, 6

[20] Alexander Kolesnikov, Lucas Beyer, Xiaohua Zhai, Joan Puigcerver, Jessica Yung, Sylvain Gelly, and Neil Houlsby. Big transfer (bit): General visual representation learning. *arXiv preprint arXiv:1912.11370*, 6(2):8, 2019. 1, 2, 5, 6

[21] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009. 5

[22] Suichan Li, Dongdong Chen, Yinpeng Chen, Lu Yuan, Lei Zhang, Qi Chu, Bin Liu, and Nenghai Yu. Improve unsupervised pretraining for few-label transfer. In *ICCV*, 2021. 2

[23] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3431–3440, 2015. 2

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

[25] Omkar M Parkhi, Andrea Vedaldi, Andrew Zisserman, and CV Jawahar. Cats and dogs. In *2012 IEEE conference on computer vision and pattern recognition*, pages 3498–3505. IEEE, 2012. 3, 5

[26] Cheng Perng Phoo and Bharath Hariharan. Self-training for few-shot transfer across extreme task differences. *arXiv preprint arXiv:2010.07734*, 2020. 1, 2

[27] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. *arXiv preprint arXiv:1506.01497*, 2015. 2

[28] Kihyuk Sohn, David Berthelot, Chun-Liang Li, Zizhao Zhang, Nicholas Carlini, Ekin D Cubuk, Alex Kurakin, Han Zhang, and Colin Raffel. Fixmatch: Simplifying semi-supervised learning with consistency and confidence. *arXiv preprint arXiv:2001.07685*, 2020. 7

[29] Chen Sun, Abhinav Shrivastava, Saurabh Singh, and Abhinav Gupta. Revisiting unreasonable effectiveness of data in deep learning era. In *Proceedings of the IEEE international conference on computer vision*, pages 843–852, 2017. 2

[30] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1–9, 2015. 1, 2

[31] Mingxing Tan and Quoc Le. Efficientnet: Rethinking model scaling for convolutional neural networks. In *International Conference on Machine Learning*, pages 6105–6114. PMLR, 2019. 1

[32] Pascal Vincent, Hugo Larochelle, Yoshua Bengio, and Pierre-Antoine Manzagol. Extracting and composing robust features with denoising autoencoders. In *Proceedings of the 25th international conference on Machine learning*, pages 1096–1103, 2008. 2

[33] Guanshuo Wang, Yufeng Yuan, Xiong Chen, Jiwei Li, and Xi Zhou. Learning discriminative features with multiple granularities for person re-identification. In *Proceedings of the 26th ACM international conference on Multimedia*, pages 274–282, 2018. 2

[34] 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. 2

[35] Sangdoo Yun, Dongyoon Han, Seong Joon Oh, Sanghyuk Chun, Junsuk Choe, and Youngjoon Yoo. Cutmix: Regularization strategy to train strong classifiers with localizable features. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6023–6032, 2019. 5

[36] Richard Zhang, Phillip Isola, and Alexei A Efros. Split-brain autoencoders: Unsupervised learning by cross-channel prediction. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1058–1067, 2017. 2

[37] Luowei Zhou, Hamid Palangi, Lei Zhang, Houdong Hu, Jason Corso, and Jianfeng Gao. Unified vision-language pretraining for image captioning and vqa. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 13041–13049, 2020. 2

[38] Barret Zoph, Golnaz Ghiasi, Tsung-Yi Lin, Yin Cui, Hanxiao Liu, Ekin D Cubuk, and Quoc V Le. Rethinking pre-training and self-training. *arXiv preprint arXiv:2006.06882*, 2020. 2