Bi-Tuning:
Efficient Transfer from Pre-Trained Models

Jincheng Zhong, Haoyu Ma, Ximei Wang, Zhi Kou, and Mingsheng Long✉

School of Software, BNRist, Tsinghua University, China
{zjc22,mhy22,kz19}@mails.tsinghua.edu.cn
messixmwang@tencent.com   mingsheng@tsinghua.edu.cn

Abstract. It is a *de facto* practice in the deep learning community to first pre-train a deep neural network from a large-scale dataset and then fine-tune the pre-trained model to a specific downstream task. Recently, both supervised and unsupervised pre-training approaches to learning representations have achieved remarkable advances, which exploit the discriminative knowledge of labels and the intrinsic structure of data, respectively. It follows the natural intuition that both the discriminative knowledge and the intrinsic structure of the downstream task can be useful for fine-tuning. However, existing fine-tuning methods mainly leverage the former and discard the latter. A natural question arises: How to fully explore the intrinsic structure of data for boosting fine-tuning? In this paper, we propose *Bi-tuning*, a general learning approach that is capable of fine-tuning both supervised and unsupervised pre-trained representations to downstream tasks. Bi-tuning generalizes the vanilla fine-tuning by integrating two heads upon the backbone of pre-trained representations: a classifier head with an improved contrastive cross-entropy loss to better leverage the label information in an instance-contrast way, and a projector head with a newly-designed categorical contrastive learning loss to fully exploit the intrinsic structure of data in a category-consistent way. Comprehensive experiments confirm that Bi-tuning achieves state-of-the-art results for fine-tuning tasks of both supervised and unsupervised pre-trained models by large margins.

Keywords: Transfer Learning · Fine-Tuning · Contrastive Learning.

1 Introduction

In the last decade, remarkable advances in deep learning have been witnessed in diverse applications across many fields, such as computer vision, robotic control, and natural language processing in the presence of large-scale labeled datasets. However, in many practical scenarios, we may have only access to a small labeled dataset, making it impossible to train deep neural networks from scratch. Therefore, it has become increasingly common within the deep learning community to first *pre-train* a deep neural network from a large-scale dataset and then *fine-tune* the pre-trained model to a specific downstream task. Fine-tuning requires fewer labeled data, enables faster training, and usually achieves better performance than training from scratch [17]. This two-stage style of pre-training and fine-tuning lays as the transfer learning foundation of various deep learning applications.

In the *pre-training* stage, there are mainly two approaches to pre-train a deep model: supervised pre-training and unsupervised pre-training. Recent years have witnessed the success of numerous supervised pre-trained models, e.g. ResNet [18] and EfficientNet [36], by exploiting the discriminative knowledge of manually-annotated labels on a large-scale dataset like ImageNet [6]. Meanwhile, unsupervised representation learning is recently changing the field of natural language processing by models pre-trained with a large-scale corpus, e.g. BERT [7] and GPT [33]. In computer vision, remarkable advances in unsupervised representation learning [40,16,3], which exploit the intrinsic structure of data by contrastive learning [15], have also changed the field dominated chronically by supervised pre-trained representations.

In the *fine-tuning* stage, transferring a model from supervised pre-trained models has been empirically studied in [21]. During the past years, several sophisticated fine-tuning methods were proposed, including L2-SP [24], DELTA [23] and BSS [5]. These methods focus on leveraging the discriminative knowledge of labels from the downstream task by a cross-entropy loss and the implicit bias of pre-trained models by a regularization term. However, the intrinsic structure of data in the downstream task is generally discarded during fine-tuning. Further, rare attention has been paid to fine-tuning efficiently from an unsupervised pre-trained model. In a prior study, we empirically observed that unsupervised pre-trained representations focus more on the intrinsic structure, while supervised pre-trained representations explain better on the label information, as shown in Figure 3. This implies that fine-tuning unsupervised pre-trained representations [16] would be more difficult and deserves further investigation.

Regarding to the success of supervised and unsupervised pre-training approaches, it follows a natural intuition that both *discriminative knowledge* and *intrinsic structure* of the downstream task can be useful for fine-tuning. A question arises: How to fully explore the intrinsic structure of data for boosting fine-tuning? To tackle this major challenge of deep learning, we propose **Bi-tuning**, a general learning approach that is capable of fine-tuning both supervised and unsupervised pre-trained representations to downstream tasks. Bi-tuning generalizes the vanilla fine-tuning by integrating two specific heads upon the backbone of pre-trained representations:

- A classifier head with an improved contrastive cross-entropy loss to better leverage the label information in an instance-contrast way, which is the dual view of the vanilla cross-entropy loss and is expected to achieve a more compact intra-class structure.
- A projector head with a newly-designed categorical contrastive learning loss to fully exploit the intrinsic structure of data in a category-consistent way, resulting in a more harmonious cooperation between the supervised and unsupervised fine-tuning mechanisms.

Designed as a general-purpose fine-tuning approach, Bi-tuning can be applied with a variety of backbones without any additional assumptions. Comprehensive experiments confirm that Bi-tuning achieves state-of-the-art results for fine-tuning tasks of both supervised and unsupervised pre-trained models by large margins. We justify through ablations and analyses the effectiveness of the proposed two-heads fine-tuning architecture with their novel loss functions. Code is available at `https://github.com/thuml/Transfer-Learning-Library`.

2 Related Work

2.1 Pre-training

During the past years, supervised pre-trained models achieve impressive advances by exploiting the inductive bias of label information on a large-scale dataset like ImageNet [6], such as GoogleNet [35], ResNet [18], DenseNet [19], EfficientNet [36] and ViT [10], to name a few. Meanwhile, unsupervised representation learning is recently shining in the field of natural language processing by models pre-trained with a large-scale corpus, including GPT [33], BERT [7] and XLNet [41]. Even in computer vision, recent advances in unsupervised representation learning [40,16,3], which exploit the inductive bias of data structure, are shaking the long-term dominated status of representations learned in a supervised way. Further, a wide range of handcrafted pretext tasks have been proposed for unsupervised representation learning, such as relative patch prediction [8], solving jigsaw puzzles [29], colorization [43], multi-modal prediction [32], etc.

2.2 Contrastive Learning

Specifically, a variety of unsupervised pretext tasks are based on some forms of contrastive learning, in which the instance discrimination approach [40,16,3] is one of the most general forms. Other variants of contrastive learning methods include contrastive predictive learning (CPC) [30] and colorization contrasting [37]. Recent advances of deep contrastive learning benefit from contrasting positive keys against *very large* number of negative keys. Therefore, how to efficiently generate keys becomes a fundamental problem in contrastive learning. To achieve this goal, [9] explored the effectiveness of in-batch samples, [40] proposed to use a memory bank to store all representations of the dataset, [16] further replaced a memory bank with the momentum contrast (MoCo) to be memory-efficient, and [3] showed that a brute-force huge batch of keys works well. A new branch of works [13,4] explores contrastive learning without negative keys.

2.3 Fine-tuning

Fine-tuning a model from supervised pre-trained models has been empirically explored in [21] by launching a systematic investigation with grid search of the hyper-parameters. During the past years, a line of fine-tuning methods have been proposed to exploit the inductive bias of pre-trained models: L2-SP [24] drives the weight parameters of target task to the pre-trained values by imposing L2 constraint based on the inductive bias of parameter; DELTA [23] computes channel-wise discriminative knowledge to reweight the feature map regularization with an attention mechanism based on the inductive bias of behavior; BSS [5] penalizes smaller singular values to suppress nontransferable spectral components based on singular values.

Other fine-tuning methods including learning with similarity preserving [20] or learning without forgetting [25] also work well on some downstream classification tasks. However, the existing fine-tuning methods mainly focus on leveraging the knowledge of the downstream labels with a cross-entropy loss. Intuitively, encouraging a model to capture the label information and intrinsic structure simultaneously may help the model transition between the upstream unsupervised models to the downstream classification tasks. In natural language processing, GPT [33,34] has employed a strategy that jointly optimizes unsupervised training criteria while fine-tuning with supervision. However, we empirically found that trivially following this kind of force-combination between supervised learning loss and unsupervised contrastive learning loss is beneficial but limited. The plausible reason is that these two loss functions will contradict with each other and result in a very different but not discriminative feature structure compared to that of the supervised cross-entropy loss, as revealed by a prior study shown in Figure 3.

3 Backgrounds

It is worth noting that the principles of contrastive learning actually can date back very far [1,15,14]. The key idea of contrastive learning is to maximize the likelihood of the input distribution $p(\mathbf{x}|D)$ conditioned on the dataset D contrasting to the artificial noise distribution $p_n(\mathbf{x})$, also known as noise-contrastive estimation (NCE). Later, [12] pointed out the relations between generative adversarial networks and noise-contrastive estimation. Meanwhile, [30] revealed that contrastive learning is related to mutual information between a query and the corresponding positive key, which is known as InfoNCE. Considering a query $\mathbf{q}$ with a large key pool $\mathcal{K} = \{\mathbf{k}_0, \mathbf{k}_1, \mathbf{k}_2, \cdots, \mathbf{k}_{|\mathcal{K}|}\}$ where $|\mathcal{K}|$ is the number of keys, the kind of non-parametric form [30,40] of contrastive loss can be defined as

$$L_{\text{InfoNCE}} = -\log \frac{\exp(\mathbf{q} \cdot \mathbf{k}_0 / \tau)}{\sum_{i=0}^{|\mathcal{K}|} \exp(\mathbf{q} \cdot \mathbf{k}_i / \tau)}, \tag{1}$$

where τ is the temperature hyper-parameter. Note that $\mathbf{k}_0$ is the only positive key that $\mathbf{q}$ matches while negative keys $\{\mathbf{k}_1, \mathbf{k}_2, \cdots, \mathbf{k}_{|\mathcal{K}|}\}$ are selected from a dynamic queue which iteratively and progressively replaces the oldest samples by the newly-generated keys. Intuitively, contrastive learning can be defined as a query-key pair matching problem, where a contrastive loss is a $(|\mathcal{K}| + 1)$-way cross-entropy loss to distinguish $\mathbf{k}_0$ from a large key pool. A contrastive loss is to maximize the similarity between the query and the corresponding positive key $\mathbf{k}_0$ since they are extracted from different views of the same data example.

4 Methods

In inductive transfer learning, a.k.a. fine-tuning, we have access to a target dataset $\mathcal{D} = \{(\mathbf{x}_i, y_i)\}_{i=1}^N$ with N labeled examples and a pre-trained model $\mathcal{M}$ attained on a large-scale source dataset. Instantiated as a deep neural network, $\mathcal{M}$ usually consists of a pre-trained backbone f_0 and a pre-trained head g_0 whose fine-tuned ones are denoted by f and g, respectively. Following the common practice of fine-tuning, f is initialized as f_0. Contrarily, g is usually a randomly initialized fully-connected layer parameterized by $\mathbf{W}$, since the target dataset usually has a label space of size C different from that of pre-trained models.

4.1 Vanilla Fine-tuning of Pre-trained Representations

For each query sample $\mathbf{x}_i^{\mathrm{q}}$ from the target dataset, we can first utilize a pre-trained feature encoder $f(\cdot)$ to extract its pre-trained representation as $\mathbf{h}_i^{\mathrm{q}} = f(\mathbf{x}_i^{\mathrm{q}})$. Without any additional assumptions, the pre-trained feature encoder $f(\cdot)$ can be commonly used network backbones according to the downstream tasks, including ResNet [18] and DenseNet [19] for supervised pre-trained models, and MoCo [16] and SimCLR [3] for unsupervised pre-trained models.

Given a pre-trained representation $\mathbf{h}_i^{\mathrm{q}}$, a fundamental step of vanilla fine-tuning is to feedforward the representation $\mathbf{h}_i^{\mathrm{q}}$ into a C-way classifier $g(\cdot)$, in which C is the number of categories for the downstream classification task. Denote the parameters of the classifier $g(\cdot)$ as $\mathbf{W} = [\mathbf{w}_1, \mathbf{w}_2, \cdots, \mathbf{w}_C]$, where $\mathbf{w}_j$ corresponds to the parameter for the j-th class. Given the training dataset of the downstream task, the parameters of the classifier and the backbone can be updated by optimizing a standard cross-entropy (CE) loss as

$$L_{\mathrm{CE}} = -\sum_{i=1}^{N} \log \frac{\exp(\mathbf{w}_{y_i} \cdot \mathbf{h}_i^{\mathrm{q}})}{\sum_{j=1}^{C} \exp(\mathbf{w}_j \cdot \mathbf{h}_i^{\mathrm{q}})}. \tag{2}$$

With a CE loss on the target labeled dataset, the vanilla fine-tuning approach leverages the discriminative knowledge of labels. As later experiments revealed, the vanilla fine-tuning approach underperforms in a low data regime since it will easily suffer from heavy overfitting on the limited target labeled dataset. Regarding the success of supervised and unsupervised pre-training approaches, we realize that it is significant to further exploit the *discriminative knowledge* and *intrinsic structure* of the downstream task for fine-tuning. To achieve this, we propose a contrastive cross-entropy (CCE) loss on the classifier head to further exploit the label information and a categorical contrastive learning (CCL) loss on the projector head to capture the intrinsic structure of target data, which will be detailed orderly in the following sections.

4.2 Classifier Head with Contrastive Cross-Entropy Loss

First, we delve into the cross-entropy loss to figure out the mechanism of how it exploits label information. For each instance-class pair $(\mathbf{x}_i, y_i)$ on a given dataset,

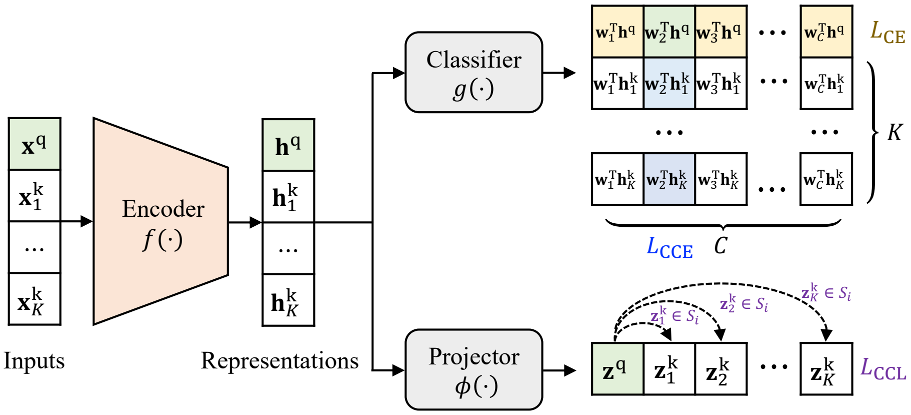

$\mathbf{x}^q$
$\mathbf{x}_1^k$
...
$\mathbf{x}_K^k$
Inputs
Encoder $f(\cdot)$
$\mathbf{h}^q$
$\mathbf{h}_1^k$
...
$\mathbf{h}_K^k$
Representations
Classifier $g(\cdot)$
$\mathbf{w}_1^T\mathbf{h}^q$ $\mathbf{w}_2^T\mathbf{h}^q$ $\mathbf{w}_3^T\mathbf{h}^q$ ... $\mathbf{w}_C^T\mathbf{h}^q$
$\mathbf{w}_1^T\mathbf{h}_1^k$ $\mathbf{w}_2^T\mathbf{h}_1^k$ $\mathbf{w}_3^T\mathbf{h}_1^k$ ... $\mathbf{w}_C^T\mathbf{h}_1^k$
$\mathbf{w}_1^T\mathbf{h}_K^k$ $\mathbf{w}_2^T\mathbf{h}_K^k$ $\mathbf{w}_3^T\mathbf{h}_K^k$ ... $\mathbf{w}_C^T\mathbf{h}_K^k$
L_{CE}
K
L_{CCE} C
Projector $\phi(\cdot)$
$\mathbf{z}^q$ $\mathbf{z}_1^k$ $\mathbf{z}_2^k$ ... $\mathbf{z}_K^k$
$\mathbf{z}_1^k \in S_i$
$\mathbf{z}_2^k \in S_i$
$\mathbf{z}_K^k \in S_i$
L_{CCL}

Fig. 1: The **Bi-tuning** approach, which includes an encoder for pre-trained representations, a classifer head and a projector head. Bi-tuning enables a dual fine-tuning mechanism: a contrastive cross-entropy (CCE) loss on the classifier head to exploit label information and a categorical contrastive learning (CCL) loss on the projector head to capture the intrinsic structure of target data.

the predicted output of the fine-tuned model is a probability vector of size C where C is the number of categories. From another perspective, the cross-entropy loss of vanilla fine-tuning can be regarded as a *class-wise championship*, i.e., the category that is the same as the ground-truth label of each instance is expected to win the game. As revealed in Figure 1, to find the correct class, the cross-entropy loss performs *column-wise championship* for each instance.

To further exploit the label information of the downstream task, we propose an alternative form of the conventional cross-entropy loss on the classifier head, named contrastive cross-entropy loss L_{CCE}. Correspondingly, L_{CCE} performs a *row-wise championship* for each class while L_{CE} is a *column-wise championship* for each instance. Instead of operating loss computation along the class dimension (i.e., the number of classes C), L_{CCE} operates along the key-set dimension (i.e., the number of keys $K+1$). As an instance-wise championship in L_{CCE}, the instance nearest to the prototype of each class is expected to win the game. For each sample $(\mathbf{x}_i, y_i)$ in the target dataset, the representation encoded by f is $\mathbf{h}_i$. For clarity, we focus on a particular data example $(\mathbf{x}, y)$ and omit the subscript i. The proposed L_{CCE} for each data example is formulated as

$$L_{\mathrm{CCE}} = -\frac{1}{|\mathcal{K}_p|} \sum_{\mathbf{h}^+ \in \mathcal{K}_p} \log \frac{\exp(\mathbf{w}_y \cdot \mathbf{h}^+ / \tau)}{\sum_{\mathbf{h} \in \mathcal{K}_p \cup \mathcal{K}_n} \exp(\mathbf{w}_y \cdot \mathbf{h} / \tau)}, \tag{3}$$

where $\mathcal{K}_p$ is the positive key set, $\mathcal{K}_n$ is the negative key set, and τ is the hyper-parameter for temperature scaling. Note that, $\mathcal{K}_p$ consists of $\mathbf{k}_0$ and keys with the same label y where $\mathbf{k}_0$ is extracted from a differently augmented view of the query $\mathbf{q}$. On the contrary, $\mathcal{K}_n$ includes examples from other classes

$\{1, 2, \cdots, C\}\backslash y$. Here, $\mathbf{h}$'s are samples from the hidden key pool produced by the key generating mechanism (except $\mathbf{h}^{\mathrm{q}}$). Without loss of generality, we adopt the key generating approach in Momentum Contrast (MoCo) [16] as our default one due to its simplicity, high-efficacy, and memory-efficient implementation. In summary, by encouraging instances in the training dataset to approach towards their corresponding class prototypes (feature center of the same-label samples), L_{CCE} further exploits the label information of the target dataset and tends to achieve a more compact intra-class structure than the vanilla fine-tuning.

4.3 Projector Head with Categorical Contrastive Learning Loss

Till now, we have proposed the contrastive cross-entropy loss on the classifier head to fully exploit the label information. However, this kind of loss function may still fall short in capturing the intrinsic structure. Inspired by the remarkable success of unsupervised pre-training, which also aims at modeling the intrinsic structure in data, we first introduce a projector $\phi(\cdot)$ which is usually off the shelf to embed a pre-trained representation $\mathbf{h}_i^{\mathrm{q}}$ into a latent metric space as $\mathbf{z}_i^{\mathrm{q}}$. Intuitively, we apply the standard contrastive learning loss (InfoNCE) defined in Eq. (1) on the target dataset to capture intrinsic structure in data. However, the InfoNCE loss assumes that there is a *single* key $\mathbf{k}_+$ (also denoted as $\mathbf{k}_0$) in the dictionary to match the given query $\mathbf{q}$, which implicitly requires every instance to belong to an individual class. In other words, it regards every sample in the key pool as a negative sample except $\mathbf{k}_+$, which requires minimizing the similarity between the query with all negative samples. Yet, from the perspective of discriminative learning, we should maximize inter-class distance but minimize intra-class distance. As a consequence, those samples with the same class as the query sample should not be treated as negative samples, and the similarity between them should be maximized.

As aforementioned, if we simply apply InfoNCE loss on the labeled downstream dataset, it will result in an extremely different but not discriminative feature structure compared with that of the supervised cross-entropy loss, making the classifier struggle. Obviously, this dilemma reveals that the naive combination of the supervised cross-entropy loss and the unsupervised contrastive loss is not an optimal solution for fine-tuning, which is also backed by our experiments in Table 3. To capture the label information and intrinsic structure simultaneously, we propose a novel categorical contrastive loss L_{CCL} on the projector head based on the following hypothesis: when we fine-tune a pre-trained model to a downstream task, it is reasonable to regard other keys in the same class as the positive keys that the query matches. In this way, L_{CCL} expands the scope of positive keys from *single instance* to *a set of instances*, resulting in more harmonious collaboration between the supervised and unsupervised learning mechanisms. Similar to the format of the InfoNCE loss, L_{CCL} is defined as

$$L_{\mathrm{CCL}} = -\frac{1}{|\mathcal{K}_p|} \sum_{\mathbf{z}^+ \in \mathcal{K}_p} \log \frac{\exp(\mathbf{z}^{\mathrm{q}} \cdot \mathbf{z}^+ / \tau)}{\sum_{\mathbf{z} \in \mathcal{K}_p \cup \mathcal{K}_n} \exp(\mathbf{z}^{\mathrm{q}} \cdot \mathbf{z} / \tau)}, \tag{4}$$

with notations in parallel to that of Eq. (3). Note that the sum is taken over all positive keys, indicating that there may be more than one positive key for a single query, i.e., $|\mathcal{K}_p| \geq 1$.

We provide an intuitive explanation of why L_{CCL} is complementary to vanilla fine-tuning. While using the standard cross-entropy loss, we can learn a hyperplane for discriminating each class from the other classes, and the instances of each class are only required to be far away from its associated hyperplane—they are not required to form into a compact structure in the metric space. As for the proposed categorical contrastive loss, besides requiring the instances of each class to stay far away from those of the other classes, we further require that they should form a compact structure in the metric space. This is exactly the advantage of *contrast-by-metric* over *discriminate-by-hyperplane*, which better facilitates the fine-tuning on downstream task.

4.4 Optimization Objective of Bi-Tuning

Finally, we reach a novel fine-tuning approach for efficient transfer from both supervised and unsupervised pre-trained models. Due to the dual-head design, the approach is coined **Bi-tuning**, which jointly optimizes the contrastive cross-entropy loss on classifier head and the categorical contrastive learning loss on projector head, as well as the standard cross-entropy loss, in an end-to-end deep architecture. The overall loss function of Bi-tuning is

$$\min_{\Theta=\{f,g,\phi\}} L_{\mathrm{CE}} + L_{\mathrm{CCE}} + L_{\mathrm{CCL}}, \tag{5}$$

where Θ denotes the parameters of the encoder f, the classifier head g and the projector head ϕ. Desirably, since the magnitude of the above loss terms is comparable, we empirically find that there is no need to introduce any extra hyper-parameters to trade-off them. This simplicity makes Bi-tuning easy to be applied to different datasets or tasks. The full portrait of Bi-tuning is shown in Figure 1.

5 Experiments

We follow the common fine-tuning principle described in [42], replacing the last task-specific layer in the classifier head with a randomly initialized fully connected layer whose learning rate is 10 times of that for pre-trained parameters. Meanwhile, the projector head is set to be another randomly initialized fully connected layer. For the key generating mechanisms, we follow the style in [16], employing a momentum contrast branch with a default momentum coefficient $m = 0.999$ and two cached queues both normalized by their L2-norm [40] with dimensions of 2048 and 128 respectively. For each task, the best learning rate is selected by cross-validation under a 100% sampling rate and applied to all four sampling rates. Queue size K is set as $8, 16, 24, 32$ for each category according to the dataset scales, respectively. Other hyper-parameters in Bi-tuning are

Table 1: Top-1 accuracy on various datasets using ResNet-50 by *supervised* pre-training.

Dataset	Method	Sampling Rates			
		25%	50%	75%	100%
CUB	Fine-tuning	61.36±0.11	73.61±0.23	78.49±0.18	80.74±0.15
	L2SP [24]	61.21±0.19	72.99±0.13	78.11±0.17	80.92±0.22
	DELTA [23]	62.89±0.11	74.35±0.28	79.18±0.24	81.33±0.24
	BSS [5]	64.69±0.31	74.96±0.21	78.91±0.15	81.52±0.11
	Bi-tuning	**67.47**±0.08	**77.17**±0.13	**81.07**±0.09	**82.93**±0.23
Cars	Fine-tuning	56.45±0.21	75.24±0.17	83.22±0.17	86.22±0.12
	L2SP [24]	56.29±0.21	75.62±0.32	83.60±0.13	85.85±0.12
	DELTA [23]	58.74±0.23	76.53±0.08	84.53±0.29	86.01±0.37
	BSS [5]	59.74±0.14	76.78±0.16	85.06±0.13	87.64±0.21
	Bi-tuning	**66.15**±0.20	**81.10**±0.07	**86.07**±0.23	**88.47**±0.11
Aircraft	Fine-tuning	51.25±0.18	67.12±0.41	75.22±0.09	79.18±0.20
	L2SP [24]	51.07±0.45	67.46±0.22	75.06±0.45	79.07±0.21
	DELTA [23]	53.71±0.30	68.51±0.24	76.51±0.55	80.34±0.14
	BSS [5]	53.38±0.22	69.19±0.18	76.39±0.22	80.83±0.32
	Bi-tuning	**58.27**±0.26	**72.40**±0.22	**80.77**±0.10	**84.01**±0.33

fixed for all experiments. The temperature τ in Eq. (3) and Eq. (4) is set as 0.07 [40]. The trade-off coefficients between these three losses are kept as 1 since the magnitude of the loss terms is comparable. All tasks are optimized using SGD with a momentum 0.9. All results in this section are averaged over 5 trials, and standard deviations are provided.

5.1 Supervised Pre-trained Representations

Standard benchmarks. We first verify our approach on three fine-grained classification benchmarks: `CUB-200-2011` [38] (with 11788 images for 200 bird species), `Stanford Cars` [22] (containing 16185 images of 196 classes of cars) and `FGVC Aircraft` [28] (containing 10000 samples 100 different aircraft variants). For each benchmark, we create four configurations which randomly sample $25\%, 50\%, 75\%$, and 100% of training data for each class respectively, to reveal the detailed effect while fine-tuning to different data scales. We choose recent fine-tuning technologies: L2-SP [24], DELTA [23], and the state-of-the-art method BSS [5], as competitors of Bi-tuning while regarding vanilla fine-tuning as a baseline. Note that vanilla fine-tuning is a strong baseline when sufficient data is provided. Results are averaged over 5 trials. As shown in Table 1, Bi-tuning significantly outperforms all competitors across all three benchmarks by large margins (e.g. 10.7% absolute rise on *CUB* with a sampling rate of 25%). Note that even under 100% sampling rate, Bi-tuning still outperforms others.

Table 2: Top-1 accuracy on COCO-70 dataset using DenseNet-121 by *supervised* pre-training.

Method	Sampling Rates			
	25%	50%	75%	100%
Fine-tuning	80.01±0.25	82.50±0.25	83.43±0.18	84.41±0.22
L2SP [24]	80.57±0.47	80.67±0.29	83.71±0.24	84.78±0.16
DELTA [23]	76.39±0.37	79.72±0.24	83.01±0.11	84.66±0.08
BSS [5]	77.29±0.15	80.74±0.22	83.89±0.09	84.71±0.13
Bi-tuning	**80.68**±0.23	**83.48**±0.13	**84.16**±0.05	**85.41**±0.23

Large-scale benchmarks. Previous fine-tuning methods mainly focus on improving performance under low-data regime paradigms. We further extend Bi-tuning to large-scale paradigms. We use annotations of the COCO dataset [26] to construct a large-scale classification dataset, cropping object with padding for each image and removing minimal items (with height and width less than 50 pixels), resulting in a large-scale dataset containing 70 classes with more than 1000 images per category. The scale is comparable to ImageNet in terms of the number of samples per class. On this constructed large-scale dataset named COCO-70, Bi-tuning is also evaluated under four sampling rate configurations. Since even the 25% sampling rate of COCO-70 is much larger than each benchmark in Section 5.1, previous fine-tuning competitors show micro contributions to these paradigms. Results in Table 2 reveal that Bi-tuning brings general gains for all tasks. We hypothesize that the intrinsic structure introduced by Bi-tuning contributes substantially.

5.2 Unsupervised Pre-trained Representations

Representations of MoCo [16]. In this round, we use ResNet-50 pre-trained unsupervisedly via MoCo on ImageNet as the backbone. Since suffering from the large discrepancy between unsupervised pre-trained representations and downstream classification tasks as demonstrated in Figure 3, previous fine-tuning competitors usually perform very poorly. Hence we only compare *Bi-tuning* to the state-of-the-art method *BSS* [5] and vanilla fine-tuning as baselines. Besides, we add two intuitively related baselines: (1) **GPT***, which follows a GPT [33,34] fine-tuning style but replaces its predictive loss with the contrastive loss; (2) **Center** loss, which introduces compactness of intra-class variations [39] that is effective in recognition tasks. As reported in Table 3, trivially borrowing fine-tuning strategy in GPT [33] or center loss brings tiny benefits, and is even harmful on some datasets, e.g. CUB. Bi-tuning yields consistent gains on all fine-tuning tasks of unsupervised representations.

Other unsupervised pre-trained representations. To justify Bi-tuning's general efficacy, we extend our method to unsupervised representations by other pre-training methods. Bi-tuning is applied to MoCo (version 2) [16], SimCLR [3],

Table 3: Top-1 accuracy on various datasets using ResNet-50 *unsupervised* pre-training by MoCo.

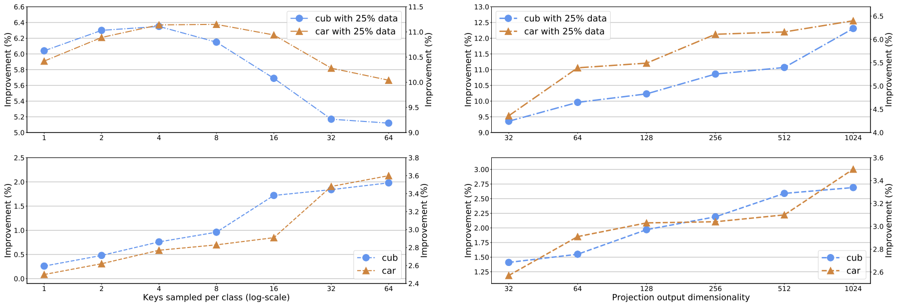
Dataset	Method	Sampling Rates			
		25%	50%	75%	100%
CUB	Fine-tuning	38.57±0.13	58.97±0.16	69.55±0.18	74.35±0.18
	GPT* [34]	36.43±0.17	57.62±0.14	67.82±0.05	72.95±0.29
	Center [39]	42.53±0.41	62.15±0.51	70.86±0.39	75.61±0.33
	BSS [5]	41.73±0.14	59.15±0.21	69.93±0.19	74.16±0.09
	Bi-tuning	**50.54**±0.23	**66.88**±0.13	**74.27**±0.05	**77.14**±0.23
Cars	Fine-tuning	62.40±0.26	81.55±0.36	88.07±0.19	89.81±0.48
	GPT* [34]	65.83±0.27	82.39±0.17	88.62±0.11	90.56±0.18
	Center [39]	67.57±0.12	82.78±0.30	88.55±0.24	89.95±0.1
	BSS [5]	62.13±0.22	81.72±0.22	88.32±0.17	90.41±0.15
	Bi-tuning	**69.44**±0.32	**84.41**±0.07	**89.32**±0.23	**90.88**±0.13
Aircraft	Fine-tuning	58.98±0.54	77.39±0.31	84.82±0.24	87.35±0.17
	GPT* [34]	60.70±0.08	78.93±0.17	85.09±0.10	87.56±0.15
	Center [39]	62.23±0.09	79.30±0.14	85.20±0.41	87.52±0.20
	BSS [5]	60.13±0.32	77.98±0.29	84.85±0.21	87.25±0.07
	Bi-tuning	**63.16**±0.26	**79.98**±0.22	**86.23**±0.29	**88.55**±0.38

InsDisc [40], Deep Cluster [2], CMC [37] on `Car` dataset with 100% training data. Table 4 is a strong signal that Bi-tuning is not bound to specific pre-training pretext tasks.

Analysis on components of contrastive learning. Recent advances in contrastive learning, i.e. momentum contrast [16] and memory bank [40] can be plugged into Bi-tuning smoothly to achieve similar performance and the detailed discussions are deferred to Appendix. Previous works [16,3] reveal that a large amount of contrasts is crucial to contrastive learning. In Figure 2(a), we report the sensitivity of the numbers of sampling keys in Bi-tuning (MoCo) under 25% and 100% sampling ratio configurations. Note that CUB has various categories with a few images in each category. We let K balancedly sampled from every category to simplify our analysis here. Figure 2(a) shows that though a larger key pool is beneficial, we cannot expand the key pool due to the limit of training data, which may lose sampling stochasticity during training. This result suggests that there is a trade-off between stochasticity and a large number of keys. [3] pointed out that the dimension of the projector also has a big impact. The sensitivity of the dimension of the projector head is also presented in Figure 2(b). Note that the unsupervised pre-trained model (e.g., MoCo) may provide an off-the-shelf projector, fine-tuning or re-initializing it is almost the same (90.88 vs. 90.78 on `Car` when L is 128).

Table 4: Top-1 accuracy on `Car` dataset (100%) with different *unsupervised* pre-trained representations.

Pre-training Method	Fine-tuning	**Bi-tuning**
Deep Cluster [2]	83.90±0.48	**87.71**±0.34
InsDisc [40]	86.59±0.22	**89.54**±0.25
CMC [37]	86.71±0.62	**88.35**±0.44
MoCov2 [16]	90.15±0.48	90.79±0.34
SimCLR(1×) [3]	89.30±0.18	**90.84**±0.22
SimCLR(2×) [3]	91.22±0.19	**91.93**±0.19



(a) Number of Sampling Keys (K)

(b) Dimension of Projector Head (L)

Fig. 2: Sensitivity analysis of hyper-parameters K and L for Bi-tuning.

5.3 Collaborative Effect of Loss Functions

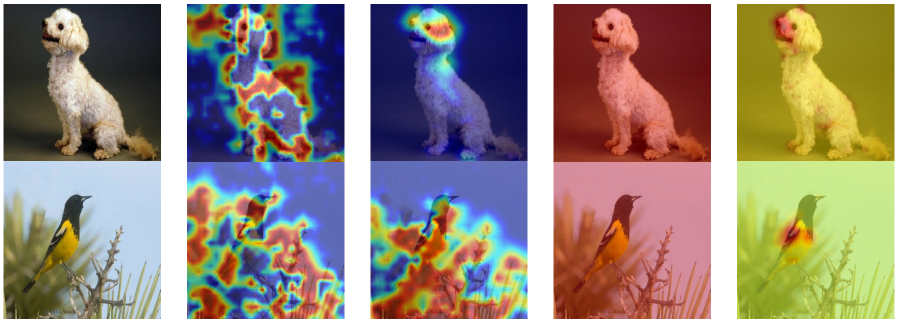
As shown in Table 5, using either contrastive cross-entropy (CCE) or categorical contrastive (CCL) with vanilla cross-entropy (CE) already achieves relatively good results. These experiments prove that there is collaborative effect between CCE and CCL loss empirically. It is worth mentioning that CCE and CCL can work independently of CE (see the fourth row in Table 5), while we optimize these three losses simultaneously to yield the best result. As discussed in prior sections, we hypothesize that Bi-tuning helps fine-tuning models characterize the intrinsic structure of training data when using CCE and CCL simultaneously.

5.4 Interpretable Visualization of Representations

We use a popular visualization tool proposed in [11] to give a interpretable visualization as shown in Figure 3. Note that 3(a) is the original image, Figure 3(b), Figure 3(c) and Figure 3(d) are respectively obtained from a randomly initialized model, a supervised pre-trained model on ImageNet, and an unsupervised pre-trained model via MoCov1 [16]. We infer that supervised pre-training will obtain representations focusing on the discriminative part and ignoring

Table 5: Collaborative effect in Bi-tuning on CUB-200-2011 using ResNet-50 pre-trained by MoCo.

Loss Function			Sample Rate			
CE	CCE	CCL	25%	50%	75%	100%
✓	✗	✗	38.57±0.13	58.97±0.16	69.55±0.18	74.35±0.18
✓	✓	✗	45.42±0.11	64.33±0.28	71.56±0.30	75.82±0.21
✓	✗	✓	41.09±0.23	60.77±0.31	70.30±0.29	75.30±0.20
✗	✓	✓	47.70±0.41	64.77±0.15	71.69±0.11	76.54±0.24
✓	✓	✓	**50.54**±0.23	**66.88**±0.13	**74.27**±0.05	**77.12**±0.23



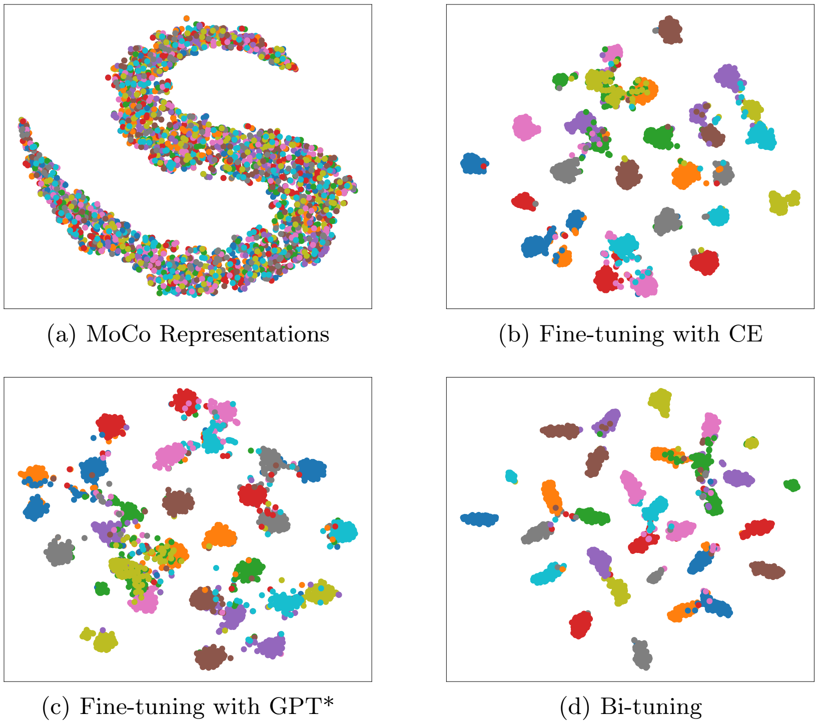
(a) Original (b) Random (c) Supervised (d) MoCo (e) Bi-tuning

Fig. 3: Interpretable visualization of learned representations via various training methods.

the background part. In contrast, unsupervised pre-training pays uninformative attention to every location of an input image. This could be the reason that why fine-tuning unsupervised representations is harder than their supervised counterparts. Impressively, Bi-tuning in 3(e) captures both local details and global category-structures. Bi-tuning benefits from both the supervised discriminative knowledge and the unsupervised intrinsic structure. And this is the reason why Bi-tuning works well.

5.5 Visualization by t-SNE

We train the t-SNE [27] visualization model on the MoCo representations fine-tuned on Pets dataset [31]. Visualization of the validation set is shown in Figure 4. Note that representations in Figure 4(a) do not present good classification structures. Figure 4(c) suggests that forcefully combining the unsupervised loss as GPT [34] may cause conflict with CE and clutter the classification boundaries. Figure 4(d) suggests that Bi-tuning encourages the fine-tuning model to learn



(a) MoCo Representations

(b) Fine-tuning with CE

(c) Fine-tuning with GPT*

(d) Bi-tuning

Fig. 4: T-SNE [27] of representations on Pets [31].

better intrinsic structure besides the label information. Therefore, Bi-tuning presents the best classification boundaries as well as intrinsic structures.

6 Conclusion

In this paper, we propose a general Bi-tuning approach to fine-tuning both supervised and unsupervised representations. Bi-tuning generalizes the standard fine-tuning with an encoder for pre-trained representations, a classifier head and a projector head for exploring both the discriminative knowledge of labels and the intrinsic structure of data, which are trained end-to-end by two novel loss functions. Bi-tuning yields state-of-the-art results for fine-tuning tasks on both supervised and unsupervised pre-trained models by large margins.

Acknowledgements This work was supported by the National Key Research and Development Plan (2020AAA0109201), National Natural Science Foundation of China (62022050 and 62021002), and Beijing Nova Program (Z201100006820041).

References

1. Becker, S., Hinton, G.E.: Self-organizing neural network that discovers surfaces in random-dot stereograms. Nature **355**(6356), 161–163 (1992)
2. Caron, M., Bojanowski, P., Joulin, A., Douze, M.: Deep clustering for unsupervised learning of visual features. In: ECCV (2018)
3. Chen, T., Kornblith, S., Norouzi, M., Hinton, G.: A Simple Framework for Contrastive Learning of Visual Representations. arXiv preprint arXiv:2002.05709 (2020)
4. Chen, X., He, K.: Exploring simple siamese representation learning. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 15750–15758 (2021)
5. Chen, X., Wang, S., Fu, B., Long, M., Wang, J.: Catastrophic forgetting meets negative transfer: Batch spectral shrinkage for safe transfer learning. In: NeurIPS. pp. 1906–1916 (2019)
6. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: Imagenet: A large-scale hierarchical image database. In: CVPR. pp. 248–255. Ieee (2009)
7. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805 (2018)
8. Doersch, C., Gupta, A., Efros, A.A.: Unsupervised visual representation learning by context prediction. In: ICCV. pp. 1422–1430 (2015)
9. Doersch, C., Zisserman, A.: Multi-task self-supervised visual learning. In: ICCV. pp. 2051–2060 (2017)
10. Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., et al.: An image is worth 16x16 words: Transformers for image recognition at scale. International Conference on Learning Representations (2020)
11. Fong, R.C., Vedaldi, A.: Interpretable explanations of black boxes by meaningful perturbation. In: ICCV. pp. 3429–3437 (2017)
12. Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., Bengio, Y.: Generative adversarial nets. In: NeurIPS. pp. 2672–2680 (2014)
13. Grill, J.B., Strub, F., Altché, F., Tallec, C., Richemond, P., Buchatskaya, E., Doersch, C., Avila Pires, B., Guo, Z., Gheshlaghi Azar, M., et al.: Bootstrap your own latent-a new approach to self-supervised learning. Advances in neural information processing systems **33**, 21271–21284 (2020)
14. Gutmann, M., Hyvärinen, A.: Noise-contrastive estimation: A new estimation principle for unnormalized statistical models. In: AISTATS (2010)
15. Hadsell, R., Chopra, S., Lecun, Y.: Dimensionality reduction by learning an invariant mapping. In: CVPR (2006)
16. He, K., Fan, H., Wu, Y., Xie, S., Girshick, R.: Momentum contrast for unsupervised visual representation learning. In: CVPR (2020)
17. He, K., Girshick, R.B., Dollár, P.: Rethinking imagenet pre-training. In: ICCV. pp. 4917–4926. IEEE (2019)
18. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: CVPR. pp. 770–778 (2016)
19. Huang, G., Liu, Z., van der Maaten, L., Weinberger, K.Q.: Densely connected convolutional networks. In: CVPR. IEEE Computer Society (2017)
20. Kang, Z., Lu, X., Lu, Y., Peng, C., Xu, Z.: Structure learning with similarity preserving. arXiv preprint arXiv:1912.01197 (2019)

21. Kornblith, S., Shlens, J., Le, Q.V.: Do better imagenet models transfer better? In: CVPR. pp. 2661–2671 (2019)
22. Krause, J., Stark, M., Deng, J., Fei-Fei, L.: 3d object representations for fine-grained categorization. In: 3dRR (2013)
23. Li, X., Xiong, H., Wang, H., Rao, Y., Liu, L., Huan, J.: Delta: Deep learning transfer using feature map with attention for convolutional networks. In: ICLR (2019)
24. Li, X., Grandvalet, Y., Davoine, F.: Explicit inductive bias for transfer learning with convolutional networks. In: Dy, J.G., Krause, A. (eds.) ICML (2018)
25. Li, Z., Hoiem, D.: Learning without forgetting. TPAMI **40**(12), 2935–2947 (2017)
26. Lin, T.Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., Zitnick, C.L.: Microsoft coco: Common objects in context. In: ECCV. pp. 740–755 (2014)
27. Maaten, L.v.d., Hinton, G.: Visualizing data using t-sne. Journal of machine learning research **9**(Nov), 2579–2605 (2008)
28. Maji, S., Rahtu, E., Kannala, J., Blaschko, M., Vedaldi, A.: Fine-grained visual classification of aircraft. Technical report (2013)
29. Noroozi, M., Favaro, P.: Unsupervised learning of visual representations by solving jigsaw puzzles. In: ECCV. pp. 69–84. Springer (2016)
30. van den Oord, A., Li, Y., Vinyals, O.: Representation learning with contrastive predictive coding (2018)
31. Parkhi, O.M., Vedaldi, A., Zisserman, A., Jawahar, C.: Cats and dogs. In: CVPR. pp. 3498–3505. IEEE (2012)
32. Radford, A., Kim, J.W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., et al.: Learning transferable visual models from natural language supervision. In: International conference on machine learning. pp. 8748–8763. PMLR (2021)
33. Radford, A., Sutskever, I.: Improving language understanding by generative pre-training. In: arxiv (2018)
34. Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., Sutskever, I.: Language models are unsupervised multitask learners. OpenAI Blog **1**(8), 9 (2019)
35. Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S.E., Anguelov, D., Erhan, D., Vanhoucke, V., Rabinovich, A.: Going deeper with convolutions. In: CVPR (2015)
36. Tan, M., Le, Q.V.: Efficientnet: Rethinking model scaling for convolutional neural networks. ICML (2019)
37. Tian, Y., Krishnan, D., Isola, P.: Contrastive multiview coding. arXiv preprint arXiv:1906.05849 (2019)
38. Welinder, P., Branson, S., Mita, T., Wah, C., Schroff, F., Belongie, S., Perona, P.: Caltech-UCSD Birds 200. Tech. Rep. CNS-TR-2010-001, California Institute of Technology (2010)
39. Wen, Y., Zhang, K., Li, Z., Qiao, Y.: A discriminative feature learning approach for deep face recognition. In: Leibe, B., Matas, J., Sebe, N., Welling, M. (eds.) ECCV. pp. 499–515 (2016)
40. Wu, Z., Xiong, Y., Yu, S.X., Lin, D.: Unsupervised feature learning via non-parametric instance discrimination. In: CVPR (2018)
41. Yang, Z., Dai, Z., Yang, Y., Carbonell, J.G., Salakhutdinov, R., Le, Q.V.: Xlnet: Generalized autoregressive pretraining for language understanding. In: NeurIPS (2019)
42. Yosinski, J., Clune, J., Bengio, Y., Lipson, H.: How transferable are features in deep neural networks? In: Advances in neural information processing systems. pp. 3320–3328 (2014)
43. Zhang, R., Isola, P., Efros, A.A.: Colorful image colorization. In: ECCV (2016)