IMPROVED FINE-TUNING BY LEVERAGING PRE-TRAINING DATA: THEORY AND PRACTICE

Anonymous authors
Paper under double-blind review

ABSTRACT

Fine-tuning a pre-trained model on the target data is widely used in many deep learning applications, especially for small data sets. However, recent studies have empirically shown that this training strategy offers almost no benefit in computer vision tasks over training from scratch. In this work, we first revisit this observation from the perspective of generalization analysis which is popular in learning theory. Our theory reveals that the final prediction precision has a weak dependency on the pre-trained model. Besides the pre-trained model, data for pre-training are also available for fine-tuning. The observation from pre-trained model inspires us to leverage pre-training data for fine-tuning. With the theoretical analysis, we find that the final performance on target data can be improved when the appropriate pre-training data are included in fine-tuning. Therefore, we propose to select a subset from pre-training data to help the optimization on the target data. A novel selection algorithm is developed according to our analysis. Extensive experiments on 8 benchmark data sets verify the effectiveness of the proposed fine-tuning pipeline.

1 INTRODUCTION

After the success on ImageNet (Deng et al., 2009), deep learning attracts much attention and improves the performance of various tasks significantly, e.g., object detection (Ren et al., 2015), semantic segmentation(Chen et al., 2017), etc. However, when applying deep learning on the data set with a limited number of examples, researchers find that it is easy to incur the over-fitting problem and result in the performance degradation when generalizing. The phenomenon can be attributed to the massive number of parameters in deep neural networks, which can fit small data sets perfectly.

Considering that labeling is expensive, it is inapplicable to obtain sufficient labels for every application. Fortunately, given a model pre-trained on a large-scale data set as ImageNet, an effective model for the target data set, which may only have hundreds of examples, can be learned by fine-tuning the pre-trained model. It is because many vision tasks are related (Zoph et al., 2020) and a model learned from ImageNet that consists of more than one million examples can contain diverse semantic information and provides a better initialization than random initialization. Figure 1 (a) illustrates the conventional fine-tuning process.

Fine-tuning from a pre-trained model becomes a prevalent strategy for handling small data sets, but its theoretical foundation is unclear. In particular, some recent studies (He et al., 2019) have shown that there is almost no benefit from training with ImageNet pre-trained models because training from scratch can achieve the same accuracy after a period of additional training. This phenomenon raises a theoretical question: when or under what kind of conditions that fine-tuning a pre-trained model is more beneficial than training from random initialization?

To this end, we aim to answer this question from the theoretical side of generalization analysis, which is commonly explored in the learning theory literature (Hardt et al., 2016). We have shown in theory that the final prediction precision has a weak dependency on the pre-trained model. Our theoretical result also tells us that when the pre-training data are too far from the target data, the domain gap will hurt the accuracy of target tasks. These two observations lead us to a second question: can we develop a new strategy of fine-tuning that achieves better generalization performance than the standard one when the target data are similar to certain examples from the pre-training data?

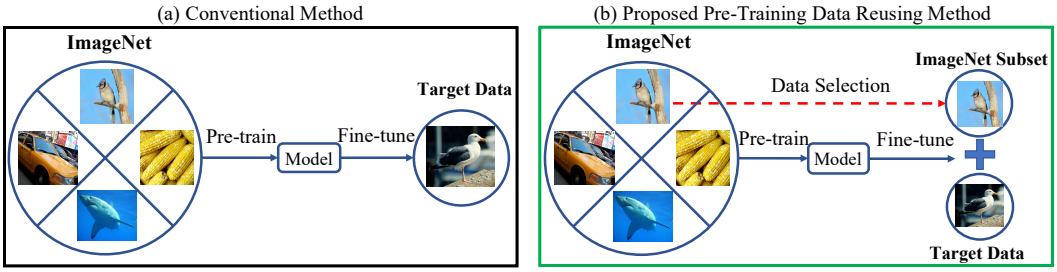

Figure 1: Comparison of fine-tuning methods. (a) is the conventional pre-training and fine-tuning method. This paper provides theoretical analysis explaining the ineffectiveness of pre-trained models compared to training from scratch. (b) shows the proposed pre-training data reusing method, which is motivated by our generalization analysis of the effect of pre-training data on fine-tuning.

This work will address this question with an affirmative answer. First, our theory indicates that the dependence of the final performance on the pre-trained model is weak. Inspired by the analysis, we propose to leverage the pre-training data, which are also available for fine-tuning, for target tasks. Concretely, we propose to reuse pre-training data and optimize its classification loss along with the target data when fine-tuning. The theoretical analysis confirms that the performance on the target data can be improved when an appropriate portion of pre-training data is selected. The proposed fine-tuning process is illustrated in Figure 1 (b). Note that including extra data for fine-tuning may increase the computational cost, but the cost is affordable when the size of examples selected from pre-training data is comparable to that of the target data.

Since target data can be from different domains, we study the reusing strategy of pre-training data for different cases. First, when the target data are closely related to the pre-training data, one can randomly sample a number of pre-training data for fine-tuning, which is referred to as **random selection**. Second, if the label information of pre-training data is available and the classes overlapped with target data are identifiable, one can directly use those data with overlapped classes in fine-tuning. For example, given the data set of CUB (Wah et al., 2011), which is a data set consisting of birds images, 59 bird classes (Qian et al., 2020) in ImageNet can be included in fine-tuning. This scheme is referred to as **label-based selection**. Finally, when the labels between pre-training and target domains cannot match exactly, the similarity measured with representations extracted from the corresponding pre-trained model will be adopted for selection. The last setting is prevalent in real-world applications and referred to as **similarity-based selection**.

Given the large scale of pre-training data, the representations from the pre-trained model can capture semantic similarity (Donahue et al., 2014). Based on this observation, we propose a novel selection algorithm to obtain a subset from pre-training data closest to the target data by solving an unbalanced optimal transport (UOT) problem. Interestingly, the proposed method performs consistently well on other scenarios, e.g., labels are overlapped, which reduces the effort of identifying overlapped pre-training classes. The main contributions of this work are summarized as follows.

- From the perspective of generalization analysis, this work explains the phenomenon that the pre-trained model has almost no benefit over training from scratch in some computer vision tasks.
- We develop the theoretical analysis when pre-training and target data are used in fine-tuning simultaneously. It demonstrates that the performance on target data can be improved when the pre-training data is similar to the target data.
- According to the analysis, we propose to select a subset of pre-training data with better similarity to the target data to further boost the final performance. A novel UOT-based algorithm is developed to handle target data from different scenarios.
- The performance of the proposed fine-tuning process is evaluated on 8 benchmark data sets. Our method surpasses the conventional fine-tuning pipeline by a large margin of 2.93% averaged over all tasks, verifying the effectiveness of reusing pre-training data.

2 Related Work

Fine-tuning as a special case of transfer learning (Pan & Yang, 2009) aims to improve the performance on the target data by transferring the knowledge from a large-scale pre-training data. For

example, supervised pre-trained models on ImageNet have been extensively used in image classification (Donahue et al., 2014), object detection (Ren et al., 2015; Lin et al., 2017) and semantic segmentation (Chen et al., 2017; Long et al., 2015). However, the empirical study in He et al. (2019) shows that the advantage of a supervised pre-trained model over random initialization cannot be observed because of the gap between pre-training and target tasks. Later, Zoph et al. (2020) demonstrates that self-supervised pre-training improves upon training from scratch in object detection and other vision tasks with strong data augmentation, indicating that self-supervised pre-training learns more general visual representations. Our work considers a general pre-training paradigm including both supervised and self-supervised approaches, and explains why pre-trained models fail to outperform random initialization from the view of generalization theory. Different from existing work that regularizes the fine-tuning optimization explicitly (Gouk et al., 2020; Aghajanyan et al., 2020), we propose to reuse pre-training data in target training based on the theoretical findings.

The most important one of the proposed data selection schemes in this work is based on a variant of optimal transport (OT) optimization. General OT is often used in computer vision to estimate or/and minimize the distance between two probability measures, such as prediction probabilities in classification (Frogner et al., 2015), density maps in crowd counting (Wan et al., 2021) and the reconstruction loss in generative models (Patrini et al., 2020; Arjovsky et al., 2017). This paper solves an unbalanced optimal transport (UOT) problem between pre-training and target data to obtain a similarity vector for pre-training data, so that we can select a portion of data close to the target task.

3 Problem Definition and Preliminary

The target problem of interest that we aim to optimize can be formulated as

$$\min_{\theta\in\mathbb{R}^d} F(\theta) := \mathrm{E}_{(x,y)\sim\mathbb{P}}\left[f(\theta;x,y)\right], \tag{1}$$

where θ is the model parameter to be learned; (x,y) is the input-label pair, which follows a unknown distribution $\mathbb{P}$; $\mathrm{E}_{(x,y)\sim\mathbb{P}}[\cdot]$ is the expectation that takes over a random variable (x,y) while we use $\mathrm{E}[\cdot]$ for the sake of simplicity when the randomness is obvious; $f(\cdot;x,y)$ is a loss function. One example of $f(\cdot;x,y)$ is cross-entropy loss for K-class classification problem which is given by $f(\theta;x,y) = \sum_{k=1}^{K} -y_k \log\left(\frac{\exp(p_k(\theta;x))}{\sum_{j=1}^{K}\exp(p_j(\theta;x))}\right)$ with prediction function $p(\theta;x)$. The problem (1) is known as population risk minimization (PRM) problem. Since the distribution $\mathbb{P}$ is unknown, the explicit formulation of (1) is difficult to be obtained. In practice, a set of training data $\{(x_1,y_1),\ldots,(x_n,y_n)\}$ drawn from $\mathbb{P}$ are given, where n is the sample size. A common approach is to solve the empirical risk minimization (ERM) problem (Vapnik, 2013):

$$\min_{\theta\in\mathbb{R}^d} F_n(\theta) := \frac{1}{n}\sum_{i=1}^{n} f(\theta;x_i,y_i). \tag{2}$$

Stochastic gradient descent (SGD) (Robbins & Monro, 1951) is a very efficient algorithm for solving problem (2) in many computer vision applications, whose updating is given by

$$\theta_{t+1} = \theta_t - \eta\nabla_\theta f(\theta_t;x_{i_t},y_{i_t}), \tag{3}$$

where $t=0,1,\ldots$ is the iteration number, $\eta>0$ is the learning rate, $\nabla_\theta f(\theta;x,y)$ is the gradient of function $f(\theta;x,y)$ with respective to variable θ. When the variable to be taken a gradient is obvious, we use ∇f for simplicity. We use the **excess risk** as the performance measurement for a solution $\widehat{\theta}$:

$$F(\widehat{\theta}) - F(\theta_*), \tag{4}$$

where $\theta_* \in \arg\min_{\theta\in\mathbb{R}^d} F(\theta)$ is the optimal solution of (1) and $\widehat{\theta}$ is the output of SGD.

In order to describe the pre-trained model, we denote by $G(\theta) := \mathrm{E}_{(x,y)\sim\mathbb{Q}}[g(\theta;x,y)]$ the objective function that pre-trained model aims to optimize. We use a parallel notation $G_m(\theta) := \frac{1}{m}\sum_{i=1}^{m} g(\theta;x_i,y_i)$, where $\{(x_1,y_1),\ldots,(x_m,y_m)\}$ is a set of training data drawn from $\mathbb{Q}$. Usually, the sample size of pre-training data is larger than that of target data, i.e., $m \gg n$. In this work, for the sake of analysis, we let m is large enough and both the pre-trained model and the target learning task share the same set of parameters. In order to ensure that the model learned by optimizing $G(\theta)$ will be valuable to the optimization of $F(\theta)$, we make the following assumption about $F(\theta)$ and $G(\theta)$.

Assumption 1. *There exists* $\Delta > 0$ *such that* $\|\nabla F(\theta) - \nabla G(\theta)\| \leq \Delta, \quad \forall \theta \in \mathbb{R}^d$.

To establish the generalization bound, we first present some assumptions for problem (1) that will be used in the analysis. Specifically, we make the following two assumptions, which are widely used in the literature (Ghadimi & Lan, 2013; Wang et al., 2019; Li et al., 2020).

Assumption 2. *The stochastic gradient of* $F(\theta)$ *is unbiased, i.e.,* $\mathrm{E}_{(x,y)}[\nabla f(\theta; x, y)] = \nabla F(\theta)$, *and the variance of stochastic gradient is bounded, i.e., there exists a constant* $\sigma^2 > 0$, *such that* $\mathrm{E}_{(x,y)}\left[\|\nabla f(\theta; x, y)) - \nabla F(\theta)\|^2\right] \leq \sigma^2$.

Assumption 3. $F(\theta)$ *is smooth with an* L*-Lipchitz continuous gradient, i.e., it is differentiable and there exists a constant* $L > 0$ *such that* $\|\nabla F(\theta_1) - \nabla F(\theta_2)\| \leq L\|\theta_1 - \theta_2\|, \forall \theta_1, \theta_2 \in \mathbb{R}^d$.

Assumption 3 says the objective function $F(\theta)$ is smooth with module parameter $L > 0$. This assumption has an equivalent expression according to (Nesterov, 2004): $F(\theta_1) - F(\theta_2) \leq \langle \nabla F(\theta_2), \theta_1 - \theta_2 \rangle + \frac{L}{2}\|\theta_1 - \theta_2\|^2, \quad \forall \theta_1, \theta_2 \in \mathbb{R}^d$. We further assume the objective function $F(\theta)$ satisfies a Polyak-Łojasiewicz (PL) condition (Polyak, 1963) with parameter $\mu > 0$.

Assumption 4. *There exists a constant* $\mu > 0$ *such that* $2\mu(F(\theta) - F(\theta_*)) \leq \|\nabla F(\theta)\|^2, \quad \forall \theta \in \mathbb{R}^d$, *where* $\theta_* \in \arg\min_{\theta \in \mathbb{R}^d} F(\theta)$ *is a optimal solution.*

This PL condition is widely used in the literature (e.g., (Wang et al., 2019; Karimi et al., 2016; Li & Li, 2018; Charles & Papailiopoulos, 2018)), and it has been observed in training deep neural networks both theoretically (Allen-Zhu et al., 2019) and empirically (Yuan et al., 2019).

4 Generalization Analysis and The Proposed Strategy

In this section, we first show the value of the pre-trained model in target tasks by examining its excess risk bound. Then we propose a new method that leverages pre-training data during the fine-tuning process. Our main goal is to get some insights of using pre-training data for the theoretical result through the generalization analysis. Finally, we describe the details of pre-training data use strategies with mainly focusing on the UOT-based method.

4.1 Value of Pre-trained Model

To see the value of a pre-trained model, we first give its excess risk bound for a target task, showing that the bound heavily depends on Δ.

Lemma 1. *Under Assumptions 1, 3, 4, suppose the function* g *satisfies the condition of unbiased and bounded stochastic gradient as described in Assumption 2, by setting the learning rate* $\eta = \min\left(1/L, \Delta^2/(2\sigma^2)\right)$, *then the pre-trained model, denoted by* θ_p, *provides the following performance guarantee for the target task* $F(\theta)$, $\mathrm{E}[F(\theta_p) - F(\theta_*)] \leq \frac{\Delta^2}{\mu}$.

As indicated in the above lemma, the performance gap between pre-trained model θ_p and the optimal model θ_* is bounded by Δ^2/μ, where both Δ describes the approximation accuracy when replacing $\nabla F(\theta)$ with $\nabla G(\theta)$ according to Assumption 1.

After completing the pre-trained model, we will run the fine-tuning process by further training the pre-trained model θ_p against the set of training examples $(x_i, y_i), i = 1, \ldots, n$ for the target task. Lemma below provides the performance guarantee in terms of excess risk bound of the final model θ_f for the target task.

Lemma 2. *Under Assumptions 2, 3, 4, suppose the learning rate* $\eta = \frac{2}{n\mu}\log\left(\frac{n\mu\Delta^2}{2L\sigma^2}\right) \leq \frac{1}{L}$, *then the final model after fine tuning* θ_p *against a set of* n *training examples, denoted by* θ_f, *provides the following performance guarantee for the target task* $F(\theta)$, $\mathrm{E}\left[F(\theta_f) - F(\theta_*)\right] \leq \frac{4L\sigma^2}{n\mu^2}\log\left(\frac{n\mu\Delta^2}{2L\sigma^2}\right)$.

Remark 1. Note that Δ only appears in the logarithmic term, implying that the final prediction precision has a weak dependency on the pre-trained model. That is to say, the pre-trained model has almost no benefit over training from scratch.

4.2 Value of Pre-training Data

Given the potential negative fact about fine-tune a pre-trained model in the last subsection, i.e. fine-tuning will not be able to improve prediction accuracy from the pre-trained model when n is too

small, a natural question is if it is possible to design a better fine-tuning process that can overcome the limitation of the existing one. To this end, we develop a better approach for fine-tuning that aims to leverage the data used for pre-trained model during the phase of fine-tuning. We note, during the phase of fine-tuning, we may have two ways of estimating the gradient $\nabla F(\theta_t)$. The first estimator is based on the samples for fine-tuning data, which provides an unbiased estimation but with a large variance. The second estimator is based on pre-training data, which provides a biased estimator with almost no variance (since the sample size of pre-training data is large enough). Our goal is to linearly combine these two estimators to provide an estimator of gradient $\nabla F(\theta_t)$ that makes the best trade-off between bias and variance. Hence, at each iteration t of fine tuning, our gradient estimator is given as

$$\nabla \widetilde{f}(\theta_t) = \alpha \nabla f(\theta_t; \zeta_{i_t}) + (1-\alpha)\nabla h(\theta_t; \xi_{i_t}) \tag{5}$$

where $\alpha \in (0, 1]$, $\zeta_{i_t} := (x_{i_t}, y_{i_t})$ is a training example from target data, $\xi_{i_t} := \{(x_{i_t}, y_{i_t}), i_t = 1, \ldots, m\}$ is a set of training examples from pre-training data, and h is a loss function (e.g., cross-entropy loss) that is related to target task. Since the sample size of pre-training data is large enough, we use mini-batching stochastic gradient $\nabla h(\theta_t; \xi_{i_t}) := \frac{1}{\widetilde{m}} \sum_{i_t=1}^{\widetilde{m}} \nabla h(\theta_t; x_{i_t}, y_{i_t})$ where $\widetilde{m}$ is the batch size. Please note that the loss function h can be same as the loss function of target task. Our solution is updated by $\theta_{t+1} = \theta_t - \eta \nabla \widetilde{f}(\theta_t)$. The theorem below provides a performance guarantee for using $\nabla \widetilde{f}(\theta_t)$ for fine-tuning.

Theorem 1. *Under Assumptions 1, 2, 3, 4, suppose the function h satisfies the condition of unbiased and bounded stochastic gradient as described in Assumption 2 and the learning rate $\eta = \frac{2}{n\mu} \log\left(\frac{n\mu\Delta^2}{2\alpha L\sigma^2}\right) \leq \frac{1}{L}$, by using the gradient estimator in (5) for updating solutions, we have*

$$\mathrm{E}\left[F(\theta_{f_*}) - F(\theta_*)\right] \leq \frac{4\alpha L\sigma^2}{n\mu^2} \log\left(\frac{n\mu\Delta^2}{2\alpha L\sigma^2}\right) + \frac{2(1-\alpha)\delta^2}{\mu}, \tag{6}$$

where $\delta^2 := \max_{\theta_t, \xi_{i_t}} \{\mathrm{E}[\|\nabla F(\theta_t) - \nabla h(\theta_t; \xi_{i_t})\|^2]\}$ and θ_{f_*} is the final model for the target task.

Remark 2. As indicated by Theorem 1, the upper bound depends on the coefficient α and the term δ^2 which measures the difference between the gradient on target data and the stochastic gradient on selected pre-training data during fine-tuning process. When $\alpha = 1$, that is, when we ignore the pre-training data, the bound is identical to that of Lemma 2. Several observations can be found from the result in Theorem 1. When δ^2 is small, by choosing appropriate $\alpha \in (0, 1]$, we may be able to further reduce the error from $F(\theta_f)$. For example, if the target and pre-training data are from the same distribution, i.e., $\mathbb{P} = \mathbb{Q}$, let h be the same loss function as f, then the term δ^2 can be arbitrary small such that $\delta^2 \leq \frac{\sigma^2}{\widetilde{m}}$ since the batching size $\widetilde{m}$ of pre-training data is large enough. When δ^2 is large, that is, when the second term of upper bound in (6) dominates the total error, then it would be worse than the result of standard fine-tuning a pre-trained model in Lemma 2. That is to say, when the pre-training data used in the fine-tuning process is too far from the target data, the mixed use strategy will lead to performance degradation. These theoretical observations inspire us to design a selection strategy for pre-training data, that is, to select images "similar" to those of target data from pre-training data and use these selected images during fine-tuning. A detailed description of the data selection strategy is introduced in the next subsection.

4.3 The Data Selection Strategy

Theorem 1 shows that the benefit of pre-trained model can be enhanced when pre-training data are used during the fine-tuning process. This inspires us to propose data selection strategies and to choose an appropriate portion of pre-training data. In experiments, we follow the standard pre-training practice in computer vision to use a deep neural network pre-trained on ImageNet, and then select data from ImageNet to help fine-tuning on target classification tasks. We summarize the proposed pre-training data reuse strategies as follows.

- When the label information of pre-training data is available, and the overlapping classes with target data are recognizable, one can only simply select the overlapping classes and use them during the fine-tuning (i.e., the use of label-specific pre-training data).
- When the difference between pre-training and target data sets is small (so δ^2 is small), a simple scheme is to uniformly sample pre-training data to use them during fine-tuning (i.e., the use of general pre-training data).

- In general cases, the third proposed selection scheme is to select similar data as target data from pre-training data (i.e., the use of similarity-specific pre-training data). For the sake of simplicity, we only describe the details of UOT-based data selection.

Label-based Selection The first scheme is to select images with classes seen in target tasks. For instance, the bird images from ImageNet are all selected when fine-tuning CUB. Unfortunately, this scheme heavily depends on the label match between pre-training and target data, which may worsen the performance in some real-world applications without perfectly matched classes.

Random Selection The second data selection scheme is to choose classes with uniform sampling, referred to as random selection. This strategy can improve the performance of target tasks if the domain gap δ^2 between pre-training and target data is small, and keep the weights close to initialization if the selected data are sufficiently large. The drawback of uniform selection is that the domain gap δ^2 is not considered in the data reusing process, so the performance heavily depends on the inherent property of data sets.

Similarity-based Selection To reduce the domain gap, we propose the third data selection scheme, an UOT-based method, to choose data classes from the pre-training set whose distributional distance to the target data set is small. To exploit the representation ability of pre-trained models, each class is represented as the mean of deep features, e.g. 512-dim features from the penultimate layer of a pre-trained ResNet18 model. Since the training set often has balanced classes, all classes are assigned with unit weights for both pre-training and target set. So we have two probability measures for the target set and pre-training set, i.e. $\{(\mathbf{a}_i, w_i^{(f)} = 1)\}_{i=1}^{K_f}$ and $\{(\mathbf{b}_j, w_j^{(g)} = 1)\}_{j=1}^{K_g}$, respectively. Denote the features of target and pre-training data as $\mathbf{v}_i^{(f)}$ and $\mathbf{v}_j^{(g)}$, $\mathbf{a}_i = \sum_{y_s=i} \mathbf{v}_s^{(f)}/n_i^{(f)}$ and $\mathbf{b}_j = \sum_{y_t=j} \mathbf{v}_t^{(g)}/m_j^{(g)}$ where $n_i^{(f)}$ is the number of images in i-th class of target data and $m_j^{(g)}$ is defined similarly for pre-training data. In the general case where $K_f \neq K_g$, the two measures have different total masses so we propose to compute the unbalanced OT distance between the two by a generalized Sinkhorn iteration (Peyré et al., 2019). Specifically, the optimization objective is formulated as a UOT problem,

$$\min_{\mathbf{P}} < \mathbf{P}, \mathbf{C} > -\epsilon h(\mathbf{P}) + \tau_1 KL(\mathbf{P}\mathbf{1}, \mathbf{w}^{(g)}) + \tau_2 KL(\mathbf{P}^T\mathbf{1}, \mathbf{w}^{(f)}), \mathbf{P} \in \mathbb{R}_+^{K_g \times K_f}, \tag{7}$$

where $\mathbf{C}_{i,j}$ is the distance between $\mathbf{a}_i$ and $\mathbf{b}_j$; $\mathbf{P}$ is the transportation matrix solved by the generalized Sinkhorn iteration; τ_1 and τ_2 determine the constraint on the reconstruction loss of pre-training and target density measures; $KL(\cdot,\cdot)$ and $h(\cdot)$ are Kullback-Leibler divergence and entropy function. Note that as a result of unbalanced total masses, we cannot perfectly reconstruct pre-training and target measures at the same time. Using this property, we can create a similarity ranking effect in the $\mathbf{P}\mathbf{1}$ vector by using a large value for τ_2 but a small value for τ_1. $\mathbf{P}\mathbf{1}$ is the density measure of pre-training data and $\mathbf{P}^T\mathbf{1}$ is the measure for target data. Since we want all classes of the target data to be covered, a large τ_2 is needed; while we need to select a subset of classes, τ_1 should be small to relax the constraint. Thus, a large $[\mathbf{P}\mathbf{1}]_j$ indicates a high similarity of class-j of pre-training data to the target data. Finally by ranking the elements in $\mathbf{P}\mathbf{1}$ and selecting top-K classes, we obtain the selected classes for a target data set.

Before ending this subsection, we demonstrate how the gradient combination (5) is computed in the experiment of this work. In the context of deep neural networks, we add two classification heads on top of the network backbone. One classification head has K_f-dim output to predict the target data and the other has K_g-dim output to predict the pre-training data. The cross entropy loss is computed for both heads and a weighted sum of losses are backpropogated to update the network parameters during fine-tuning, i.e. $\nabla \tilde{f}(\theta_t) = \frac{1}{\tilde{n}} \sum_{i=1}^{\tilde{n}} \nabla f(\theta_t; \zeta_i) + \lambda \frac{1}{\tilde{m}} \sum_{j=1}^{\tilde{m}} \nabla h(\theta_t; \xi_j)$, where $\tilde{n}$ and $\tilde{m}$ are mini batch size of target and pre-training data, λ is the weight for pre-training classification loss, which controls the weight α in (5), and both f and h are cross-entropy losses.

5 Experiments

This section presents the empirical analysis of the pre-training data reusing in image classification tasks. The experiment uses both supervised and unsupervised pre-trained models to fine-tune a variety of image classification data sets. First, data reusing fine-tuning schemes consistently improves the performance of vanilla fine-tuning, which corroborates our theoretical result. Second, the

(a) Supervised Pre-Training Model

	Method	Dogs	Cars	CUB	Pets	SUN	Aircraft	DTD	Caltech	Avg.
Baseline	Fine-Tune	82.65	85.87	75.49	91.40	58.03	77.62	70.64	90.11	78.98
Data Selection	Random	83.29	86.52	75.54	91.58	58.18	78.10	70.69	90.64	79.32
	Greedy-OT	84.63	86.79	76.92	91.66	58.70	78.43	70.90	90.67	79.84
	UOT	**84.67**	**87.03**	**77.21**	**91.98**	**59.06**	**78.94**	**71.17**	**91.11**	**80.15**

(b) Self-Supervised Pre-Training Model

	Method	Dogs	Cars	CUB	Pets	SUN	Aircraft	DTD	Caltech	Avg.
Baseline	Fine-Tune	78.64	**91.05**	77.44	90.44	61.12	87.25	75.80	92.82	81.82
Data Selection	Random	79.87	90.85	78.82	91.48	62.42	88.60	77.34	93.26	82.83
	Greedy-OT	79.43	90.89	78.63	91.27	62.27	**89.40**	76.81	93.36	82.76
	UOT	**88.14**	90.89	**80.98**	**93.05**	**64.76**	89.28	**77.45**	**93.45**	**84.75**

Table 1: Comparison of testing top-1 accuracy (%) on different data sets by fine-tuning the supervised and self-supervised pre-trained model. The proposed data selection fine-tuning consistently improves the vanilla fine-tuning, with UOT being the best method.

comparison between different data selection strategies demonstrates that the UOT selection is advantageous over random and greedy selection. Third, we simulate the situations where the training data are scarce by sub-sampling the given training data and show that as the training data get insufficient, the performance gain of the pre-training data reusing method will increase. Finally, some ablation studies on the number of selected classes and other settings in UOT are given.

5.1 Experiment Setup

The empirical study is done on both supervised and self-supervised pre-trained models. For the supervised training, we use the official ResNet18 (He et al., 2016) pre-trained on ImageNet. For the self-supervised training, we use the official MoCo-v2 (He et al., 2020) ResNet50 pre-trained with 800 epochs. Images are represented in the supervised pre-trained ResNet18 by 512-dim features from the penultimate layer while in MoCo-v2 ResNet50 by 128-dim features from the final FC layer. The pre-trained model is fine-tuned on 8 target image classification data sets, i.e. Stanford dogs (Dogs) (Khosla et al., 2011), Stanford cars (Cars) (Krause et al., 2013), Caltech-UCSD birds (CUB) (Wah et al., 2011), Oxford-IIIT Pet (Pets) (Parkhi et al., 2012), SUN (Xiao et al., 2010), FGVC-Aircraft (Aircraft) (Maji et al., 2013), Describable Textures Dataset (DTD) (Cimpoi et al., 2014) and Caltech101 (Caltech) (Fei-Fei et al., 2004). During the fine-tuning process, both the backbone and random initialized classification heads are updated using SGD with Nesterov Momentum. The training epochs are fixed to be 100 in our experiment for a sufficient training while other hyper-parameters like learning rate, weight decay and λ are determined by grid search for all methods in the comparison (details in the appendix).

We test 3 data selection methods, i.e. random selection, greedy selection and UOT selection, and set the number of selected classes to be 100 unless mentioned otherwise. Specifically, we use OT-based greedy algorithm (Cui et al., 2018) for comparison. The batch size for fine-tuning data is 256, and if pre-training data are reused, the batch size keeps the same as target data which makes a total batch size of 512. In random selection, we use the uniform selection over classes to be consistent with the other data selection methods. In Greedy-OT, we use the same setting as in the original paper where $\mathbf{C}_{ij}$ is the l_2 distance. In UOT, we set $\epsilon = 1.0$, $\tau_1 = 1.0$ and $\tau_2 = 100.0$. The distance cost is based on the cosine similarity $\mathbf{C}_{ij} = \frac{-\cos(\mathbf{a}_i, \mathbf{b}_j)+1}{\epsilon_c}$ with $\epsilon_c = 0.01$.

5.2 Comparison of Data Selection Strategies

Table 1 shows the comparison between the standard fine-tuning and 3 data reuse methods on 8 image classification data sets, with supervised and self-supervised pre-trained models. To make a fair comparison, all data reuse experiments select 100 classes of ImageNet data. The first observation is that, since the pre-training data are large enough to have similar images to target ones, even random selection achieves better performance than the standard fine-tuning in most data sets. Secondly, the benefit of data reuse is amplified by the similarity-based data selections, as predicted by Theorem 1. Finally, the comparison between Greedy-OT and UOT data selections demonstrates the advantage of the global UOT in terms of the similarity measure. Fig. 3(a) shows the performance of label-based selection on CUB (blue line), since the birds classes happen to exist in ImageNet. It turns out the accuracy of UOT selection method (77.21%) is comparable to the label-based selection's (76.89%). In addition, we also test the performance of label-based selection on Dogs (selecting 118 dog classes

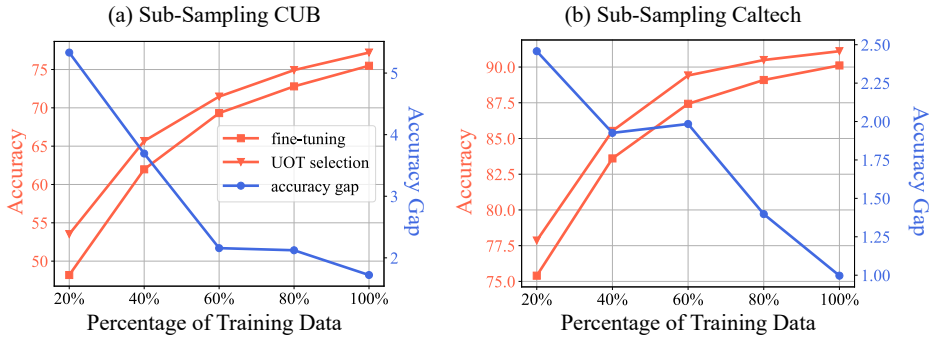

Figure 2: Accuracy and performance gap when sub-sampling training datausing the supervised pre-trained ResNet18. (a) and (b) show a decreasing trend of performance gain when more training data are added. The advantage of pre-training data reusing is larger when training data are not sufficient.

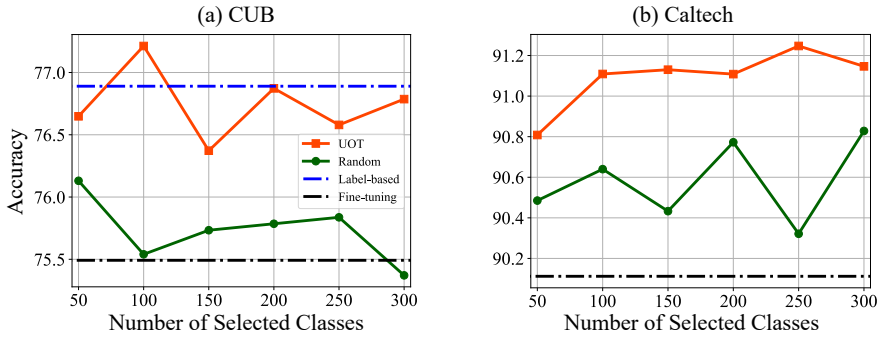

Figure 3: Accuracy of fine-tuning using UOT data selection with different numbers of selected classes using the supervised pre-trained ResNet18. (a) shows the performance on CUB and the blue line is fine-tuning with all birds classes from ImageNet. The UOT selection achieves a comparable performance to the label-based data selection. (b) shows the increased performance of UOT selection on Caltech as more data are reused in UOT, while the performance of random selection is consistently worse than the UOT's .

of ImageNet), the performance of which (85.05%) is again comparable to UOT's (84.67%). This comparison demonstrates that the proposed UOT selection is generic yet effective.

Another interesting finding is that the advantage of UOT is more evident in the self-supervised pre-training case than in the supervised pre-training one. The most considerable improvement is achieved in Dogs, Birds and Pets data sets because the animal-related classes are dominant in ImageNet (398 classes of birds, dogs, animals and mammals) and self-supervised training learns good visual features without label information. Once the label information is added to the fine-tuning process by data reuse, the model is taught to recognize those familiar features and achieves giant improvements. Note that the only data on which data reuse does not help is the Cars, indicating that the gap between ImageNet and Cars data is large when measured by the self-supervised model.

5.3 Simulation of Low-Data Regime

The generalization analysis in Lemma 2 indicates that when the target data size is not large enough, the pre-training model can outperform training from scratch. More importantly, Theorem 1 shows that when λ is properly tuned and the domain gap between reused pre-training data and target data is small, the benefit of the pre-trained model will be strengthened. To study the effect of data reusing in the scarce data scenario, we simulate low-data target tasks in this experiment by sub-sampling CUB and Caltech training data. The reason why we select the two data sets is that they represent the fine-grained and general classification task, respectively. In sub-sampling, for each class of training data we randomly sample 20%, 40%, 60% and 80% of images to get class-balanced training data.

Figure 2 shows that accuracy and performance gap between vanilla and data-reusing fine-tuning when different amounts of training data are available. On both data sets, the performance gap is increased as the training data get less, indicating that the UOT-selection data reusing scheme helps more when the target data are insufficient. This experiment demonstrates that the proposed data reusing paradigm is particularly effective when the target task does not have enough data, which could be a typical case in real-world applications.

(a) Ablation of distance function

Method	Greedy-OT-l_2	Greedy-OT-cos	UOT-l_2	UOT-cos
Supervised	86.44%	92.92%	94.92%	**98.31%**
Self-supervised	16.95%	38.98%	16.95%	**93.22%**

(b) Sensitivity of ϵ_c in UOT-cos

Method	$\epsilon_c = 1.0$	$\epsilon_c = 0.3$	$\epsilon_c = 0.1$	$\epsilon_c = 0.03$	$\epsilon_c = 0.01$	$\epsilon_c = 0.003$
Supervised	94.92%	94.92%	94.92%	**98.31%**	**98.31%**	96.61
Self-supervised	50.85%	69.49%	89.83%	**94.92%**	93.22%	93.22%

Table 2: Comparison of data selection methods on CUB. (a) The UOT-cos selection is better than Greedy-OT selection in terms of bird classes recall rate. (b) On the supervised pre-trained model, the recall rate is not sensitive to ϵ_c; on the self-supervised model, when ϵ_c is small, its sensitivity is small.

5.4 Ablation Study

Number of selected pre-training classes. We investigate the effect of selected class number on the target classification accuracy. Figure 3 shows the performance of target tasks (CUB and Caltech) when the number of selected classes ranges from 50 to 300 in UOT selection. The increased pre-training data added in fine-tuning do not improve the performance of CUB, since there are 59 classes of birds in the ImageNet and more reused images enlarge the gap δ^2. Surprisingly, we observe that only using the birds images (blue line) is not the best strategy on CUB. It is because that there can be a certain number of related classes in ImageNet, which will help the prediction on birds images. The result shows that even when labels of pre-training and target data are given and overlapped, UOT selection can achieve a better performance by including extra relevant classes from pre-training data. On the general classification data set (Caltech), more reused pre-training data help gain the performance improvement because the diverse data set needs a large number of images to have a small domain gap. On both data sets, the UOT selection performs better than the random selection as the number of selected classes changes.

Distance function and ϵ_c. To investigate the influence of different factors in the UOT selection, we define a recall rate as a metric to make the comparison. For a target data set whose classes happen to exist in ImageNet, the similarity-based data selection is expected to choose those matched classes. For example, select all 59 birds classes from ImageNet when fine-tuning on CUB. Thus, the recall rate on CUB is defined as the ratio between the number of birds classes in top-100 similar vector or EMD distance and 59. With the performance metric, we first compare UOT with Greedy-OT under l_2 and cosine distance in Table 2(a). With a supervised pre-trained model, Greedy-OT is only slightly worse than UOT, while with a self-supervised model the weakness of Greedy-OT is amplified. It means that Greedy-OT heavily relies on the label information in supervised training but UOT only needs generic visual features to have a good similarity measure. In addition, the cosine distance is better than the l_2 distance, especially in the self-supervised model. The importance of cosine distance is due to the cosine similarity loss used in MoCo training. Finally, Table 2(b) shows the recall rate when using different ϵ_c. The supervised model is not sensitive to the choice of ϵ_c but a small ϵ_c is crucial to the good performance of OT-selection in the self-supervised model. Note that the recall rate of Greedy-OT does not depend on ϵ_c so the performance is worse than UOT no matter what ϵ_c is used.

6 Conclusion

This paper theoretically investigates the generalization problem of pre-trained models when fine-tuning on target tasks. Our theory illustrates that the pre-trained model can have little positive influence on learning from target data under certain conditions. Therefore, we consider to include pre-training data directly for better fine-tuning. The theoretical analysis confirms that the performance on the target data can be improved when similar data are selected from the pre-training data for fine-tuning. According to this result, a novel similarity-based selection algorithm is developed. Empirical studies on diverse data sets demonstrate the effectiveness of the proposed fine-tuning process. Our future work will focus on the self-supervised pre-trained case in which class information is not given, and investigate label-free data selection methods to boost the performance of self-supervised pre-trained models in target tasks.

REFERENCES

Armen Aghajanyan, Akshat Shrivastava, Anchit Gupta, Naman Goyal, Luke Zettlemoyer, and Sonal Gupta. Better fine-tuning by reducing representational collapse. In *International Conference on Learning Representations*, 2020.

Zeyuan Allen-Zhu, Yuanzhi Li, and Zhao Song. A convergence theory for deep learning via over-parameterization. In *International Conference on Machine Learning*, pp. 242–252, 2019.

Martin Arjovsky, Soumith Chintala, and Léon Bottou. Wasserstein generative adversarial networks. In *International conference on machine learning*, pp. 214–223. PMLR, 2017.

Zachary Charles and Dimitris Papailiopoulos. Stability and generalization of learning algorithms that converge to global optima. In *International Conference on Machine Learning*, pp. 745–754, 2018.

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.

Mircea Cimpoi, Subhransu Maji, Iasonas Kokkinos, Sammy Mohamed, and Andrea Vedaldi. Describing textures in the wild. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3606–3613, 2014.

Yin Cui, Yang Song, Chen Sun, Andrew Howard, and Serge Belongie. Large scale fine-grained categorization and domain-specific transfer learning. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 4109–4118, 2018.

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*, pp. 248–255. Ieee, 2009.

Jeff Donahue, Yangqing Jia, Oriol Vinyals, Judy Hoffman, Ning Zhang, Eric Tzeng, and Trevor Darrell. Decaf: A deep convolutional activation feature for generic visual recognition. In *International conference on machine learning*, pp. 647–655. PMLR, 2014.

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*, pp. 178–178. IEEE, 2004.

Charlie Frogner, Chiyuan Zhang, Hossein Mobahi, Mauricio Araya-Polo, and Tomaso Poggio. Learning with a wasserstein loss. *arXiv preprint arXiv:1506.05439*, 2015.

Saeed Ghadimi and Guanghui Lan. Stochastic first-and zeroth-order methods for nonconvex stochastic programming. *SIAM Journal on Optimization*, 23(4):2341–2368, 2013.

Henry Gouk, Timothy Hospedales, et al. Distance-based regularisation of deep networks for fine-tuning. In *International Conference on Learning Representations*, 2020.

Moritz Hardt, Ben Recht, and Yoram Singer. Train faster, generalize better: Stability of stochastic gradient descent. In *International Conference on Machine Learning*, pp. 1225–1234, 2016.

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*, pp. 770–778, 2016.

Kaiming He, Ross Girshick, and Piotr Dollár. Rethinking imagenet pre-training. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 4918–4927, 2019.

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*, pp. 9729–9738, 2020.

Hamed Karimi, Julie Nutini, and Mark Schmidt. Linear convergence of gradient and proximal-gradient methods under the polyak-łojasiewicz condition. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pp. 795–811. Springer, 2016.

Aditya Khosla, Nityananda Jayadevaprakash, Bangpeng Yao, and Fei-Fei Li. Novel dataset for fine-grained image categorization: Stanford dogs. In *Proc. CVPR Workshop on Fine-Grained Visual Categorization (FGVC)*, volume 2, 2011.

Jonathan Krause, Michael Stark, Jia Deng, and Li Fei-Fei. 3d object representations for fine-grained categorization. In *Proceedings of the IEEE international conference on computer vision workshops*, pp. 554–561, 2013.

Xiaoyu Li, Zhenxun Zhuang, and Francesco Orabona. Exponential step sizes for non-convex optimization. *arXiv preprint arXiv:2002.05273*, 2020.

Zhize Li and Jian Li. A simple proximal stochastic gradient method for nonsmooth nonconvex optimization. In *Advances in Neural Information Processing Systems*, pp. 5564–5574, 2018.

Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *Proceedings of the IEEE international conference on computer vision*, pp. 2980–2988, 2017.

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*, pp. 3431–3440, 2015.

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

Yurii Nesterov. *Introductory lectures on convex optimization : a basic course*. Applied optimization. Kluwer Academic Publ., 2004. ISBN 1-4020-7553-7.

Sinno Jialin Pan and Qiang Yang. A survey on transfer learning. *IEEE Transactions on knowledge and data engineering*, 22(10):1345–1359, 2009.

Omkar M Parkhi, Andrea Vedaldi, Andrew Zisserman, and CV Jawahar. Cats and dogs. In *2012 IEEE conference on computer vision and pattern recognition*, pp. 3498–3505. IEEE, 2012.

Giorgio Patrini, Rianne van den Berg, Patrick Forre, Marcello Carioni, Samarth Bhargav, Max Welling, Tim Genewein, and Frank Nielsen. Sinkhorn autoencoders. In *Uncertainty in Artificial Intelligence*, pp. 733–743. PMLR, 2020.

Gabriel Peyré, Marco Cuturi, et al. Computational optimal transport: With applications to data science. *Foundations and Trends® in Machine Learning*, 11(5-6):355–607, 2019.

Boris Teodorovich Polyak. Gradient methods for minimizing functionals. *Zhurnal Vychislitel'noi Matematiki i Matematicheskoi Fiziki*, 3(4):643–653, 1963.

Qi Qian, Juhua Hu, and Hao Li. Hierarchically robust representation learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 7334–7342, 2020.

Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. *Advances in neural information processing systems*, 28: 91–99, 2015.

Herbert Robbins and Sutton Monro. A stochastic approximation method. *The annals of mathematical statistics*, pp. 400–407, 1951.

Vladimir Vapnik. *The nature of statistical learning theory*. Springer science & business media, 2013.

Catherine Wah, Steve Branson, Peter Welinder, Pietro Perona, and Serge Belongie. The caltech-ucsd birds-200-2011 dataset. 2011.

Jia Wan, Ziquan Liu, and Antoni B Chan. A generalized loss function for crowd counting and localization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 1974–1983, 2021.

Zhe Wang, Kaiyi Ji, Yi Zhou, Yingbin Liang, and Vahid Tarokh. Spiderboost and momentum: Faster variance reduction algorithms. In *Advances in Neural Information Processing Systems*, pp. 2403–2413, 2019.

Jianxiong Xiao, James Hays, Krista A Ehinger, Aude Oliva, and Antonio Torralba. Sun database: Large-scale scene recognition from abbey to zoo. In *2010 IEEE computer society conference on computer vision and pattern recognition*, pp. 3485–3492. IEEE, 2010.

Zhuoning Yuan, Yan Yan, Rong Jin, and Tianbao Yang. Stagewise training accelerates convergence of testing error over sgd. In *Advances in Neural Information Processing Systems*, pp. 2604–2614, 2019.

Barret Zoph, Golnaz Ghiasi, Tsung-Yi Lin, Yin Cui, Hanxiao Liu, Ekin Dogus Cubuk, and Quoc Le. Rethinking pre-training and self-training. *Advances in Neural Information Processing Systems*, 2020.

A PROOFS

A.1 PROOF OF LEMMA 1

Proof. For the sake of simplicity, let the training examples $\xi_t := (x_{i_t}, y_{i_t}), i_t = 1, \ldots, m$ are sampled from $\mathbb{Q}$. By the smoothness of function F from Assumption 3, we have

$$\begin{aligned}
&\mathrm{E}[F(\theta_{t+1}) - F(\theta_t)] \\
\leq &\mathrm{E}[\langle \theta_{t+1} - \theta_t, \nabla F(\theta_t)\rangle] + \frac{L}{2}\mathrm{E}[\|\theta_{t+1} - \theta_t\|^2] \\
= &-\eta \mathrm{E}[\langle \nabla g(\theta_t; \xi_t), \nabla F(\theta_t)\rangle] + \frac{\eta^2 L}{2}\mathrm{E}[\|\nabla g(\theta_t; \xi_t)\|^2] \\
= &\frac{\eta}{2}\|\nabla F(\theta_t) - \nabla G(\theta_t)\|^2 - \frac{\eta}{2}\|\nabla F(\theta_t)\|^2 - \frac{\eta(1-\eta L)}{2}\mathrm{E}[\|\nabla G(\theta_t)\|^2] \\
&+ \frac{\eta^2 L}{2}\mathrm{E}[\|\nabla g(\theta_t; \xi_t) - \nabla G(\theta_t)\|^2]. \qquad (8)
\end{aligned}$$

where the last inequality uses $\mathrm{E}[\nabla g(\theta_t; \xi_t)] = \nabla g(\theta_t)$. Due to Assumptions 1, the condition of unbiased and bounded stochastic gradient for pre-trained objective function $G(\theta)$ and $\eta \leq 1/L$ we have

$$\mathrm{E}[F(\theta_{t+1}) - F(\theta_t)] \leq \frac{\eta\Delta^2}{2} + \frac{\eta^2\sigma^2}{2} - \frac{\eta}{2}\|\nabla F(\theta_t)\|^2 \qquad (9)$$

Since $F(\cdot)$ is a μ-PL function under Assumption 4, we have

$$\mathrm{E}[F(\theta_{t+1}) - F(\theta_t)] \leq -\frac{\eta\mu}{2}\mathrm{E}[(F(\theta_t) - F(\theta_*))] + \frac{\eta\Delta^2}{2} + \frac{\eta^2\sigma^2}{2} \qquad (10)$$

and thus

$$\mathrm{E}[F(\theta_{T+1}) - F(\theta_*)] \leq \exp\left(-\frac{\eta\mu T}{2}\right)(F(\theta_1) - F(\theta_*)) + \frac{\Delta^2}{2\mu} + \frac{\eta\sigma^2}{2\mu}, \qquad (11)$$

where $\theta_* \in \arg\min_{\theta\in\mathbb{R}^d} F(\theta)$. By selecting that η is small such that $\eta \leq \frac{\Delta^2}{2\sigma^2}$ and selecting that T is sufficiently large, i.e. $\exp\left(-\frac{\eta\mu T}{2}\right)(F(\theta_1) - F(\theta_*)) \leq \frac{\Delta^2}{4\mu}$, we have the following guarantee for the pre-trained model θ_p

$$\mathrm{E}[F(\theta_p) - F(\theta_*)] \leq \frac{\Delta^2}{\mu}.$$

□

A.2 PROOF OF LEMMA 2

Proof. For the sake of simplicity, let the training examples $\zeta_t := (x_{i_t}, y_{i_t}), i_t = 1, \ldots, n$ are sampled from $\mathbb{P}$. By the smoothness of function F from Assumption 3, we have

$$\begin{aligned}
&\mathrm{E}[F(\theta_{t+1}) - F(\theta_t)] \\
\leq &\mathrm{E}[\langle \theta_{t+1} - \theta_t, \nabla F(\theta_t)\rangle] + \frac{L}{2}\mathrm{E}[\|\theta_{t+1} - \theta_t\|^2] \\
= &-\eta \mathrm{E}[\langle \nabla f(\theta_t; \zeta_t), \nabla F(\theta_t)\rangle] + \frac{\eta^2 L}{2}\mathrm{E}[\|\nabla f(\theta_t; \zeta_t)\|^2] \\
= &-\eta\left(1 - \frac{\eta L}{2}\right)\|\nabla F(\theta_t)\|^2 + \frac{\eta^2 L}{2}\mathrm{E}[\|\nabla f(\theta_t; \zeta_t) - \nabla F(\theta_t)\|^2] \\
\leq &-\frac{\eta}{2}\|\nabla F(\theta_t)\|^2 + \frac{\eta^2 L\sigma^2}{2}, \qquad (12)
\end{aligned}$$

where the second equality is due to $\mathrm{E}[\nabla f(\theta; \zeta)] = \nabla F(\theta)$ in Assumption 2 and the last inequality uses the fact that $\eta \leq 1/L$ and Assumption 2. Following the similar analysis as that for Lemma 1 with $\theta_1 = \theta_p$, for t, by using the condition that $F(\cdot)$ is a μ-PL function in Assumption 4 we have

$$\mathrm{E}\left[F(\theta_{t+1}) - F(\theta_*)\right] \leq \left(1 - \frac{\eta\mu}{2}\right)\mathrm{E}\left[F(\theta_t) - F(\theta_*)\right] + \frac{\eta^2 L\sigma^2}{2}$$

and therefore

$$\mathrm{E}\left[F(\theta_{n+1}) - F(\theta_*)\right] \leq \exp\left(-\frac{\eta\mu n}{2}\right)\mathrm{E}[F(\theta_p) - F(\theta_*)] + \frac{\eta L\sigma^2}{\mu}$$

We complete the proof by plugging the bound for $F(\theta_p)$ from Lemma 1, i.e.,

$$\mathrm{E}\left[F(\theta_f) - F(\theta_*)\right] \leq \exp\left(-\frac{\eta\mu n}{2}\right)\frac{\Delta^2}{\mu} + \frac{\eta L\sigma^2}{\mu} \tag{13}$$

By setting $\eta = \frac{2}{n\mu}\log\left(\frac{n\mu\Delta^2}{2L\sigma^2}\right)$, we will have the following bound

$$\mathrm{E}\left[F(\theta_f) - F(\theta_*)\right] \leq \frac{4L\sigma^2}{n\mu^2}\log\left(\frac{n\mu\Delta^2}{2L\sigma^2}\right). \tag{14}$$

□

A.3 Proof of Theorem 1

Proof. Let $\nabla H(\theta) := \mathrm{E}_\xi[\nabla h(\theta;\xi)]$. By the smoothness of function F from Assumption 3, following the standard analysis, we have

$$\begin{aligned}
&\mathrm{E}\left[F(\theta_{t+1}) - F(\theta_t)\right] \\
\leq&\mathrm{E}\left[\langle\nabla F(\theta_t), \theta_{t+1} - \theta_t\rangle\right] + \frac{L}{2}\mathrm{E}\left[\|\theta_{t+1} - \theta_t\|^2\right] \\
=&-\eta\mathrm{E}\left[\langle\nabla F(\theta_t), \alpha\nabla f(\theta_t;\zeta_{i_t}) + (1-\alpha)\nabla h(\theta_t;\xi_{i_t})\rangle\right] + \frac{\eta^2 L}{2}\mathrm{E}\left[\|\alpha\nabla f(\theta_t;\zeta_{i_t}) + (1-\alpha)\nabla h(\theta_t;\xi_{i_t})\|^2\right] \\
\leq&-\eta\mathrm{E}\left[\langle\nabla F(\theta_t), \alpha\nabla F(\theta_t) + (1-\alpha)\nabla gh(\theta_t;\xi_{i_t})\rangle\right] + \frac{\eta^2 L}{2}\mathrm{E}\left[(1-\alpha)\|\nabla H(\theta_t)\|^2 + \alpha\|\nabla F(\theta_t)\|^2\right] \\
&+\frac{\alpha\sigma^2\eta^2 L}{2} + \frac{(1-\alpha)\sigma^2\eta^2 L}{2\widetilde{m}} \\
=&\frac{\eta(1-\alpha)}{2}\mathrm{E}\left[\|\nabla F(\theta_t) - \nabla h(\theta_t;\xi_{i_t})\|^2\right] + \left(-\eta\alpha + \frac{\eta^2 L\alpha - \eta(1-\alpha)}{2}\right)\mathrm{E}[\|\nabla F(\theta_t)\|^2 \\
&+\left(\frac{\eta(1-\alpha)(\eta L - 1)}{2}\right)\mathrm{E}[\|\nabla H(\theta_t)\|^2] + \frac{\alpha\sigma^2\eta^2 L}{2} + \frac{(1-\alpha)\sigma^2\eta^2 L}{2\widetilde{m}} \\
\leq&\frac{\eta(1-\alpha)\delta^2}{2} - \frac{\eta(1+\alpha-\eta L\alpha)}{2}\mathrm{E}[\|\nabla F(\theta_t)\|^2] + \frac{\alpha\sigma^2\eta^2 L}{2} + \frac{(1-\alpha)\sigma^2\eta(\eta L + 1)}{2\widetilde{m}} \\
\leq&\frac{\eta(1-\alpha)\delta^2}{2} - \frac{\eta}{2}\mathrm{E}[\|\nabla F(\theta_t)\|^2] + \frac{\alpha\sigma^2\eta^2 L}{2} + \frac{(1-\alpha)\sigma^2\eta}{\widetilde{m}}
\end{aligned}$$

where $\delta^2 := \max_{\theta_t,\xi_{i_t}}\{\mathrm{E}[\|\nabla F(\theta_t) - \nabla h(\theta_t;\xi_{i_t})\|^2]\}$; the last inequality is due to $\eta \leq 1/L$. As a result, we have

$$\begin{aligned}
&\mathrm{E}\left[F(\theta_{n+1}) - F(\theta_*)\right] \\
\leq&\exp\left(-\frac{\eta\mu n}{2}\right)\mathrm{E}\left[F(\theta_p) - F(\theta_*)\right] + \frac{(1-\alpha)\delta^2}{\mu} + \frac{\alpha\eta L\sigma^2}{\mu} + \frac{2(1-\alpha)\sigma^2}{\widetilde{m}\mu} \\
\leq&\exp\left(-\frac{\eta\mu n}{2}\right)\frac{\Delta^2}{\mu} + +\frac{(1-\alpha)\delta^2}{\mu} + \frac{\alpha\eta L\sigma^2}{\mu} + \frac{2(1-\alpha)\sigma^2}{\widetilde{m}\mu}
\end{aligned} \tag{15}$$

By setting $\eta = \frac{2}{n\mu}\log\left(\frac{n\mu\Delta^2}{2\alpha L\sigma^2}\right)$ and $\theta_{f_*} = \theta_{n+1}$, the inequality (15) will lead to the following bound

$$\begin{aligned}
&\mathrm{E}\left[F(\theta_{f_*}) - F(\theta_*)\right] \\
\leq&\frac{4\alpha L\sigma^2}{n\mu^2}\log\left(\frac{n\mu\Delta^2}{2\alpha L\sigma^2}\right) + \frac{(1-\alpha)\delta^2}{\mu} + \frac{2(1-\alpha)\sigma^2}{\widetilde{m}\mu} \\
\leq&\frac{4\alpha L\sigma^2}{n\mu^2}\log\left(\frac{n\mu\Delta^2}{2\alpha L\sigma^2}\right) + \frac{2(1-\alpha)\delta^2}{\mu},
\end{aligned} \tag{16}$$

where the last inequality is due to $\widetilde{m}$ is large enough such that $\widetilde{m} \geq \frac{2\sigma^2}{\delta}$. □

B ADDITIONAL EXPERIMENT DETAILS

Our experiment is run on 4 Nvidia V100 GPUs using PyTorch. The hyperparameters in the fine-tuning are selected by a grid search on validation sets and the hyperparameter setting with the best accuracy is reported in Table 3 and 4. The learning rate is searched over $[0.1, 0.03, 0.01, 0.003, 0.001]$, λ is over $[1.0, 0.3, 0.1]$, weight decay is over $[10^{-4}, 10^{-5}, 0.0]$. The head and backbone learning rate ratio is searched over $[10.0, 1.0, 0.1]$ and the ratio is fixed for one data set's experiment. In Table 3 and 4, for fine-tuning, the hyperparameter vector denotes [learning rate, weight decay]; for data reusing, the hyperparameter vector denotes [learning rate, λ, weight decay].

In the training data sub-sampling experiment, the UOT data selection is done on the sub-sampled data. To keep the ratio between target data and pre-training data the same in different training data sizes, we sub-sample the images in each selected pre-training class with the same sub-sampling ratio as in target data. The pre-training data sub-sampling is done for both UOT and random selection.

Table 3: Hyperparameter settings on the supervised model.

Method	Fine-Tune	Random	Greedy-OT	UOT
Stanford Dogs	$[0.001, 10^{-5}]$	$[0.001, 1.0, 10^{-4}]$	$[0.001, 1.0, 10^{-4}]$	$[0.001, 1.0, 10^{-4}]$
Stanford Cars	$[0.1, 10^{-5}]$	$[0.1, 0.3, 10^{-5}]$	$[0.1, 0.1, 0.0]$	$[0.1, 0.3, 10^{-4}]$
CUB	$[0.003, 10^{-5}]$	$[0.003, 1.0, 10^{-5}]$	$[0.003, 1.0, 10^{-5}]$	$[0.003, 1.0, 10^{-4}]$
Pets	$[0.001, 10^{-5}]$	$[0.003, 1.0, 10^{-4}]$	$[0.003, 1.0, 10^{-5}]$	$[0.001, 1.0, 0.0]$
SUN	$[0.001, 10^{-5}]$	$[0.001, 1.0, 10^{-4}]$	$[0.001, 1.0, 10^{-5}]$	$[0.001, 1.0, 10^{-5}]$
Aircraft	$[0.03, 0.0, 1.0]$	$[0.03, 0.1, 10^{-5}]$	$[0.03, 0.1, 0.0]$	$[0.03, 0.3, 10^{-4}]$
DTD	$[0.001, 10^{-5}]$	$[0.003, 0.3, 0.0]$	$[0.003, 0, 3, 10^{-5}]$	$[0.003, 0.3, 10^{-4}]$
Caltech	$[0.003, 10^{-5}]$	$[0.003, 1.0, 0.0]$	$[0.003, 1.0, 0.0]$	$[0.01, 1.0, 10^{-4}]$

Table 4: Hyperparameter settings on the self-supervised model.

Method	Fine-Tune	Random	Greedy-OT	UOT
Stanford Dogs	$[0.003, 10^{-4}]$	$[0.003, 1.0, 10^{-4}]$	$[0.003, 1.0, 10^{-5}]$	$[0.003, 1.0, 0.0]$
Stanford Cars	$[0.01, 0]$	$[0.01, 0.1, 10^{-5}]$	$[0.01, 0.3, 0.0]$	$[]0.01, 0.3, 10^{-4}$
CUB	$[0.003, 10^{-4}]$	$[0.01, 0.3, 10^{-4}]$	$[0.01, 0.3, 10^{-5}]$	$[0.01, 1.0, 10^{-4}]$
Pets	$[0.003, 0.0]$	$[0.003, 0.3, 10^{-5}]$	$[0.003, 1.0, 0.0]$	$[0.003, 1.0, 10^{-4}]$
SUN	$[0.003, 10^{-4}]$	$[0.003, 1.0, 10^{-5}]$	$[0.003, 1.0, 10^{-4}]$	$[0.003, 1.0, 0.0]$
Aircraft	$[0.01, 10^{-5}]$	$[0.01, 0.1, 10^{-4}]$	$[0.01, 0.1, 10^{-5}]$	$[0.01, 0.1, 10^{-4}]$
DTD	$[0.001, 10^{-4}]$	$[0.001, 0.3, 0.0]$	$[0.001, 1.0, 10^{-5}]$	$[0.003, 1.0, 10^{-4}]$
Caltech	$[0.003, 10^{-5}]$	$[0.003, 1.0, 0.0]$	$[0.003, 1.0, 10^{-4}]$	$[0.003, 1.0, 0.0]$