Active Finetuning: Exploiting Annotation Budget in the Pretraining-Finetuning Paradigm

Yichen Xie[1], Han Lu[2], Junchi Yan[2], Xiaokang Yang[2], Masayoshi Tomizuka[1], Wei Zhan[1]
[1] University of California, Berkeley [2] Shanghai Jiao Tong University
{yichen_xie,tomizuka,wzhan}@berkeley.edu,{sjtu_luhan,yanjunchi,xkyang}@sjtu.edu.cn

Abstract

Given the large-scale data and the high annotation cost, pretraining-finetuning becomes a popular paradigm in multiple computer vision tasks. Previous research has covered both the unsupervised pretraining and supervised finetuning in this paradigm, while little attention is paid to exploiting the annotation budget for finetuning. To fill in this gap, we formally define this new active finetuning task focusing on the selection of samples for annotation in the pretraining-finetuning paradigm. We propose a novel method called ActiveFT for active finetuning task to select a subset of data distributing similarly with the entire unlabeled pool and maintaining enough diversity by optimizing a parametric model in the continuous space. We prove that the Earth Mover's distance between the distributions of the selected subset and the entire data pool is also reduced in this process. Extensive experiments show the leading performance and high efficiency of ActiveFT superior to baselines on both image classification and semantic segmentation. Our code is released at https://github.com/yichen928/ActiveFT.

1. Introduction

Recent success of deep learning heavily relies on abundant training data. However, the annotation of large-scale datasets often requires intensive human labor. This dilemma inspires a popular *pretraining-finetuning paradigm* where models are pretrained on a large amount of data in an unsupervised manner and finetuned on a small labeled subset.

Existing literature pays significant attention to both the unsupervised pretraining [7, 13, 14, 18] and supervised finetuning [26]. In spite of their notable contributions, these researches build upon an unrealistic assumption that *we already know which samples should be labeled*. As shown in Fig. 1, given a large unlabeled data pool, it is necessary to pick up the most useful samples to exploit the limited annotation budget. In most cases, this selected subset only counts a small portion (*e.g.* <10%) of this large unlabeled pool. Despite the long-standing under-exploration, the selection strategy is still crucial since it may significantly affect the final results.

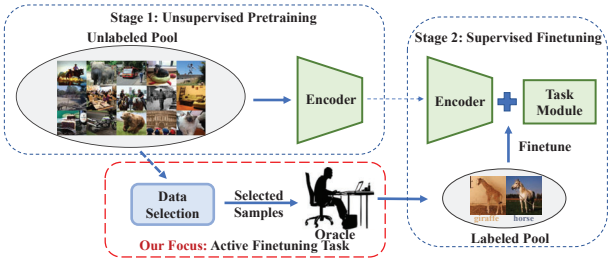


Figure 1. **Pretraining-Finetuning Paradigm:** We focus on the selection strategy of a small subset from a large unlabeled data pool for annotation, named as active finetuning task, which is under-explored for a long time.

Active learning algorithms [34, 38, 39] seem to be a potential solution, which aims to select the most suitable samples for annotation when models are trained from scratch. However, their failures in this pretraining-finetuning paradigm are revealed in both [4] and our experiments (Sec. 4.1). A possible explanation comes from the *batch-selection strategy* of most current active learning methods. Starting from a random initial set, this strategy repeats the model training and data selection processes multiple times until the annotation budget runs out. Despite their success in from-scratch training, it does not fit this pretraining-finetuning paradigm well due to the typically low annotation budget, where too few samples in each batch lead to harmful bias inside the selection process.

To fill in this gap in the pretraining-finetuning paradigm, we formulate a new task called *active finetuning*, concentrating on the sample selection for supervised finetuning. In this paper, a novel method, **ActiveFT**, is proposed to deal with this task. Starting from purely unlabeled data, ActiveFT fetches a proper data subset for supervised finetuning in a negligible time. Without any redundant heuristics, we

directly bring close the distributions between the selected subset and the entire unlabeled pool while ensuring the diversity of the selected subset. This goal is achieved by continuous optimization in the high-dimensional feature space, which is mapped with the pretrained model.

We design a parametric model p_{θ_S} to estimate the distribution of the selected subset. Its parameter θ_S is exactly the high-dimensional features of those selected samples. We optimize this model via gradient descent by minimizing our designed loss function. Unlike traditional active learning algorithms, our method can select all the samples from scratch in a single-pass without iterative batch-selections. We also mathematically show that the optimization in the continuous space can exactly reduce the earth mover's distance (EMD) [35, 36] between the entire pool and selected subset in the discrete data sample space.

Extensive experiments are conducted to evaluate our method in the pretraining-finetuning paradigm. After pretraining the model on ImageNet-1k [37], we select subsets of data from CIFAR-10, CIFAR-100 [23], and ImageNet-1k [37] for image classification, as well as ADE20k [50] for semantic segmentation. Results show the significant performance gain of our ActiveFT in comparison with baselines.

Our contributions are summarized as follows:

- To our best knowledge, we are the first to identify the gap of data selection for annotation and supervised finetuning in the pretraining-finetuning paradigm, which can cause inefficient use of annotation budgets as also verified in our empirical study. Meanwhile, we formulate a new task called *active finetuning* to fill in this gap.

- We propose a novel method, ActiveFT, to deal with the active finetuning task through parametric model optimization which theoretically reduces the earth mover's distance (EMD) between the distributions of the selected subset and entire unlabeled pool. To our best knowledge, we are the first to directly optimize samples to be selected in the continuous space for data selection tasks.

- We apply ActiveFT to popular public datasets, achieving leading performance on both classification and segmentation tasks. In particular, our ablation study results justify the design of our method to fill in the data selection gap in the pretraining-finetuning paradigm. The source code will be made public available.

2. Related Work

Unsupervised Learning aims to learn the feature representation without the participation of labels. Both contrastive and generative methods achieve great success in this field. Contrastive methods model the similarity and dissimilarity between different input samples. Some early work resorts to a large batch size [7] or memory bank [15, 46] to include enough negative samples in each iteration. Challenging the necessity of negative samples, some following study tries to train the network only with positive samples. To this end, they introduce the momentum encoder [13], clustering strategy [5], or stop-gradient operation [8] into contrastive learning frameworks. Based on the success of prior arts, [6, 9] succeed in transplanting contrastive learning to vision transformers [10]. Some recent research [2, 14, 19, 45] explores generative methods that predict the missing content inside input samples, also achieving promising performance over vision transformers.

For both kinds of methods, prior research has well investigated their positive roles in downstream supervised finetuning. Of particular interest, they can bring significant performance gain in semi-supervised learning settings, where only a small part (*e.g.* 1%) of data samples are annotated.

Active Learning selects useful samples to fill up the limited annotation budget most beneficial for model training. Despite the existence of *query-synthesizing* [29, 31, 52] and *stream-based* [11, 32] methods, current mainstream approaches are *pool-based*. Given an unlabeled data pool, the pool-based algorithms choose a part of samples for annotation. There are two different selection criteria: *uncertainty* [3, 27, 49] and *diversity* [1, 38, 39]. Based on the uncertainty inside model prediction, the algorithm can select the most difficult data samples. Early work estimates the uncertainty with various heuristics such as posterior probability [25, 44], entropy [20, 28], and classification margin [41]. Some following research directly measures the uncertainty by estimating the training loss [17, 48] or influence on model performance [12, 27] of each sample. Many other algorithms focus on the diversity of selected samples so that the distribution of this selected subset could become close to the original unlabeled pool. To be specific, Sener and Savarese [38] theoretically formulate the data selection process as a k-Center problem and proposes a CoreSet algorithm. Agarwal et al. [1] replace the Euclidean distance with context-aware KL-divergence. Sinha et al. [39] train an adversarial network to discriminate labeled and unlabeled samples. Previous work [30, 38] also tries to formulate active learning as an optimization problem. They typically pay attention to the discrete space, since it trivially matches the sample distribution inside a dataset. However, discrete optimization problem tends to be much more difficult to solve than continuous problems. Some recent efforts also pay attention to the combination between active learning and unsupervised learning. For example, Yi et al. [47] guides the data selection with self-supervised learning loss, but their methods only work for some very simple pretext

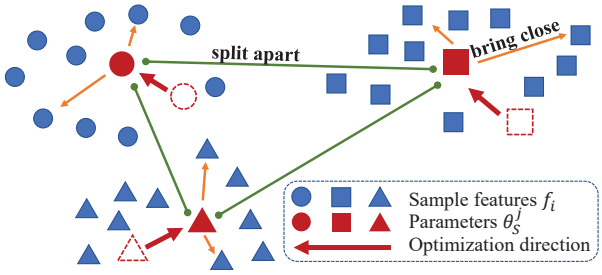


Figure 2. **Parametric Model Optimization Process:** By optimizing the loss in Eq. 11, each parameter θ_S^j is appealed by nearby sample features (orange in the figure, Eq. 9) and repelled by other parameters $\theta_S^k, k \neq j$ (green in the figure, Eq. 10).

task (*e.g.* colorization, rotation).

Most above active learning algorithms are designed for from-scratch training. Prior research [4] reveals their negative effect in finetuning after unsupervised pretraining.

3. Methodology

We first formulate this new task called *active finetuning* in Sec. 3.1. Our novel method, ActiveFT, to solve this problem based on continuous space optimization is proposed in Sec. 3.2. Afterward, we elaborate on how to optimize this model by minimizing the loss function in Sec. 3.3. An illustration of our method is shown in Fig. 2. We also clarify the correlation between our method and earth mover's distance in Sec. 3.4. Finally, the implementation of this method to deep learning model is explained in Sec. 3.5.

3.1. Formulation of Active Finetuning Task

We formally define the active finetuning task. As is demonstrated in Fig. 1, a deep neural network model $f(\cdot; w_0) : \mathcal{X} \to \mathbb{R}^C$ with pretrained weight w_0 is given, where $\mathcal{X}$ is the data space and $\mathbb{R}^C$ is the normalized high-dimensional feature space. We also have access to a large unlabeled data pool $\mathcal{P}^u = \{\mathbf{x}_i\}_{i\in[N]} \sim p_u$ inside data space $\mathcal{X}$ with distribution p_u, where $[N] = \{1, 2, \ldots, N\}$. The subset $\mathcal{P}^u_{\mathcal{S}}$ for supervised finetuning is selected from $\mathcal{P}^u$. It is worth noting that $f(\cdot; w_0)$ can be pretrained either on $\mathcal{P}^u$ or other data sources, *e.g.* pretrained on ImageNet-1k [37] and finetuned on a subset of CIFAR-10 [23].

In the active finetuning task, we should design a sampling strategy $\mathcal{S} = \{s_j \in [N]\}_{j\in[B]}$ to select a subset $\mathcal{P}^u_{\mathcal{S}} = \{\mathbf{x}_{s_j}\}_{j\in[B]} \subset \mathcal{P}^u$ from $\mathcal{P}^u$, where B is the annotation budget size for supervised finetuning. The model would have access to the labels $\{\mathbf{y}_{s_j}\}_{j\in[B]} \subset \mathcal{Y}$ of this subset through the oracle, obtaining a labeled data pool $\mathcal{P}^l_{\mathcal{S}} = \{\mathbf{x}_{s_j}, \mathbf{y}_{s_j}\}_{j\in[B]}$, where $\mathcal{Y}$ is the label space. Afterward, the model f is finetuned on $\mathcal{P}^l_{\mathcal{S}}$ supervisedly and the model parameter is updated to $w_{\mathcal{S}}$ after the finetuning. The goal of active finetuning is to find the sampling strategy $\mathcal{S}_{opt}$ minimizing the expectation model error $error(f(\mathbf{x}; w_{\mathcal{S}}), \mathbf{y})$.

$$\mathcal{S}_{opt} = \arg\min_{\mathcal{S}} \mathop{E}_{\mathbf{x},\mathbf{y}\in\mathcal{X}\times\mathcal{Y}} [error(f(\mathbf{x}; w_{\mathcal{S}}), \mathbf{y})] \quad (1)$$

Our active finetuning is different from traditional active learning in: 1) We have access to the pretrained model $f(\cdot; w_0)$, which will be finetuned, before data selection. 2) The selected samples are applied to the finetuning of the pretrained model $f(\cdot; w_0)$ instead of from-scratch training. 3) The sampled subset size $|\mathcal{P}^l_{\mathcal{S}}|$ is relatively small, less than 10% in most cases. 4) We have no access to any labels such as a random initial labeled set before data selection.

3.2. Data Selection with Parametric Model

We select samples under the guidance two basic intuitions: *1) bringing close the distributions between the selected subset $\mathcal{P}^u_{\mathcal{S}}$ and the original pool $\mathcal{P}^u \sim p_u$. 2) maintaining the diversity of $\mathcal{P}^u_{\mathcal{S}}$.* The former ensures the model finetuned on the subset performs similarly with that trained on the full set, while the latter allows the subset to cover corner cases in the full set. In comparison to distribution $p_u(\mathbf{x})$ in the data space, it is more feasible to work on its corresponding distribution $p_{f_u}(\mathbf{f})$ in the feature space. Through the agency of pretrained model $f(\cdot; w_0)$, we map each data sample $\mathbf{x}_i$ to the high dimensional feature space as $\mathbf{f}_i = f(\mathbf{x}_i; w_0)$, where $\mathbf{f}_i$ is the *normalized* feature of $\mathbf{x}_i$. As a result, we can derive the pool $\mathcal{F}^u = \{\mathbf{f}_i\}_{i\in[N]}$ from $\mathcal{P}^u$ and corresponding distribution p_{f_u} of $\mathcal{F}^u$.

Similarly, the feature pool $\mathcal{F}^u_{\mathcal{S}}$ is also associated with the selected data subset $\mathcal{P}^u_{\mathcal{S}}$. We define the corresponding distribution over $\mathcal{F}^u_{\mathcal{S}}$ in the feature space as $p_{f_{\mathcal{S}}}$. Our goal is to find the optimal selection strategy $\mathcal{S}$ as follows.

$$\mathcal{S}_{opt} = \arg\min_{\mathcal{S}} D(p_{f_u}, p_{f_{\mathcal{S}}}) - \lambda\mathcal{R}(\mathcal{F}^u_{\mathcal{S}}) \quad (2)$$

where $D(\cdot,\cdot)$ is some distance metrics between distributions, $\mathcal{R}(\cdot)$ is to measure the diversity of a set, and λ is a scale to balance these two terms. The first term aims to bring close these two distributions $p_{f_u}, p_{f_{\mathcal{S}}}$ while the second term is to ensure the diversity of subset.

Unfortunately, it is difficult to directly optimize the *discrete* selection strategy $\mathcal{S}$, so we alternatively model $p_{f_{\mathcal{S}}}$ with $p_{\theta_{\mathcal{S}}}$, where $\theta_{\mathcal{S}} = \{\theta^j_{\mathcal{S}}\}_{j\in[B]}$ are the *continuous* parameters and B is the annotation budget size. Each $\theta^j_{\mathcal{S}}$ after optimization corresponds to the feature of a selected sample $\mathbf{f}_{s_j}$. We would find $\mathbf{f}_{s_j}$ closest to each $\theta^j_{\mathcal{S}}$ after optimization to determine the selection strategy $\mathcal{S}$. Therefore, our goal in Eq. 2 is written as follows.

$$\theta_{\mathcal{S},opt} = \arg\min_{\theta_{\mathcal{S}}} D(p_{f_u}, p_{\theta_{\mathcal{S}}}) - \lambda\mathcal{R}(\theta_{\mathcal{S}}) \; s.t. \; ||\theta^j_{\mathcal{S}}||_2 = 1 \quad (3)$$

The difference between extracted sample features $\mathcal{F}^u_{\mathcal{S}} = \{\mathbf{f}_{s_i}\}$ and our define parameters $\theta_{\mathcal{S}} = \{\theta^j_{\mathcal{S}}\}$ is that $\mathbf{f}_{s_i}$ is

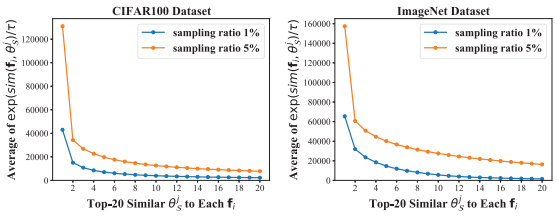


Figure 3. **Similarity between Features and Parameters:** On CIFAR100 and ImageNet, we find the Top-20 most similar parameters $\theta^j_{\mathcal{S}}$ with each sample feature $\mathbf{f}_i$, and calculate the average exponential similarity $E_{i\in[N]}[\exp(sim(\mathbf{f}_i, \theta^j_{\mathcal{S}})/\tau]$. Here $\theta_{\mathcal{S}} = \{\theta^j_{\mathcal{S}}\}_{j\in[B]}$ is randomly sampled following the distribution p_{f_u}. The model $f(\cdot; w_0)$ is DeiT-Small [42] pretrained on ImageNet [37] with DINO framework [6]. The results verify Assumption 1 that the Top-1 similarity is significantly larger than others.

a discrete feature corresponding to a sample in the dataset while $\theta^j_{\mathcal{S}}$ is continuous in the feature space.

3.3. Parametric Model Optimization

In the parametric model $p_{\theta_{\mathcal{S}}}$, the distribution is represented by B parameters $\{\theta^j_{\mathcal{S}}\}_{j\in[B]}$. We consider it as a mixture model with B components in Eq. 4.

$$p_{\theta_{\mathcal{S}}}(\mathbf{f}) = \sum_{j=1}^{B} \phi_j p(\mathbf{f}|\theta^j_{\mathcal{S}}) \tag{4}$$

where ϕ_j is the mixture weight or prior probability $p(\theta^j_{\mathcal{S}})$ of the j-th component, satisfying $\sum_{j=1}^{B}\phi_j = 1$. Since $\mathbf{f}$ and $\theta^j_{\mathcal{S}}$ both lie in the feature space, we formulate the distribution of each component based on their similarity as Eq. 5.

$$p(\mathbf{f}|\theta^j_{\mathcal{S}}) = \frac{\exp(sim(\mathbf{f}, \theta^j_{\mathcal{S}})/\tau)}{Z_j} \tag{5}$$

where Z_j is a normalizing constant, $sim(\cdot,\cdot)$ is a similarity metric, and τ is the temperature scale. We follow the protocol in [6, 46] to apply the cosine similarity between normalized features as the metric $sim(\mathbf{f}_1, \mathbf{f}_2) = \mathbf{f}_1^\top \mathbf{f}_2, ||\mathbf{f}_1||_2 = ||\mathbf{f}_2||_2 = 1$ and set the temperature $\tau = 0.07$ [6,46] throughout the paper. For each $\mathbf{f}_i \in \mathcal{F}^u$, there exists a $\theta^{c_i}_{\mathcal{S}}$ most similar (and closest) to $\mathbf{f}_i$, *i.e.*

$$c_i = \arg\max_{j\in[B]} sim(\mathbf{f}_i, \theta^j_{\mathcal{S}}) \tag{6}$$

where we keep updating c_i in the optimization process.

Since there is a very low temperature ($\tau = 0.07$), the gap between the exponential similarity $\exp(sim(\mathbf{f}_i, \theta^j_{\mathcal{S}})/\tau)$ with different $\theta^j_{\mathcal{S}}$ is significant. Therefore, it is safe to make the following *assumption*.

Assumption 1 *$\forall i \in [N], j \in [B]$, if τ is small, the following far-more-than relationship holds that*

$$\exp(sim(\mathbf{f}_i, \theta^{c_i}_{\mathcal{S}})/\tau) \gg \exp(sim(\mathbf{f}_i, \theta^j_{\mathcal{S}})/\tau), j \neq c_i$$

Although it is hard to mathematically prove, this assumption is empirically verified by the results in Fig. 3. In another word, $p(\mathbf{f}_i|\theta^{c_i}_{\mathcal{S}}) \gg p(\mathbf{f}_i|\theta^j_{\mathcal{S}}), j \neq c_i, j \in [B]$. We can approximate the parametric model for $\mathbf{f}_i \in \mathcal{F}^u$ in Eq. 4.

$$\begin{aligned} p_{\theta_{\mathcal{S}}}(\mathbf{f}_i) &\approx \phi_{c_i} p(\mathbf{f}_i|\theta^{c_i}_{\mathcal{S}}) \\ &= \frac{\exp(sim(\mathbf{f}_i, \theta^{c_i}_{\mathcal{S}})/\tau)}{Z_{c_i}/\phi_{c_i}} \\ &= \frac{\exp(sim(\mathbf{f}_i, \theta^{c_i}_{\mathcal{S}})/\tau)}{\tilde{Z}_{c_i}} \end{aligned} \tag{7}$$

where $\tilde{Z}_{c_i} = Z_{c_i}/\phi_{c_i}$ is a new normalizing constant. We can derive $p_{\theta_{\mathcal{S}}}(\mathbf{f}_i) \propto \exp(sim(\mathbf{f}_i, \theta^{c_i}_{\mathcal{S}})/\tau)$ from Eq. 7.

The two distributions $p_{f_u}, p_{\theta_{\mathcal{S}}}$ can be brought close by minimizing the KL-divergence.

$$\begin{aligned} KL(p_{f_u}|p_{\theta_{\mathcal{S}}}) &= \sum_{\mathbf{f}_i\in\mathcal{F}^u} p_{f_u}(\mathbf{f}_i) \log \frac{p_{f_u}(\mathbf{f}_i)}{p_{\theta_{\mathcal{S}}}(\mathbf{f}_i)} \\ &= \mathop{E}_{\mathbf{f}_i\in\mathcal{F}^u}[\log p_{f_u}(\mathbf{f}_i)] - \mathop{E}_{\mathbf{f}_i\in\mathcal{F}^u}[\log p_{\theta_{\mathcal{S}}}(\mathbf{f}_i)] \end{aligned} \tag{8}$$

where the first term $\mathop{E}_{\mathbf{f}_i\in\mathcal{F}^u}[\log p_{f_u}(\mathbf{f}_i)]$ is a constant without the parameter $\theta_{\mathcal{S}}$. Then, minimizing the KL-divergence $KL(p_{f_u}|p_{\theta_{\mathcal{S}}})$ equals to maximizing the second term $\mathop{E}_{\mathbf{f}_i\in\mathcal{F}^u}[\log p_{\theta_{\mathcal{S}}}(\mathbf{f}_i)]$, and according to Eq. 7, it is also equivalent with maximizing $\mathop{E}_{\mathbf{f}_i\in\mathcal{F}^u}[\log \exp(sim(\mathbf{f}_i, \theta^{c_i}_{\mathcal{S}})/\tau)] = \mathop{E}_{\mathbf{f}_i\in\mathcal{F}^u}[sim(\mathbf{f}_i, \theta^{c_i}_{\mathcal{S}})/\tau]$. Therefore, we derive the first term in Eq. 3 as follows.

$$D(p_{f_u}, p_{\theta_{\mathcal{S}}}) = -\mathop{E}_{\mathbf{f}_i\in\mathcal{F}^u}[sim(\mathbf{f}_i, \theta^{c_i}_{\mathcal{S}})/\tau] \tag{9}$$

However, directly carrying out this optimization leads to a severe *collapse* problem, *i.e.* most $\theta^j_{\mathcal{S}}, j \in [B]$ converge to the same position with the highest density of $\mathbf{f}_i, i \in [N]$, losing the diversity inside the selected data. To this end, as shown in Eq. 2, we introduce an extra regularization term to ensure the diversity of selected subset. Without bells and whistles, this regularization is implemented by minimizing the similarity between selected samples. We also add an exponential operation to make the optimization process more stable, otherwise some $\theta^j_{\mathcal{S}}$ may become outliers.

$$R(\theta_{\mathcal{S}}) = \mathop{E}_{j\in[B]}\left[\log \sum_{k\neq j, k\in[B]} \exp\left(sim(\theta^j_{\mathcal{S}}, \theta^k_{\mathcal{S}})/\tau\right)\right] \tag{10}$$

At this point, we are able to solve Eq. 3 by optimizing the following loss function continuously.

$$\begin{aligned} L &= D(p_{f_u}, p_{\theta_{\mathcal{S}}}) - \lambda \cdot R(\theta_{\mathcal{S}}) \\ &= -\mathop{E}_{\mathbf{f}_i\in\mathcal{F}^u}[sim(\mathbf{f}_i, \theta^{c_i}_{\mathcal{S}})/\tau] + \mathop{E}_{j\in[B]}\left[\log \sum_{k\neq j, k\in[B]} \exp\left(sim(\theta^j_{\mathcal{S}}, \theta^k_{\mathcal{S}})/\tau\right)\right] \end{aligned} \tag{11}$$

where the balance weight λ is empirically set as 1.

Finally, we directly optimize the loss function in Eq. 11 by gradient descent. When the optimization is finished, we find feature $\{\mathbf{f}_{s_j}\}_{j\in[B]}$ with the highest similarity to $\theta^j_{\mathcal{S}}$.

$$\mathbf{f}_{s_j} = \arg\max_{\mathbf{f}_k\in\mathcal{F}^u} sim(\mathbf{f}_k, \theta^j_{\mathcal{S}}) \tag{12}$$

The corresponding data samples $\{\mathbf{x}_{s_j}\}_{j\in[B]}$ are selected as the subset $\mathcal{P}^u_{\mathcal{S}}$ with selection strategy $\mathcal{S} = \{s_j\}_{j\in[B]}$.

3.4. Relation to Earth Mover's Distance

In this part, we show that optimizing the loss in Eq. 11 is actually minimizing the earth mover's distance between the distributions of selected subset and full set. This justifies that our optimization in the continuous space is equivalent with bringing close the distribution gap in the discrete data sample space.

After the optimization, we get the features $\mathbf{f}_{s_j}$ of selected samples. We deliberately assign the discrete probability distribution p_{f_S} as Eq. 13.

$$p_{f_S}(\mathbf{f}_{s_j}) = \frac{|C_j|}{N}, C_j = \{\mathbf{f}_i | c_i = j\}, \mathbf{f}_{s_j} \in \mathcal{F}^u_{\mathcal{S}} \tag{13}$$

where C_j is the set of features closest to f_{s_j} with c_i defined in Eq. 6. The distribution p_{f_u} is modeled as a uniform distribution over $\mathcal{F}^u$, *i.e.* $p_{f_u}(\mathbf{f}_i) = \frac{1}{N}, \mathbf{f}_i \in \mathcal{F}^u$.

The earth mover's distance (EMD) between p_{f_u}, p_{f_S} is written as [24]:

$$EMD(p_{f_u}, p_{f_S}) = \inf_{\gamma\in\Pi(p_{f_u}, p_{f_S})} \mathop{E}_{(\mathbf{f}_i, \mathbf{f}_{s_j})\sim\gamma} \left[||\mathbf{f}_i - \mathbf{f}_{s_j}||_2\right] \tag{14}$$

where $\Pi(p_{f_u}, p_{f_S})$ is the set of all possible joint distributions whose marginals are p_{f_u} and p_{f_S}. It is intuitive to come up with the infimum, *i.e.* each $\mathbf{f}_i \sim p_{f_u}$ transports to their closest $\mathbf{f}_{s_j} \sim p_{f_S}$. The detailed derivation is in the supplementary material.

$$\gamma_{f_u, f_S}(\mathbf{f}_i, \mathbf{f}_{s_j}) = \begin{cases} \frac{1}{N} & \mathbf{f}_i \in \mathcal{F}^u, \mathbf{f}_{s_j} \in \mathcal{F}^u_{\mathcal{S}}, c_i = j \\ 0 & otherwise \end{cases} \tag{15}$$

In this case, the distance in Eq. 14 becomes

$$\begin{aligned} EMD(p_{f_u}, p_{f_S}) &= \mathop{E}_{(\mathbf{f}_i, \mathbf{f}_{c_i})\sim\gamma} \left[||\mathbf{f}_i - \mathbf{f}_{s_{c_i}}||_2\right] \\ &= \frac{1}{N}\sum_{i=1}^{N} \left[\sqrt{2 - 2sim(\mathbf{f}_i, \mathbf{f}_{s_{c_i}})}\right] \end{aligned} \tag{16}$$

In Eq. 11, we minimize $-sim(\mathbf{f}_i, \theta^{c_i}_{\mathcal{S}})$, and $\mathbf{f}_{s_{c_i}}$ is set as the closest $\mathbf{f}_k \in \mathcal{F}^u$ to $\theta^{c_i}_{\mathcal{S}}$ in Eq. 12 after optimization. Then, the distance in Eq. 16 is actually also minimized. Therefore, our optimization method in Sec. 3.3 is equivalent with reducing the earth mover's distance between the distributions of the original unlabeled pool and selected subset.

Algorithm 1: Pseudo-code for ActiveFT

Input: Unlabeled data pool $\{\mathbf{x}_i\}_{i\in[N]}$, pretrained model $f(\cdot; w_0)$, annotation budget B, iteration number T for optimization

Output: Optimal selection strategy $\mathcal{S} = \{s_j \in [N]\}_{j\in[B]}$

for $i \in [N]$ **do**
  $\mathbf{f}_i = f(\mathbf{x}_i; w_0)$
/* Construct $\mathcal{F}^u = \{\mathbf{f}_i\}_{i\in[N]}$ based on $\mathcal{P}^u$, normalized to $||\mathbf{f}_i||_2 = 1$ */
Uniformly random sample $\{s^0_j \in [N]\}_{j\in[B]}$, and initialize $\theta^j_{\mathcal{S}} = \mathbf{f}_{s^0_j}$
/* Initialize the parameter $\theta_{\mathcal{S}} = \{\theta^j_{\mathcal{S}}\}_{j\in[B]}$ based on $\mathcal{F}^u$ */
for $iter \in [T]$ **do**
  Calculate the similarity between $\{\mathbf{f}_i\}_{i\in[N]}$ and $\{\theta^j_{\mathcal{S}}\}_{j\in[B]}$: $Sim_{i,j} = \mathbf{f}_i^\top \theta^j_{\mathcal{S}}/\tau$
  $MaxSim_i = \max_{j\in[B]} Sim_{i,j} = Sim_{i,c_i}$
  /* The Top-1 similarity between $\mathbf{f}_i$ and $\theta^j_{\mathcal{S}}, j \in [B]$ */
  Calculate the similarity between $\theta^j_{\mathcal{S}}$ and $\theta^k_{\mathcal{S}}, k \neq j$ for regularization: $RegSim_{j,k} = \exp({\theta^j_{\mathcal{S}}}^\top \theta^k_{\mathcal{S}}/\tau), k \neq j$
  $Loss = -\frac{1}{N}\sum_{i\in[N]} MaxSim_i + \frac{1}{B}\sum_{j\in[B]} \log\left(\sum_{k\neq j} RegSim_{j,k}\right)$
  /* Calculate the loss function in Eq. 11 */
  $\theta_{\mathcal{S}} = \theta_{\mathcal{S}} - lr \cdot \bigtriangledown_{\theta_{\mathcal{S}}} Loss$
  /* Optimize the parameter through gradient descent */
  $\theta^j_{\mathcal{S}} = \theta^j_{\mathcal{S}}/||\theta^j_{\mathcal{S}}||_2, j \in [B]$
  /* Normalize the parameters to ensure $||\theta^j_{\mathcal{S}}||_2 = 1$ */
Find $\mathbf{f}_{s_j}$ closest to $\theta^j_{\mathcal{S}}$: $s_j = \arg\max_{k\in[N]} \mathbf{f}_k^\top \theta^j_{\mathcal{S}}$ for each $j \in [B]$
Return the selection strategy $\mathcal{S} = \{s_j\}_{j\in[B]}$

3.5. Implementation as a Learning Model

Alg. 1 shows how to implement this method to deep learning models. Given a pretrained model, for each image sample $\mathbf{x}_i \in \mathcal{P}^u$, we extract the last layer [CLS] token feature in the transformer model or global pooling feature in the convolutional model, which is normalized as the high-dimensional feature $\mathbf{f}_i = f(\mathbf{x}_i; w_0)$. Before the optimization process, the parameter $\theta_{\mathcal{S}}$ is initialized by uniformly sampling $\theta^j_{\mathcal{S}}, j \in [B]$ at random from the feature pool $\mathcal{F}^u = \{\mathbf{f}_i\}_{i\in[N]}$. If $|\mathcal{F}^u|$ is extremely large, we would randomly select M elements from $\mathcal{F}^u$ (*e.g.* M=100,000 for ImageNet dataset) for the each training iteration of our

parametric model. In each iteration, we calculate the similarity between sample features and parameters, then update c_i in Eq. 6 for each $\mathbf{f}_i$ and positive feature set $\{\mathbf{f}_i|c_i = j\}$ for each $\theta_{\mathcal{S}}^j$. Afterwards, we can compute the loss function in Eq. 11 and update the parameters $\theta_{\mathcal{S}}$ by gradient descent. When the optimization process is finished, we find the sample feature $\mathbf{f}_{s_j}$ most similar to each parameter $\theta_{\mathcal{S}}^j$ (Eq. 12). Those corresponding samples $\{\mathbf{x}_{s_j}\}_{j\in[B]}$ are selected for annotation for the following supervised finetuning.

4. Experiments

Our method is evaluated on image classification (Sec. 4.1) and semantic segmentation (Sec. 4.2) tasks. The performance is compared with some baselines and traditional active learning algorithms. We make some qualitative and quantitative analyses of our method in Sec. 4.3. Finally, the roles of different modules inside our method are examined in Sec. 4.4. Experiments are run on GeForce RTX 3090 (24GB) and AMD Ryzen Threadripper 3970X 32-Core Processor.

4.1. Image Classification Results

Datasets and Benchmarks For classification task, we choose three classical datasets CIFAR10, CIFAR100 [23], and ImageNet-1k [37] for experiments. CIFAR10 and CIFAR100 contain 60,000 images of 32x32 scale with 10 and 100 categories separately, among which 50,000 images are in the training set and 10,000 images are for test. ImageNet-1k is a large-scale dataset spans 1000 classes, containing 1,281,167 training images and 50,000 validation images. Their training sets are considered as the unlabeled pool $\mathcal{P}^u$ for selection. We evaluate the performance with the *Top-1 Accuracy* metric.

Implementation Details Our method is agnostic with pretraining frameworks and networks. We apply DeiT-Small [42] pretrained with DINO [6] framework on ImageNet-1k [37] in the experiments for its verified popularity and effectiveness. We also attempt other architectures in Sec. 4.3. For all three datasets, we resize images to 224x224 consistent with the pretraining for both data selection and supervised finetuning. In the data selection process, the parameters $\theta_{\mathcal{S}}$ are optimized with Adam [22] optimizer (learning rate 1e-3) until convergence. The experiment details of supervised finetuning are available in the *supplementary materials*.

Baselines We compare our method with eight counterparts including the following three baselines and five traditional active learning methods.

1. **Random:** The samples for annotation are selected uniformly at random.

2. **FDS:** *a.k.a* K-Center-Greedy algorithm. It selects the next sample feature farthest from current selections. Proved in [38], it minimizes the gap between the expected loss over the entire pool and the selected subset. In accordance with the pretraining process [6], we apply cosine distance as the distance metric.

3. **K-Means:** We conduct K-Means over the feature pool $\mathcal{F}^u$ and choose samples closest to the centers. K equals to the budget size B.

We transplant five active learning algorithms CoreSet [38], VAAL [39], LearnLoss [48], TA-VAAL [21] and ALFA-Mix [33] to our active finetuning task. The former three are classical and the latter two are newer, all equipped with open-source codes. We refer readers to the supplementary materials for transplantation details.

Results and Comparison We average our results over three independent runs. The results are shown in Tab. 1. Traditional active learning methods typically fail in the pretraining-finetuning paradigm, which is consistent with the results in [4]. In contrast, our ActiveFT outperforms counterparts on all three datasets with different sampling ratios. On each dataset, the performance gain is especially significant when the sampling ratio is low, since our method can select the most representative samples. This phenomenon is of great practical use because the sampling number for supervised finetuning is usually much smaller than the pool size to save the annotation cost.

4.2. Semantic Segmentation Results

Datasets and Benchmarks For segmentation task, we apply ADE20k dataset [50]. It contains 20,210 images for training, 2,000 images for validation, and 3,352 images for test. All images have fine-grained labels with 150 semantic classes. The training set is considered as the unlabeled pool $\mathcal{P}^u$ for selection. We evaluate the performance with the *mIoU* metric.

Implementation Details Same with image classification task, we apply DeiT-Small [42] model pretrained with DINO framework [6] for data selection. The images are resized to 224x224 as well. Since the semantic segmentation task relies more on the local information inside images, we concatenate the [CLS] token features with the average features of other tokens as $\mathbf{f}_i$ for data selection. For the segmentation task, Segmenter [40] is adopted for finetuning, which is a pure transformer model. We use the same pretrained DeiT-Small [6] as its backbone. The finetuning details are also available in supplementary materials.

Results and Comparison In Tab. 2, we report the performance of our method when choosing $5\%, 10\%$ of training samples for finetuning. Our results are averaged over three independent trials. We compare to three baselines same with image classification. The traditional active learning methods are not included due to their failure on image classification. The performance gain of our data selection

Table 1. **Image Classification Results:** Experiments are conducted on natural images with different sampling ratios. We report the mean and std over three trials. Explanation of N/A results ("-") is in our supplementary materials.

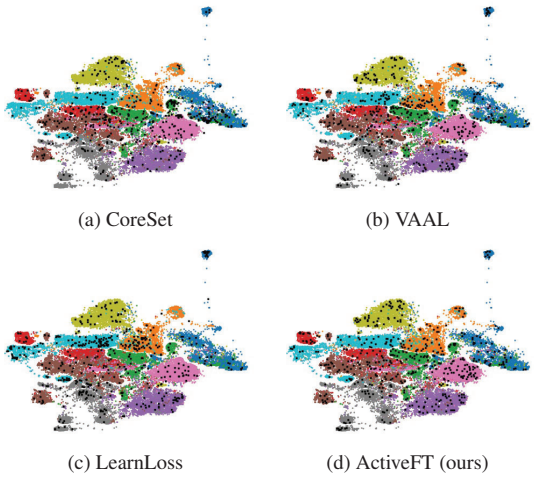
Methods	CIFAR10			CIFAR100				ImageNet	
	0.5%	1%	2%	1%	2%	5%	10%	1%	5%
Random	77.3±2.6	82.2±1.9	88.9±0.4	14.9±1.9	24.3±2.0	50.8±3.4	69.3±0.7	45.1±0.8	64.3±0.3
FDS	64.5±1.5	73.2±1.2	81.4±0.7	8.1±0.6	12.8±0.3	16.9±1.4	52.3±1.9	26.7±0.6	55.5±0.1
K-Means	83.0±3.5	85.9±0.8	89.6±0.6	17.6±1.1	31.9±0.1	42.4±1.0	70.7±0.3	-	-
CoreSet [38]	-	81.6±0.3	88.4±0.2	-	30.6±0.4	48.3±0.5	62.9±0.6	-	61.7±0.2
VAAL [39]	-	80.9±0.5	88.8±0.3	-	24.6±1.1	46.4±0.8	70.1±0.4	-	64.0±0.3
LearnLoss [48]	-	81.6±0.6	86.7±0.4	-	19.2±2.2	38.2±2.8	65.7±1.1	-	63.2±0.4
TA-VAAL [21]	-	82.6±0.4	88.7±0.2	-	34.7±0.7	46.4±1.1	66.8±0.5	-	64.3±0.2
ALFA-Mix [33]	-	83.4±0.3	89.6±0.2	-	35.3±0.8	50.4±0.9	69.9±0.6	-	64.5±0.2
ActiveFT (ours)	**85.0**±0.4	**88.2**±0.4	**90.1**±0.2	**26.1**±2.6	**40.7**±0.9	**54.6**±2.3	**71.0**±0.5	**50.1**±0.3	**65.3**±0.1

Table 2. **Semantic Segmentation Results:** experiments are conducted on ADE20k with sampling ratios 5%, 10%. Results are averaged over three trials.

Sel. Ratio	Random	FDS	K-Means	ActiveFT (ours)
5%	14.54	6.74	13.62	**15.37**±0.11
10%	20.27	12.65	19.12	**21.60**±0.40

Table 3. **Data Selection Efficiency:** We compare the time cost to select different percentages of samples from the CIFAR100 training set.

Sel. Ratio	K-Means	CoreSet	VAAL	LearnLoss	ours
2%	16.6s	1h57m	7h52m	20m	**12.6s**
5%	37.0s	7h44m	12h13m	1h37m	**21.9s**
10%	70.2s	20h38m	36h24m	9h09m	**37.3s**



(a) CoreSet (b) VAAL (c) LearnLoss (d) ActiveFT (ours)

Figure 4. **tSNE Embeddings of CIFAR10:** We visualize the embedding of selected samples using different algorithms. Different colors denote categories, and the black dots are the 1% samples selected by our method.

method is not as significant as the image classification task. This is understandable because semantic segmentation is a fine-grained vision task, focusing on subtle local visual pattern. In this case, it is hard for a global feature to represent all the details in a scene. However, despite the difficulty, our method still shows notable superiority in comparison with other baselines, reflecting the generality of our method to different tasks.

4.3. Analysis

Data Selection Efficiency It is desirable that the data selection method operates in a time-efficient manner, as close as possible to random selection. In Tab. 3, we compare the required time to select different percentages of training samples from CIFAR100. We do not take FDS into account due to its very poor performance. For those traditional active learning algorithms, both the repetitive model training and data sampling should be counted into the running time, and the former takes the majority. In contrast, our method chooses all the samples in a single-pass, so we do not have to train the model again in the selection process. As a result, our method's speed is faster than traditional active learning methods by a notable margin. Besides, unlike some active learning methods [38, 48], our method does not need access to ground-truths before the end of all selection, which enables more flexibility of annotator assignment.

Visualization of Selected Samples In Fig. 4, we visualize the feature $\mathbf{f}_i$ of each sample in CIFAR10 training set. The dimension of features is reduced by tSNE. The black dots represent the 1% samples selected by different methods. Results demonstrate that our selected samples distribute more similarly with the entire pool in the feature space than other counterparts. It reflects that optimizing our proposed parametric model helps to reduce the distribution gap between the selected subset and original unlabeled pool.

Table 4. **Generality on Pretraining Frameworks and Model Architectures:** We examine the performance of ActiveFT on different pretraining frameworks and models on CIFAR-10.

(a) **Performance on DeiT-Small Pretrained with iBOT**

Methods	0.5%	1%	2%
Random	81.7	83.0	89.8
CoreSet [38]	-	82.8	89.2
LearnLoss [48]	-	83.6	89.2
VAAL [39]	-	85.1	89.3
ActiveFT (ours)	**87.6**±0.8	**88.3**±0.2	**90.9**±0.2

(b) **Performance on ResNet-50 Pretrained with DINO**

Methods	0.5%	1%	2%
Random	64.8	76.2	83.7
CoreSet [38]	-	70.4	83.2
LearnLoss [48]	-	71.7	81.3
VAAL [39]	-	75.0	83.3
ActiveFT (ours)	**68.5** ±0.4	**78.6** ±0.7	**84.9** ±0.3

Table 5. **Ablation Study:** We examine the effect of two modules in our method. Experiments are conducted on CIFAR100 with pretrained DeiT-Small model.

(a) c_i **Update Manner**

Ratio	No-Update	Update
2%	20.6	**40.7**
5%	52.8	**54.6**

(b) **Regularization Design**

Ratio	S1	S2	ours
2%	33.1	26.8	**40.7**
5%	51.5	46.9	**54.6**

Generality of our Method ActiveFT can fit different pretraining frameworks and models well. In addition to DINO [6] framework and DeiT-Small [42] model, we also apply ActiveFT to a DeiT-Small [42] trained with generative unsupervised pretraining framework iBOT [51] and CNN model ResNet-50 [16] trained with DINO [6]. The models are pretrained on ImageNet-1k and would be finetuned in CIFAR10. Other implementation details are exactly same with Sec. 4.1. Tab 4 shows the significant superiority of our results in comparison with random sampling baseline with different sampling ratios. The results reflect the compatibility of ActiveFT with different unsupervised pretraining frameworks and model architectures.

4.4. Ablation Study

We discuss the importance of different modules in our method including the update manner of c_i, the design of diversity regularization, and the effect of temperature.

Update Manner of c_i In this part, we discuss the ways to update c_i (Eq. 6) which denotes the parameter closest to each sample $\mathbf{f}_i$ in the feature space. In Alg. 1, it is updated in each iteration. Alternatively, we remain c_i unchanged as the initial state in the optimization process. Results in Tab. 5a shows that this stationary strategy does not work well. In this case, it would rely heavily on the initial state. The frequent update of c_i could help to relieve some harmful biases inside the initial state.

Table 6. **Effect of Temperatures:** We try different temperatures in our method. Experiments are conducted on CIFAR10 with pretrained DeiT-Small model.

Ratio	$\tau = 0.04$	$\tau = 0.07$	$\tau = 0.2$	$\tau = 0.5$
0.5%	**85.6**	85.0	84.1	83.5
1%	87.4	**88.2**	85.3	86.1
2%	**90.3**	90.1	89.6	89.0

Regularization Design We try two alternative strategies to design the regularization term $R(\cdot)$ in Eq. 11. **S1) No Regularization:** We only optimize the first term $D(\cdot,\cdot)$ in Eq. 11. **S2) InfoNCE [43]:** We get inspiration from [43] to design a contrastive loss to approximate the distribution p_{f_u} with p_{θ_S}: $L = -\underset{\mathbf{f}_i \in \mathcal{F}^u}{E}\left[\log \frac{\exp(\mathbf{f}_i^T \theta_S^{c_i}/\tau)}{\sum_{k\in[N]} \exp(sim(\mathbf{f}_k^T \theta_S^{c_i}/\tau)}\right]$. In Tab. 5b, we evaluate these three strategies. We find that both **S1** and **S2** fails, and only our applied strategy **S3** succeeds. It justifies our design of the regularization strategy.

Temperature τ We analyze the effect of different temperatures in Eq. 11. Pointed out in Assumption 1, a small τ is a pre-requisite for our derivation. Tab. 6 shows the results on CIFAR10 with different temperatures. When the temperature is relatively low (e.g. τ <0.1), the performance of ActiveFT is great. However, as it becomes higher (e.g. $\tau = 0.5$), the performance drops. The results are in line with our theoretical derivation.

5. Conclusion

To fill in the gap inside the pretraining-finetuning paradigm, we define the active finetuning task, which selects samples from an unlabeled data pool for supervised model finetuning. To solve this problem, we propose a model-agnostic algorithm, ActiveFT. By optimizing a parametric model, ActiveFT chooses diverse data samples distributing similarly with the original pool for annotation. It is mathematically justified that ActiveFT helps to bring close the distributions of the selected subset and entire data pool by reducing the Earth Mover's distance. Our experiments on classification and segmentation show the state-of-the-art performance of ActiveFT, with an extremely high data selection efficiency. We believe ActiveFT can help to exploit the annotation budget for supervised finetuning in practical use and make a solid contribution to the popular pretraining-finetuning paradigms in various tasks.

References

[1] Sharat Agarwal, Himanshu Arora, Saket Anand, and Chetan Arora. Contextual diversity for active learning. In *European Conference on Computer Vision*, pages 137–153. Springer, 2020. 2

[2] Hangbo Bao, Li Dong, and Furu Wei. Beit: Bert pre-training of image transformers. *arXiv preprint arXiv:2106.08254*, 2021. 2

[3] William H Beluch, Tim Genewein, Andreas Nürnberger, and Jan M Köhler. The power of ensembles for active learning in image classification. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 9368–9377, 2018. 2

[4] Javad Zolfaghari Bengar, Joost van de Weijer, Bartlomiej Twardowski, and Bogdan Raducanu. Reducing label effort: Self-supervised meets active learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1631–1639, 2021. 1, 3, 6

[5] Mathilde Caron, Ishan Misra, Julien Mairal, Priya Goyal, Piotr Bojanowski, and Armand Joulin. Unsupervised learning of visual features by contrasting cluster assignments. *Advances in Neural Information Processing Systems*, 33:9912–9924, 2020. 2

[6] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9650–9660, 2021. 2, 4, 6, 8

[7] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, pages 1597–1607. PMLR, 2020. 1, 2

[8] Xinlei Chen and Kaiming He. Exploring simple siamese representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15750–15758, 2021. 2

[9] Xinlei Chen, Saining Xie, and Kaiming He. An empirical study of training self-supervised vision transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9640–9649, 2021. 2

[10] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*, 2020. 2

[11] Meng Fang, Yuan Li, and Trevor Cohn. Learning how to active learn: A deep reinforcement learning approach. *arXiv preprint arXiv:1708.02383*, 2017. 2

[12] Alexander Freytag, Erik Rodner, and Joachim Denzler. Selecting influential examples: Active learning with expected model output changes. In *European conference on computer vision*, pages 562–577. Springer, 2014. 2

[13] Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Guo, Mohammad Gheshlaghi Azar, et al. Bootstrap your own latent-a new approach to self-supervised learning. *Advances in Neural Information Processing Systems*, 33:21271–21284, 2020. 1, 2

[14] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. *arXiv preprint arXiv:2111.06377*, 2021. 1, 2

[15] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9729–9738, 2020. 2

[16] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 8

[17] Siyu Huang, Tianyang Wang, Haoyi Xiong, Jun Huan, and Dejing Dou. Semi-supervised active learning with temporal output discrepancy. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3447–3456, 2021. 2

[18] Siyuan Huang, Yichen Xie, Song-Chun Zhu, and Yixin Zhu. Spatio-temporal self-supervised representation learning for 3d point clouds. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6535–6545, 2021. 1

[19] Zhicheng Huang, Xiaojie Jin, Chengze Lu, Qibin Hou, Ming-Ming Cheng, Dongmei Fu, Xiaohui Shen, and Jiashi Feng. Contrastive masked autoencoders are stronger vision learners. *arXiv preprint arXiv:2207.13532*, 2022. 2

[20] Ajay J Joshi, Fatih Porikli, and Nikolaos Papanikolopoulos. Multi-class active learning for image classification. In *2009 ieee conference on computer vision and pattern recognition*, pages 2372–2379. IEEE, 2009. 2

[21] Kwanyoung Kim, Dongwon Park, Kwang In Kim, and Se Young Chun. Task-aware variational adversarial active learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8166–8175, 2021. 6, 7

[22] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 6

[23] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009. 2, 3, 6

[24] Elizaveta Levina and Peter Bickel. The earth mover's distance is the mallows distance: Some insights from statistics. In *Proceedings Eighth IEEE International Conference on Computer Vision. ICCV 2001*, volume 2, pages 251–256. IEEE, 2001. 5

[25] David D Lewis and Jason Catlett. Heterogeneous uncertainty sampling for supervised learning. In *Machine learning proceedings 1994*, pages 148–156. Elsevier, 1994. 2

[26] Pengfei Liu, Weizhe Yuan, Jinlan Fu, Zhengbao Jiang, Hiroaki Hayashi, and Graham Neubig. Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing. *arXiv preprint arXiv:2107.13586*, 2021. 1

[27] Zhuoming Liu, Hao Ding, Huaping Zhong, Weijia Li, Jifeng Dai, and Conghui He. Influence selection for active learning.

In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9274–9283, 2021. 2

[28] Wenjie Luo, Alex Schwing, and Raquel Urtasun. Latent structured active learning. *Advances in Neural Information Processing Systems*, 26, 2013. 2

[29] Dwarikanath Mahapatra, Behzad Bozorgtabar, Jean-Philippe Thiran, and Mauricio Reyes. Efficient active learning for image classification and segmentation using a sample selection and conditional generative adversarial network. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 580–588. Springer, 2018. 2

[30] Rafid Mahmood, Sanja Fidler, and Marc T Law. Low budget active learning via wasserstein distance: An integer programming approach. *arXiv preprint arXiv:2106.02968*, 2021. 2

[31] Christoph Mayer and Radu Timofte. Adversarial sampling for active learning. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 3071–3079, 2020. 2

[32] Alexander Narr, Rudolph Triebel, and Daniel Cremers. Stream-based active learning for efficient and adaptive classification of 3d objects. In *2016 IEEE International Conference on Robotics and Automation (ICRA)*, pages 227–233. IEEE, 2016. 2

[33] Amin Parvaneh, Ehsan Abbasnejad, Damien Teney, Gholamreza Reza Haffari, Anton van den Hengel, and Javen Qinfeng Shi. Active learning by feature mixing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12237–12246, 2022. 6, 7

[34] Pengzhen Ren, Yun Xiao, Xiaojun Chang, Po-Yao Huang, Zhihui Li, Brij B Gupta, Xiaojiang Chen, and Xin Wang. A survey of deep active learning. *ACM Computing Surveys (CSUR)*, 54(9):1–40, 2021. 1

[35] Yossi Rubner, Carlo Tomasi, and Leonidas J Guibas. A metric for distributions with applications to image databases. In *Sixth International Conference on Computer Vision (IEEE Cat. No. 98CH36271)*, pages 59–66. IEEE, 1998. 2

[36] Yossi Rubner, Carlo Tomasi, and Leonidas J Guibas. The earth mover's distance as a metric for image retrieval. *International journal of computer vision*, 40(2):99–121, 2000. 2

[37] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *International journal of computer vision*, 115(3):211–252, 2015. 2, 3, 4, 6

[38] Ozan Sener and Silvio Savarese. Active learning for convolutional neural networks: A core-set approach. *arXiv preprint arXiv:1708.00489*, 2017. 1, 2, 6, 7, 8

[39] Samarth Sinha, Sayna Ebrahimi, and Trevor Darrell. Variational adversarial active learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5972–5981, 2019. 1, 2, 6, 7, 8

[40] Robin Strudel, Ricardo Garcia, Ivan Laptev, and Cordelia Schmid. Segmenter: Transformer for semantic segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 7262–7272, 2021. 6

[41] Simon Tong and Daphne Koller. Support vector machine active learning with applications to text classification. *Journal of machine learning research*, 2(Nov):45–66, 2001. 2

[42] Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Hervé Jégou. Training data-efficient image transformers & distillation through attention. In *International Conference on Machine Learning*, pages 10347–10357. PMLR, 2021. 4, 6, 8

[43] Aaron Van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *arXiv e-prints*, pages arXiv–1807, 2018. 8

[44] Keze Wang, Dongyu Zhang, Ya Li, Ruimao Zhang, and Liang Lin. Cost-effective active learning for deep image classification. *IEEE Transactions on Circuits and Systems for Video Technology*, 27(12):2591–2600, 2016. 2

[45] Chen Wei, Haoqi Fan, Saining Xie, Chao-Yuan Wu, Alan Yuille, and Christoph Feichtenhofer. Masked feature prediction for self-supervised visual pre-training. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14668–14678, 2022. 2

[46] Zhirong Wu, Yuanjun Xiong, Stella X Yu, and Dahua Lin. Unsupervised feature learning via non-parametric instance discrimination. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3733–3742, 2018. 2, 4

[47] John Seon Keun Yi, Minseok Seo, Jongchan Park, and Dong-Geol Choi. Pt4al: Using self-supervised pretext tasks for active learning. In *European Conference on Computer Vision*, pages 596–612. Springer, 2022. 2

[48] Donggeun Yoo and In So Kweon. Learning loss for active learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 93–102, 2019. 2, 6, 7, 8

[49] Tianning Yuan, Fang Wan, Mengying Fu, Jianzhuang Liu, Songcen Xu, Xiangyang Ji, and Qixiang Ye. Multiple instance active learning for object detection. In *CVPR*, 2021. 2

[50] Bolei Zhou, Hang Zhao, Xavier Puig, Sanja Fidler, Adela Barriuso, and Antonio Torralba. Scene parsing through ade20k dataset. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 633–641, 2017. 2, 6

[51] Jinghao Zhou, Chen Wei, Huiyu Wang, Wei Shen, Cihang Xie, Alan Yuille, and Tao Kong. ibot: Image bert pre-training with online tokenizer. *arXiv preprint arXiv:2111.07832*, 2021. 8

[52] Jia-Jie Zhu and José Bento. Generative adversarial active learning. *arXiv preprint arXiv:1702.07956*, 2017. 2