Boundary Matters: A Bi-Level Active Finetuning Method

Han Lu[1], **Yichen Xie**[2], **Xiaokang Yang**[1], **Junchi Yan**[1,‡]
[1] Dept. of CSE & School of AI & Moe Key Lab of AI, Shanghai Jiao Tong University
[2]University of California, Berkeley
{sjtu_luhan, xkyang, yanjunchi}@sjtu.edu.cn, yichen_xie@berkeley.edu
https://github.com/Thinklab-SJTU/BiLAF

Abstract

The pretraining-finetuning paradigm has gained widespread adoption in vision tasks and other fields. However, the finetuning phase still requires high-quality annotated samples. To overcome this challenge, the concept of active finetuning has emerged, aiming to select the most appropriate samples for model finetuning within a limited budget. Existing active learning methods struggle in this scenario due to their inherent bias in batch selection. Meanwhile, the recent active finetuning approach focuses solely on global distribution alignment but neglects the contributions of samples to local boundaries. Therefore, we propose a Bi-Level Active Finetuning framework (BiLAF) to select the samples for annotation in one shot, encompassing two stages: core sample selection for global diversity and boundary sample selection for local decision uncertainty. Without the need of ground-truth labels, our method can successfully identify pseudo-class centers, apply a novel denoising technique, and iteratively select boundary samples with designed evaluation metric. Extensive experiments provide qualitative and quantitative evidence of our method's superior efficacy, consistently outperforming the existing baselines.

1 Introduction

The advancement of deep learning significantly relies on extensive training data. However, annotating large-scale datasets is challenging, requiring significant human labor and resources. To address this challenge, the pretraining-finetuning paradigm has gained widespread adoption. In this paradigm, models are first pretrained in an unsupervised manner on large datasets and then finetuned on a smaller, labeled subset. While there is substantial research about both unsupervised pretraining [7, 11, 13, 17] and supervised finetuning [15, 27], the optimization of sample set selection for annotation has received less attention, especially in scenarios with limited labeling resources.

Active learning methods [33, 36, 31], though effective in identifying valuable samples for training from scratch, face significant challenges when integrated into the pretraining-finetuning framework [3, 41]. The primary limitation stems from the batch-selection strategy commonly used by these methods. Allocating a limited annotation budget across multiple iterations can introduce harmful biases, which leads to overfitting. Consequently, this undermines the general representational quality of the pretrained model and leads to the accumulation of errors in the iterative selection process.

To fill in the research gap, the *Active Finetuning* task has been formulated in [41], which focuses on the selection of samples for supervised finetuning using pretrained models. This method optimizes sample selection by minimizing the distributional gap between the selected subset and the entire data pool.

[‡]Corresponding author. This work was supported by NSFC (92370201, 62222607) and Shanghai Municipal Science and Technology Major Project under Grant 2021SHZDZX0102.

38th Conference on Neural Information Processing Systems (NeurIPS 2024).

https://doi.org/10.52202/079017-1133

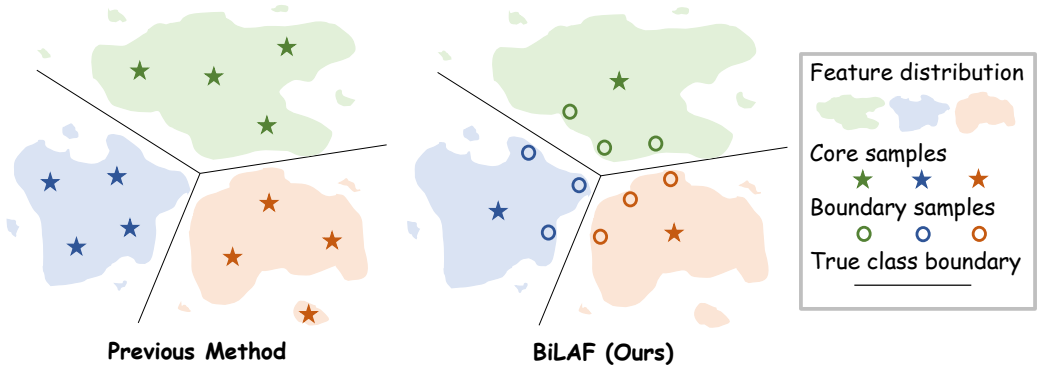


Figure 1: **Design philosophy of our BiLAF framework.** In contrast to the previous method, our method ensures the selection of central samples to maintain diversity while also reserving capacity to choose boundary samples to enhance decision boundary learning.

Despite its notable performance, it fundamentally concentrates on the global diversity of selected samples, which neglects the local decision boundaries—also referred to as sample uncertainty in data selection. As the volume of data increases, we found the model's capabilities drops dramatically, which indicates the selected samples become increasingly redundant and less informative.

To mitigate the inherent limitations of existing approaches, we introduce the innovative **Bi-Level Active Finetuning Framework (BiLAF)**, which can effectively capture both diversity and uncertainty of selected samples, as depicted in Fig. 1. The key challenge lies in measuring uncertainty within the pretraining-finetuning paradigm. Without labels and a trained classifier head, traditional methods based on posterior probability [24, 39], entropy [18, 29] and loss metrics [16, 45] are infeasible. Notably, we observe that the global feature space of pretrained models inherently captures the interrelations among samples from different classes. Consequently, we can effectively utilize their feature outputs to facilitate the identification of support samples that are proximate to the model's decision boundary. This concept has been extensively supported by theoretical [23] and methodological research [4, 5] across various domains.

To elucidate, our BiLAF framework operates through two distinct stages. The initial phase, *Core Samples Selection*, is dedicated to identifying pivotal samples for each class. The selection mechanism employed can vary, including options such as K-Means or ActiveFT [41]. The second stage, *Boundary Samples Selection*, begins with our innovative unsupervised denoising technique that precisely isolates noisy outliers. Following this, we systematically identify boundary samples adjacent to each core sample and efficiently eliminate redundant samples, employing our newly proposed boundary score.

In conclusion, our contributions are summarized as follows:

- We propose a Bi-Level Active Finetuning Framework (BiLAF) emphasizing boundary importance to balance sample diversity and uncertainty. This framework exhibits remarkable flexibility and can accommodate various methods seamlessly.
- The proposed unsupervised denoising technique in BiLAF can effectively eliminate the outlier samples and an iterative strategy with newly designed metric can identify the marginal boundary samples in feature space. Compared to other methods, our approach has high accuracy and efficiency.
- Extensive experiments and ablation studies demonstrate the effectiveness of our method. Compared to the current state-of-the-art approach, our method achieves a remarkable improvement of nearly 3% absolute accuracy on CIFAR100 and approximately 1% on ImageNet. What's more, it outperforms the other baselines in object detection, semantic segmentation, and the long-tail tasks.

2 Related Work

2.1 Active Learning / Finetuning

Active learning maximizes annotation efficiency by selecting the most informative samples. Typically, uncertainty-driven methods select difficult samples based on heuristics like posterior probability [24,

https://doi.org/10.52202/079017-1133

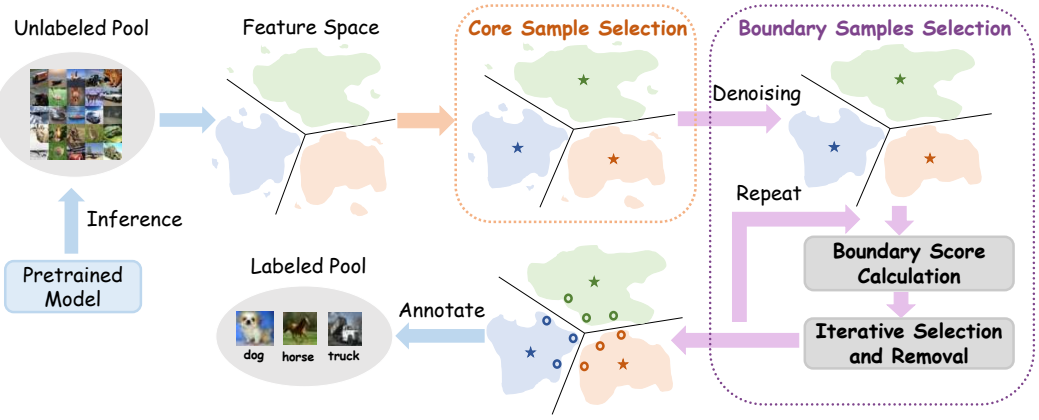


Figure 2: **Our BiLAF framework in the Active Finetuning task.** In the high-dimensional feature space, the Core Sample Selection focuses on pinpointing pseudo-class centers. Following this, we have devised a denoising method to eliminate noise samples. Subsequently, we compute the Boundary Score metric for each sample, which aids in the iterative selection of samples and the removal of candidates from the pool. Ultimately, the selected samples are labeled for supervised finetuning.

39], entropy [18, 29], or loss metrics [16, 45], while diversity-based approaches approximate the original data distribution using metrics like sample distance or gradient directions [33, 1, 40, 2, 20, 36]. However, these methods struggle within the pretraining-finetuning paradigm [12, 41]. ActiveFT [41] addresses this by aligning the sample distribution with the unlabeled pool, often prioritizing high-density over boundary areas. Our algorithm, by integrating both diversity and uncertainty, refines the selection process to enhance decision boundary samples selection for following supervised finetuning.

2.2 Decision Boundaries in Neural Networks.

Decision boundaries are pivotal in neural network-based classifiers, influencing both performance and interpretability [23, 25]. Their optimization enhances generalizability and accuracy in complex data spaces [42, 5]. In SVMs, decision boundaries separate hyperplanes and maximize geometric margins for robust classification [4]. This concept extends to neural networks, where margin maximization also promotes generalization. For imbalanced datasets, adjusting the decision boundary is crucial for accurate minority class classification, with approaches like LDAM loss [5] and ELM loss [19] developed to modify the boundary and balance errors across classes. Neural networks tend to utilize simple or highly discriminative features for decision boundaries [30, 34]. A theoretical framework exploring decision boundary complexity and its inverse relation to generalizability is introduced in [23]. Notably, this is not trivial in unsupervised scenarios. Despite the challenge, we effectively leverage features from pretrained models, thereby introducing innovative denoising and selection methods without labels. This approach addresses a significant gap in the active finetuning domain.

3 BiLAF: Bi-Level Active Finetuning

In this section, we introduce our novel **Bi**-**L**evel **A**ctive **F**inetuning Framework (**BiLAF**), as illustrated in Fig. 2. BiLAF operates in two distinct stages: Firstly, *Core Samples Selection* (Sec. 3.2) involves identifying multiple pseudo-class centers, which ensures comprehensive coverage across all classes. Secondly, *Boundary Samples Selection* (Sec. 3.3) focuses on the accurate identification of boundary samples for each pseudo-class, using the novel denoising and iterative selection method. Algorithm 1 summarizes the complete workflow. The theoretical time complexity is analyzed in Appendix D.

3.1 Preliminary: Active Finetuning Task

The active finetuning task is defined in [41]. Apart from a large unlabeled data pool $\mathcal{P}^u = \{\mathbf{x}_i\}_{i\in[N]}$ where $[N] = \{1, 2, \ldots, N\}$ identical to the traditional active learning task, we have the access to a deep neural network model $f(\cdot; w_0)$ with the pretrained weight w_0. It should be noted that the function $f(\cdot; w_0)$ can be pretrained either on the current data pool $\mathcal{P}^u$ or any other external data

https://doi.org/10.52202/079017-1133

sources. Using the pretrained model, we can map the data sample $\mathbf{x}_i$ to the high dimensional feature space as $\mathbf{f}_i = f(\mathbf{x}_i; w_0) \in \mathbb{R}^d$, where $\mathbf{f}_i$ is the *normalized* feature of $\mathbf{x}_i$. From this, we can derive the feature pool $\mathcal{F}^u = \{\mathbf{f}_i\}_{i\in[N]}$ from $\mathcal{P}^u$, which assists us in selecting the optimal samples.

Our task is to select the subset $\mathcal{P}^u_{\mathcal{S}}$ from $\mathcal{P}^u$ for annotation and the subsequent supervised finetuning. The subset $\mathcal{P}^u_{\mathcal{S}} = \{\mathbf{x}_{s_j}\}_{j\in[B]}$ is determined by the sampling strategy $\mathcal{S} = \{s_j \in [N]\}_{j\in[B]}$, where B represents the annotation budget. The labels $\{\mathbf{y}_{s_j}\}_{j\in[B]} \subset \mathcal{Y}$, a subset of the label space $\mathcal{Y}$, are accessed through an oracle, resulting in a labeled data pool $\mathcal{P}^l_{\mathcal{S}} = \{\mathbf{x}_{s_j}, \mathbf{y}_{s_j}\}_{j\in[B]}$. Subsequently, the pretrained model $f(\cdot; w_0)$ undergoes supervised finetuning on $\mathcal{P}^l_{\mathcal{S}}$. Our objective is to optimize the sampling strategy $\mathcal{S}$ to select the labeled set that minimizes the expected error of the finetuned model under the given annotation budget constraints.

3.2 Core Samples Selection

We start data selection procedure by selecting core samples. Those core samples can represent the distribution of the entire dataset. Popular methods include K-Means, Coreset [33], and ActiveFT [41], differing in their design targets and optimization procedures. We employ ActiveFT, the most advanced method to date, as our primary approach in the initial phase. For detailed implementation of ActiveFT, see Appendix B. Using ActiveFT, we obtain K core samples, which serve as the center for each pseudo-class i, where K represents the predefined budget for core samples.

3.3 Boundary Samples Selection

In addition to those K core samples, we continue to select samples close to the boundaries of semantic categories, which can enhance the training of the model's decision boundaries.

By leveraging pretrained models, data samples are mapped to robust feature representations that elucidate the relationships among samples, their intra-class counterparts, and inter-class samples from diverse classes. We introduce a novel method for boundary sample selection— the first in this field. Given the pseudo-class centers, our method involves three steps: *1) Identifying the samples associated with each pseudo-class center; 2) Implementing denoising processes to refine these samples; 3) Developing precise metrics to select boundary samples for each pseudo-class*.

3.3.1 Sample Clustering

Given K pseudo-class centers denoted by $C = \{c_1, c_2, \ldots, c_K\}$ where $c_i \in [N]$, a sample $\mathbf{x}_j$ with its feature vector $\mathbf{f}_j \in \mathbb{R}^d$ is assigned to the pseudo-class center c_i that minimizes the distance $D(\mathbf{f}_j, \mathbf{f}_{c_i})$. This assignment can be mathematically represented as:

$$c_i = \arg\min_{c\in C} D(\mathbf{f}_j, \mathbf{f}_c) \tag{1}$$

where $D(\cdot, \cdot)$ denotes the distance function. While the choice of D can vary based on the design of the clustering model, we employ the Euclidean distance in our implementation, which efficiently aligns each sample point with a corresponding pseudo-class center during the optimization process.

3.3.2 Boundary Sample Denoising

Boundary samples within a pseudo-class are crucial for optimizing decision boundaries. However, they can also introduce noise, which potentially hinders the model's performance. For each class i, with N_i samples, we define a removal ratio P_{rm} to identify and eliminate $N_{i,rm} = N_i \cdot P_{rm}$ peripheral noisy samples from the candidate boundary set. This elimination process is based on the density of samples in the feature space, where we define the concept of density distance as follows:

Definition 1 (Density Distance)**.** *The density distance of a sample is defined as the average distance to its k nearest neighbors. Formally, for a given sample x_j characterized by its feature vector $\mathbf{f}_j$, its density distance $\rho(\mathbf{x}_j)$ is defined as:*

$$\rho(\mathbf{x}_j) = \frac{1}{k}\sum_{l=1}^{k} D(\mathbf{f}_j, \mathbf{f}_{n_{jl}}), \tag{2}$$

where n_{jl} represents the index of the l-th nearest neighbor of x_j in the feature space, and D denotes a distance function.

Inspired by the classical clustering method DBSCAN [8], we propose an Iterative Density-based Clustering (IDC) algorithm. IDC clusters the candidate samples $\mathbf{X}_i = \{\mathbf{x}_{a_{i,1}}, \mathbf{x}_{a_{i,2}}, ...\mathbf{x}_{a_{i,N_i}}\}$ where the index $a_{i,j} \in [N]$ and $j \in [N_i]$. The algorithm initiates with the core sample index c_i as the seed set $U_i = \{c_i\}$. In each subsequent iteration, a fraction P_{in} of samples are included, which corresponds to $N_{i,in} = N_i \cdot P_{in}$ peripheral samples being integrated into the existing cluster U_i.

The density distance of each candidate sample is redefined from Eq. 2 as the average distance to the nearest k selected sample in the cluster U_i. Specifically, the density distance $\rho(\mathbf{x}_j)$ for each remaining point $\mathbf{x}_j$ is calculated as:

$$\rho(\mathbf{x}_j) = \frac{1}{k} \sum_{l=1, n_{jl} \in U_i}^{k} D(\mathbf{f}_j, \mathbf{f}_{n_{jl}}), \qquad (3)$$

where n_{jl} represents the index of the l-th nearest sample in U_i to $\mathbf{x}_j$. In each iteration, the $N_{i,in}$ samples with the lowest density distance are selected. This process repeats until all samples have been included. The order of inclusion into the cluster reflects each point's proximity to the center, with later-added samples more likely to be peripheral and noisy. Consequently, the last $N_{i,rm}$ samples can be removed from U_i.

3.3.3 Iterative Sample Selection

After successfully eliminating noise samples, we advance to the selection of boundary samples from the centers of pseudo-classes. To streamline this process, we introduce a set of definitions that facilitate the computation of a boundary score for each sample.

Definition 2 (Intra-class Distance)**.** *Let $\mathbf{x}_j$ be a sample in the set U_i of the pseudo-class c_i. The intra-class distance for $\mathbf{x}_j$ is the average distance from $\mathbf{f}_j$ to all other samples within the same class set U_i. It can be formally defined as:*

$$d_{intra}(\mathbf{x}_j) = \frac{1}{|U_i|} \sum_{l \in U_i} D(\mathbf{f}_j, \mathbf{f}_l) \qquad (4)$$

where D is the distance function and $|U_i|$ is the number of samples in pseudo-class c_i.

Algorithm 1 Pseudo-code for BiLAF

Input: Unlabeled data pool $\mathcal{P}^u = \{\mathbf{x}_i\}_{i \in [N]}$, pretrained model $f(\cdot; w_0)$, annotation budget B.
Parameter: Core samples budget K, removal ratio P_{rm}, density neighbours k, IDC cluster ratio P_{in}, opponent penalty coefficient δ, distance function $D(\cdot, \cdot)$.
Output: The selected samples index $\mathcal{S} = \{s_j\}_{j \in [B]}$

▷ *Stage 1: Core Samples Selection*
for $i \in [N]$ **do**
  $\mathbf{f}_i = f(\mathbf{x}_i; w_0)$
Select K centers in feature space $\mathcal{F}$ using ActiveFT
  ▷ *Can be K-Means and any other core selection method*
▷ *Stage 2: Boundary Samples Selection*
for each pseudo-class c_i **do**
  ▷ *Step1: Assign the pseudo-class samples*
  Initial the candidate index set $U_i^0 = \{\}$
  for $j \in [N]$ **do**
    if $c_i == \arg\min_{c \in C} D(\mathbf{f}_j, \mathbf{f}_c)$ **then**
      Add the index j to the set U_i^0
  ▷ *Step2: Iterative Density-based cluster then denoise*
  Initialize the pseudo-class cluster $U_i = \{c_i\}$.
  while $|U_i^0| > |U_i|$ **do**
    for each candidate index $j \in U_i^0$ and $j \notin U_i$ **do**
      $\rho(\mathbf{x}_j) = \frac{1}{k} \sum_{l=1, \mathbf{n_{jl}} \in U_i}^{k} D(\mathbf{f}_j, \mathbf{f}_{n_{jl}})$ where n_{jl} is the nearest l-th sample index of $\mathbf{x}_j$.
    Sort samples by increasing density distance $\rho(\mathbf{x}_j)$
    Add the previous $P_{in} \cdot |U_i^0|$ samples to U_i
  Remove the last $P_{rm} \cdot |U_i|$ samples from U_i to denoise
  ▷ *Step3: Selection Process*
  Initialize opponent penalty counts $t_l = 0$ for $l \neq i$
  for $m = 1$ to B_i **do**
    for each candidate index $j \in U_i$ **do**
      $d_{intra}(\mathbf{x}_j) = \frac{1}{|U_i|} \sum_{l \in U_i} D(\mathbf{f}_j, \mathbf{f}_l)$
      $BS(\mathbf{x}_j) = \min_{c_l \in C}^{i \neq l} \frac{\delta^{t_l} \cdot D(\mathbf{f}_j, \mathbf{f}_{c_l}) - d_{intra}(\mathbf{x}_j)}{\max(D(\mathbf{f}_j, \mathbf{f}_{c_l}), d_{intra}(\mathbf{x}_j))}$
    Select $\mathbf{x}_{j_{min}}$ with the lowest Boundary Score if $m > 1$, otherwise set $j_{min} = c_i$.
    Add the index j_{min} to selected samples set $\mathcal{S}$
    $t_l = t_l + 1$ where nearest opponent class is l if $m > 1$.
    Remove the nearest $|U_i|/B_i$ samples of $\mathbf{x}_{j_{min}}$ from U_i
return the selected samples index $\mathcal{S}$

Definition 3 (Inter-class Distance)**.** *The inter-class distance for a sample $\mathbf{x}_j$ in the pseudo-class c_i is defined as the distance from $\mathbf{f}_j$ to the nearest center of other pseudo-classes. It is defined as:*

$$d_{inter}(\mathbf{x}_j) = \min_{c_l \in C, i \neq l} D(\mathbf{f}_j, \mathbf{f}_{c_l}) \qquad (5)$$

where D is the distance function, $C = \{c_1, \ldots, c_K\}$ are the pseudo-class centers, and $i \neq l$ ensures that the pseudo-class c_i of the sample $\mathbf{x}_j$ is excluded from the calculation.

Definition 4 (Boundary Score)**.** *The Boundary Score for sample $\mathbf{x}_j$ in psuedo-class c_i can be defined as a function of both intra-class and inter-class distances. It is defined as:*

$$BS(\mathbf{x}_j) = \frac{d_{inter}(\mathbf{x}_j) - d_{intra}(\mathbf{x}_j)}{\max(d_{inter}(\mathbf{x}_j), d_{intra}(\mathbf{x}_j))} \qquad (6)$$

where d_{intra} and d_{inter} are the intra-class and inter-class distances, respectively. A smaller Boundary Score indicates closer proximity to the boundary.

https://doi.org/10.52202/079017-1133

For each pseudo-class c_i with the samples set U_i, we allocate the sample selection budget in proportion to the size of the pseudo-class. Specifically, the budget B_i for pseudo-class c_i is calculated as: $B_i = B \cdot |U_i| \,/\, \sum_{j=1}^{K} |U_j|$ where $|U_i|$ represents the number of samples in pseudo-class c_i.

Initially, one might consider simply selecting the top B_i samples with the lowest boundary scores. However, this approach risks over-concentration of selections within specific areas of the feature space. To address this, we employ an **iterative selection and removal** strategy, which progressively selects and eliminates candidate samples. Furthermore, to prevent the aggregation of multiple samples near the same pseudo-class boundary that faces the same opposing pseudo-class center, we introduce an **opponent penalty**. This mechanism increases the influence of previously selected boundary samples on subsequent selections, encouraging greater diversity across different boundaries.

Iterative selection and removal begins by selecting samples with the lowest Boundary Score in each iteration. Subsequently, the nearest $\lfloor |U_i|/B_i \rfloor$ samples surrounding the selected sample are removed to prevent clustering. This process is repeated B_i times, starting from the center samples.

Opponent penalty monitors the relationship between the currently selected samples and the boundaries to other pseudo-classes, imposing a penalty on opponent pseudo-classes whose boundaries have already been selected. Specifically, for the selected pool of pseudo-class c_i, if boundary samples related to an opposing pseudo-class $l \neq i$ have been selected t_l times, the inter-class distance for the subsequent samples to this pseudo-class l will be scaled by a factor of δ^{t_l}, where the opponent penalty coefficient δ is a hyperparameter greater than 1. This indicates that the more often a sample is selected in relation to pseudo-class l's boundary, the higher its Boundary Score with that pseudo-class becomes, making it less likely to be chosen in future rounds. Consequently, we recalibrate the Boundary Score for each sample in each iteration, leading to a modification from Eq. 6:

$$BS(\mathbf{x}_j) = \min_{c_l \in C, i \neq l} \frac{\delta^{t_l} D(\mathbf{f}_j, \mathbf{f}_{c_l}) - d_{intra}(\mathbf{x}_j)}{\max(D(\mathbf{f}_j, \mathbf{f}_{c_l}), d_{intra}(\mathbf{x}_j))} \tag{7}$$

where $D(\mathbf{f}_j, \mathbf{f}_{c_l})$ measures the distance from sample $\mathbf{x}_j$ to the opponent pseudo-class l and the δ^{t_l} reflects the imposed penalty.

In each pseudo-class i, we select the B_i samples through an iterative selection and removal strategy that incorporates this opponent penalty in the increasing order of $BS(\mathbf{x}_j)$. The selected samples from various pseudo-classes are ultimately aggregated to form the comprehensive annotation subset $\mathcal{P}^u_{\mathcal{S}}$.

4 Experiments

Our approach has been rigorously evaluated using three primary image classification benchmarks, alongside various tasks with different sampling ratios, as detailed in Sec. 4.1. We compare its performance against multiple baselines and conventional active learning techniques, with results discussed in Sec. 4.2. The analysis of our method is presented in Sec. 4.3, and a comprehensive ablation study is provided in Sec. 4.4. All experiments were conducted using GeForce RTX 3090(24G) GPUs and Intel(R) Core(TM) i9-10920X CPUs. The source code will be made publicly available.

4.1 Experiment Setup

Datasets and Evaluation Metrics. Firstly, we evaluate our method using three widely recognized classification datasets: CIFAR10, CIFAR100 [22], and ImageNet-1k [32]. The raw training data from these datasets constitute the unlabeled pool $\mathcal{P}^u$ from which selections are made. For performance evaluation, we employ the *Top-1 Accuracy* metric. In addition to these classification tasks, extensive experiments have also been conducted in object detection, semantic segmentation, and long-tail tasks. Further details on these experiments can be found in Appendix E and F.

Baselines. We compare our approach with three heuristic baselines Random, FDS and K-Means; five active learning methods CoreSet [33], VAAL [36], LearnLoss [45], TA-VAAL [20], and ALFA-Mix [31]; and the well-designed active finetuning method ActiveFT [41]. We utilize the overlapping results reported in [41]. The detailed information of the baselines is listed in Appendix E.

Implementation Details. In line with the SOTA method ActiveFT [41], we use DeiT-Small [38] model, pretrained using the DINO [6] framework on ImageNet-1k in the unsupervised pretraining phase. For all the datasets, we resize images to 224×224 consistent with the pretraining for both

Table 1: **Benchmark Results.** Experiments are conducted on three popular datasets with different annotation ratios. We report the mean and standard deviation over three trials. Traditional active learning methods require random initial data to start, thus we use "-" to represent. BiLAF has shown a significant competitive advantage across the majority of scenarios, affirming its effectiveness.

Methods	CIFAR10			CIFAR100				ImageNet		
	0.5%	1%	2%	1%	2%	5%	10%	1%	2%	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	52.1±0.6	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	43.1±0.4	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	49.8±0.5	55.4±0.3	64.0±0.2
CoreSet [33]	-	81.6±0.3	88.4±0.2	-	30.6±0.4	48.3±0.5	62.9±0.6	-	52.7±0.4	61.7±0.2
VAAL [36]	-	80.9±0.5	88.8±0.3	-	24.6±1.1	46.4±0.8	70.1±0.4	-	54.7±0.5	64.0±0.3
LearnLoss [45]	-	81.6±0.6	86.7±0.4	-	19.2±2.2	38.2±2.8	65.7±1.1	-	54.3±0.6	63.2±0.4
TA-VAAL [20]	-	82.6±0.4	88.7±0.2	-	34.7±0.7	46.4±1.1	66.8±0.5	-	55.0±0.4	64.3±0.2
ALFA-Mix [31]	-	83.4±0.3	89.6±0.2	-	35.3±0.8	50.4±0.9	69.9±0.6	-	55.3±0.3	64.5±0.2
ActiveFT [41]	**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	55.8±0.3	65.3±0.1
BiLAF (ours)	81.0±1.2	**89.2**±0.6	**92.5**±0.4	**31.8**±1.6	**43.5**±0.8	**62.8**±1.2	**73.7**±0.5	**50.8**±0.4	**56.9**±0.3	**66.2**±0.2

Table 2: **Performance on Object Detection, Semantic Segmentation and Long-Tailed Dataset.**

Methods	PASCAL VOC (mAP) ↑				ADE20k (mIoU) ↑			CIFAR100LT (IR=100) ↑		
	#1000	#2000	#3000	#5000	5%	10%	15%	5%	10%	15%
Random	51.1±0.9	60.9±0.6	64.2±0.5	67.5±0.4	14.54±1.02	20.27±1.68	23.55±1.35	21.49±2.37	28.55±1.93	34.58±1.29
FDS	49.1±0.6	58.4±0.4	62.1±0.3	64.5±0.2	12.14±0.27	17.66±0.33	23.55±0.53	16.53±0.65	23.45±0.57	28.76±0.36
K-Means	52.3±0.4	60.7±0.4	64.9±0.3	67.8±0.3	13.62±0.52	19.12±0.47	23.10±0.69	22.98±0.99	29.26±0.82	34.47±0.61
Coreset [33]	-	61.5±0.3	66.0±0.3	69.0±0.2	-	20.25±0.44	23.68±0.62	-	28.37±0.71	35.12±0.55
VAAL [36]	-	61.2±0.5	65.7±0.4	68.9±0.3	-	20.54±0.75	23.87±0.88	-	29.97±0.66	35.62±0.35
LearnLoss [45]	-	60.8±0.6	64.9±0.4	68.9±0.3	-	19.71±1.05	22.94±1.19	-	28.16±1.43	34.02±1.01
TA-VAAL [20]	-	61.4±0.4	65.9±0.3	69.1±0.2	-	20.63±0.77	24.24±0.93	-	30.01±0.96	35.59±0.64
ActiveFT [41]	54.9±0.3	61.8±0.3	66.0±0.2	69.1±0.2	15.37±0.11	21.60±0.40	25.03±0.87	24.60±1.02	31.58±0.76	37.01±0.63
BiLAF(Ours)	**55.7**±0.5	**62.3**±0.3	**66.3**±0.3	**69.2**±0.2	**16.54**±0.37	**22.03**±0.53	**25.71**±0.79	**26.56**±1.18	**32.35**±0.81	**37.33**±0.71

data selection and supervised finetuning. In the core samples selection stage, we utilize ActiveFT and optimize the parameters θ_S using the Adam [21] optimizer (learning rate 1e-3) until convergence. We set the core number K as $50(0.1\%), 250(0.5\%), 6405(0.5\%)$ for CIFAR10, CIFAR100 and ImageNet separately. In the boundary samples selection stage, we consistently set nearest neighbors number k as 10, both removal ratio P_{rm} and clustering fraction P_{in} as 10%, opponent penalty coefficient δ as 1.1. The experiment details of supervised finetuning are listed in the Appendix E.

4.2 Overall Peformance Comparison

The average performance and standard deviation from three independent runs are presented in Tab. 1. Under a low sampling ratio of 0.5% for CIFAR10, our method is marginally outperformed by ActiveFT. This can be attributed to the more stable model training aided by core point selection at extremely low budgets, whereas boundary samples tend to introduce greater instability and perturbation. However, as the volume of data increases, the advantages of constructing precise boundaries become more pronounced. Our approach BiLAF exhibits significant superiority across most scenarios, markedly outperforming competing methods. Notably, on CIFAR100, BiLAF consistently outperforms the previously best-performing model, ActiveFT, by approximately 3%. On ImageNet, BiLAF achieves a consistent improvement of about 1%. These significant enhancements underscore the effectiveness of the BiLAF method. Further details on the robust performance of our method across various pretraining paradigms and model architectures, which substantiate both its effectiveness and generality, are provided in Appendix F.

Tab. 2 presents the performance of our method across diverse tasks and scenarios. Although our design, based on boundary sample selection, appears to be tailored for classification tasks, the performance of BiLAF still effectively surpasses that of other models in these domains. We attribute this success to our effective denoising of outlier samples and the method's focus on selected sample uncertainty. The less pronounced advantage in certain tasks may be due to global features not fully representing task requirements such as in object detection involving multiple objects, and the default denoising removal ratio parameters might remove minority samples in long-tail distributions. Despite these challenges, BiLAF demonstrates robust performance across various scenarios, confirming its

https://doi.org/10.52202/079017-1133

Table 4: **Ablation on CIFAR100.** In the denoising process,"DG", "DB" and "IDC" indicate the basic distance-guide method, density-based method and iterative density-based clustering method. In the selection criterion,"BD", "BS" indicate the basic distance metric and the boundary score metric. In the selection process,"OS", "ISR", "OP" indicate selecting the top samples in one shot, iterative selection and removal and whether to use opponent penalty. BiLAF represents the complete implementation and we explore the influence of three designs separately.

ID	Denoising Process			Selection Criterion		Selection Process			Annotation budget			
	DG	DB	IDC	BD	BS	OS	ISR	OP	1%	2%	5%	10%
BiLAF	-	-	✓	-	✓	-	✓	✓	**31.82**	**43.48**	**62.75**	**73.67**
1	-	✓	-	-	✓	-	✓	✓	31.10 (−0.72)	41.06 (−2.42)	61.60 (−1.15)	73.39 (−0.28)
2	✓	-	-	-	✓	-	✓	✓	29.27 (−2.55)	36.42 (−7.06)	61.16 (−1.59)	72.87 (−0.80)
3	-	-	-	-	✓	-	✓	✓	28.88 (−2.94)	36.78 (−6.70)	60.83 (−1.92)	72.76 (−0.91)
4	-	-	✓	✓	-	-	✓	✓	27.94 (−3.88)	33.76 (−9.72)	54.03 (−8.72)	71.17 (−2.50)
5	-	-	✓	-	✓	-	✓	-	**32.13** (+0.31)	43.09 (−0.39)	62.07 (−0.68)	72.92 (−0.75)
6	-	-	✓	-	✓	✓	-	-	30.52 (−1.30)	42.13 (−1.35)	60.24 (−2.51)	69.66 (−4.01)

strength as a general sample selection method. As a flexible framework, there is still considerable scope for improvement in our method by further designing it based on task-specific features.

4.3 Analysis

Data Selection Efficiency. Traditional active learning methods typically follow a paradigm involving multiple iterations of selection and training. However, this approach not only demands substantial computational resources due to repeated training but also results in prolonged data selection times and increased management costs associated with multiple annotation processes in practical scenarios. In contrast, our method, alongside ActiveFT [41], adopts a one-shot selection strategy, offering distinct advantages. Tab. 3 presents a comparison of the time spent in selecting varying proportions of data across different methods on CIFAR100. For a deeper understanding of the theoretical time complexity and further analysis, please refer to Appendix D.

Table 3: **Time compelxity.** Our method exhibits significant advantages beyond conventional active learning approaches and comparable speed to ActiveFT.

Ratio	K-Means	CoreSet	VAAL	LearnLoss	ActiveFT	BiLAF(ours)
2%	16.6s	1h57m	7h52m	20m	**12.6s**	18.6s
5%	37.0s	7h44m	12h13m	1h37m	21.9s	**19.2s**
10%	70.2s	20h38m	36h24m	9h09m	37.3s	**20.3s**

Selected Samples Visualization. Fig. 3 illustrates the sample selection process of our method. Firstly, we identify the central samples represented by pentagrams, and then expand to the boundary points denoted by circles from each center. Our method focuses on the boundaries between two categories rather than purely fitting the entire distribution. This approach allows for the selection of more valuable samples. For a more detailed comparison with other methods and the visualization of denoising process, please refer to the Appendix J.

4.4 Ablation Study

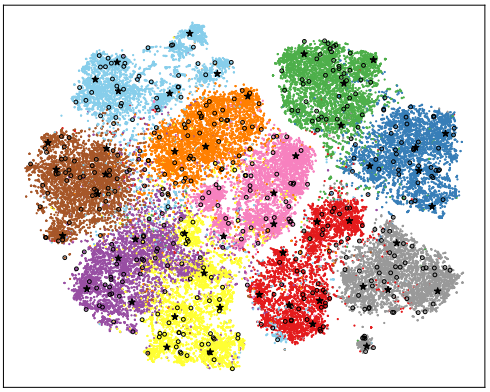


Figure 3: **tSNE Embeddings on CIFAR10 with 1% annotation budget of BiLAF.** Pentagrams represent the chosen core samples, while circles denote the chosen boundary samples.

Effectiveness of Designs. Tab. 4 demonstrates the contributions of all proposed components outlined in Sec. 3.3 to the model performance. Our framework is structured around three core components: Denoising Process, Selection Criterion, and Selection Process. IDs 1 to 3 evaluate the impact of the Denoising Process, comparing iterative density-based clustering with other strategies detailed in Appendix C, and scenarios without denoising. We observe that performance degradation is substantial under smaller budgets but diminishes as the budget increases. This suggests that with fewer data selected, the adverse effects of noisy samples are more pronounced; however, with an increased data volume, these deficiencies are mitigated, underscoring the efficacy of our denoising strategy. ID 4 explores the impact of our Selection Criterion design. In the absence of the Boundary Score metric,

https://doi.org/10.52202/079017-1133

models struggle to accurately identify potential boundary samples and are negatively impacted by irrelevant marginal samples, significantly reducing performance. IDs 5 and 6 assess the effects of our Selection Process design. While removing the opponent penalty does not critically undermine performance and might even enhance it at a 1% data volume, iterative selection and removal are essential, particularly as data volume increases. These steps effectively increase the diversity of boundary samples selected. Conversely, a one-time selection approach at increased data volumes leads to the accumulation of redundant samples, resulting in wastage and performance degradation.

Core Samples Selection Number. Without ground truth labels, ensuring that each pseudo-class center selected during the core sample selection stage of our BiLAF method represents a distinct category is challenging. Tab. 5 illustrates the impact of varying the number of core samples on accuracy across different annotation budgets within the CIFAR100 dataset. We found that performance significantly suffers when fewer centers are selected. For instance, selecting 125 centers for 100 categories resulted in suboptimal performance, primarily due to the limited number of centers being unable to represent all categories adequately. This limitation poses significant challenges for subsequent boundary sample selection. However, performance stabilizes once a sufficient number of centers to encompass all categories is established. Optimizing the ratio of center samples to boundary samples can yield the performance gains. For example, the best performance at 5% data volume was achieved with 375 centers, and at 10% data volume, 500 centers were optimal. Although there is potential to further enhance our method, we maintained the number of centers across all experiments with different annotation ratio for consistency.

Table 5: **Ablation for Core Samples Numbers.**

Budget	Core Ratio / Core Number			
	0.25% / 125	0.50% / 250	0.75% / 375	1.00% / 500
1%	21.51	**31.82**	28.37	27.24
2%	36.64	**43.48**	42.68	42.18
5%	59.20	62.75	**63.32**	62.46
10%	71.86	73.67	73.58	**74.32**

Table 6: **Ablation for Core Selection Method.**

Budget	Core Selection Method			
	Random	FDS	K-Means	ActiveFT
1%	25.58	20.69	28.80	**31.82**
2%	36.22	33.22	41.17	**43.48**
5%	60.68	60.13	62.39	**62.75**
10%	72.84	71.05	**74.27**	73.67

Core Samples Selection Method. In the core sample selection, we primarily utilized ActiveFT. However, there are numerous existing methods, such as Random, FDS, and K-Means. Tab. 6 presents the model performance based on boundary selection using different pseudo-class centers. We found that the accuracy of the BiLAF framework is closely tied to the quality of the method used for selecting pseudo-class centers. Mis-selection of centers can introduce significant bias, adversely affecting subsequent sample selection. Notably, ActiveFT tended to yield the highest performance, while the traditional K-Means method also demonstrated strong results and outperform ActiveFT even in 10% budget, validating the robustness of our framework with well-defined centers.

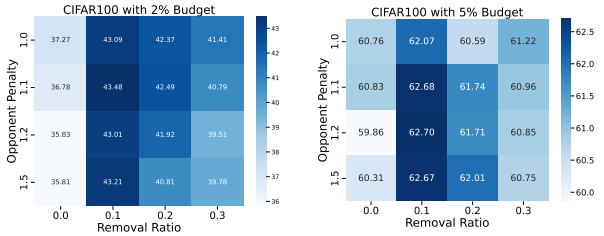


Figure 4: **The Hyperparameter Influence.**

Hyperparameter Influence. Our primary handcrafted parameters include the removal rate of noise samples during the denoising process and the opponent penalty applied during the selection process to penalize boundaries within the same class. Fig. 4 examines the impact of varying these parameters on our model's performance on the CIFAR100 dataset, with annotation budgets of 2% and 5%. We observe that the removal rate significantly influences performance. A low setting allows excessive noise accumulation, while a high setting depletes boundary sample points as the budget increases, adversely affecting performance. In contrast, the opponent penalty exerts a more subtle effect and can modestly enhance model performance.

Threshold of Core Numbers. Tab. 5 illustrates the impact of different Core Numbers on performance. In practical applications, determining the optimal Core Number directly is a question worth exploring. Similarly, in traditional active learning, the effectiveness of different methods often varies with the scale of the data. The classic Coreset [33] seeks to cover all samples using selected points. However, ProbCover [44] highlights that Coreset struggles when the dataset is very small, prompting the introduction of a coverage radius to ensure each selected point influences a fixed area.

https://doi.org/10.52202/079017-1133

Table 7: **Threshold of Core Numbers.** Distance and Rate of Return on CIFAR10 and CIFAR100.

Core Num	50	100	150	250	375	500	1000	1500	2500	5000
CIFAR10										
Distance	0.7821	0.7588	0.7472	0.7307	0.7203	0.7117	0.6896	0.6746	0.6506	0.6010
Rate of Return $\times 10^{-4}$	-	4.6575	2.3264	1.6422	0.8362	0.6887	0.4420	0.3002	0.2398	0.1984
CIFAR100										
Distance	0.8378	0.8082	0.7913	0.7724	0.7564	0.7478	0.7229	0.7059	0.6800	0.6272
Rate of Return $\times 10^{-4}$	-	5.9221	3.3791	1.8906	1.2797	0.6845	0.4985	0.3404	0.2589	0.2112

In our case, using central points to approximate dense distributions seems more appropriate, especially when the budget is highly constrained, without selecting additional boundary samples. However, as the budget increases, the selection of boundary samples becomes more meaningful. To further investigate, we explore what an appropriate threshold might look like. We employed the Core Sample Selection method to derive different numbers of central points and analyzed their benefits. Specifically, we define *Distance* as the average Euclidean distance from each sample to its nearest selected sample in the feature space. Additionally, we examine the *Rate of Return* (incremental benefit per core sample) within different ranges, where *Rate of Return* = Distance Difference / Core Number Difference between two adjacent columns.

In Tab. 7, We found that the *Rate of Return* diminishes gradually, indicating that core samples are crucial in the early stages, while the benefits decrease significantly later on. A clear demarcation point can serve as a guide for when to begin Boundary Sample Selection, such as the range of 250-375 for CIFAR-10 and 375-500 for CIFAR-100. This provides a simple yet effective guideline. Additionally, in practical applications, we discovered that introducing boundary points earlier may yield better results, such as CIFAR100 with 1% (500 samples) annotation samples.

Table 8: **Performance on Different Pretraining Frameworks and Models on CIFAR10.**

Methods	DeiT-S Pretained with iBOT		ResNet50 Pretrained with DINO	
	1%	2%	1%	2%
Random	83.0	89.8	76.2	83.7
CoreSet	82.8	89.2	70.4	83.2
LearnLoss	83.6	89.2	71.7	81.3
VAAL	85.1	89.3	75.0	83.3
ActiveFT	88.3	90.9	78.6	84.9
BiLAF(Ours)	**89.1**	**92.2**	**79.3**	**85.8**

Generality on Pretraining Frameworks and Model Architectures. Our method BiLAF demonstrates versatility across various pretraining frameworks and models. BiLAF has effectively integrated with the DINO [6] framework and the DeiT-Small [38] model. Here, we apply the method to a DeiT-Small [38] trained with generative unsupervised pretraining framework iBOT [47] and CNN model ResNet50 [14] trained with DINO [6]. All models are pretrained on ImageNet-1k and finetuned on CIFAR10 with other same implementation details described in Appendix E. Tab. 8 highlights our approach's substantial improvement over other sample selection baseline across different sampling ratios, illustrating our method's broad applicability to diverse pretraining strategies and model types.

5 Conclusion

In this paper, we underscore the significance of active finetuning tasks and critically examine existing methods, which often overlook uncertainty aspects, particularly under the pretraining-finetuning paradigm. We propose an innovative solution: the Bi-Level Active Finetuning Framework (BiLAF). This framework not only ensures diversity in the selection of central points but also prioritizes boundary samples with higher uncertainty. BiLAF effectively amalgamates existing core sample selection models and introduces a novel strategy for boundary sample selection in unsupervised scenarios. Our extensive empirical studies validate BiLAF's effectiveness, demonstrating its capability to enhance predictive performance. Through comparative experiments, we explore new avenues, such as finding the optimal balance between central and boundary points. We believe our work offers valuable insights into Active Finetuning and will serve as a catalyst for further research in this field.

References

[1] S. Agarwal, H. Arora, S. Anand, and C. Arora. Contextual diversity for active learning. In *European Conference on Computer Vision*, pages 137–153. Springer, 2020.

[2] J. T. Ash, C. Zhang, A. Krishnamurthy, J. Langford, and A. Agarwal. Deep batch active learning by diverse, uncertain gradient lower bounds. *arXiv preprint arXiv:1906.03671*, 2019.

[3] J. Z. Bengar, J. van de Weijer, B. Twardowski, and B. Raducanu. Reducing label effort: Self-supervised meets active learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1631–1639, 2021.

[4] C. J. Burges. A tutorial on support vector machines for pattern recognition. *Data mining and knowledge discovery*, 2(2):121–167, 1998.

[5] K. Cao, C. Wei, A. Gaidon, N. Arechiga, and T. Ma. Learning imbalanced datasets with label-distribution-aware margin loss. *Advances in neural information processing systems*, 32, 2019.

[6] M. Caron, H. Touvron, I. Misra, H. Jégou, J. Mairal, P. Bojanowski, and A. Joulin. Emerging properties in self-supervised vision transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9650–9660, 2021.

[7] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

[8] M. Ester, H.-P. Kriegel, J. Sander, X. Xu, et al. A density-based algorithm for discovering clusters in large spatial databases with noise. In *kdd*, volume 96, pages 226–231, 1996.

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

[10] A. Freytag, E. Rodner, and J. Denzler. Selecting influential examples: Active learning with expected model output changes. In *European conference on computer vision*, pages 562–577. Springer, 2014.

[11] J.-B. Grill, F. Strub, F. Altché, C. Tallec, P. Richemond, E. Buchatskaya, C. Doersch, B. Avila Pires, Z. Guo, M. 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.

[12] G. Hacohen, A. Dekel, and D. Weinshall. Active learning on a budget: Opposite strategies suit high and low budgets. *arXiv preprint arXiv:2202.02794*, 2022.

[13] K. He, X. Chen, S. Xie, Y. Li, P. Dollár, and R. Girshick. Masked autoencoders are scalable vision learners. *arXiv preprint arXiv:2111.06377*, 2021.

[14] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[15] E. J. Hu, P. Wallis, Z. Allen-Zhu, Y. Li, S. Wang, L. Wang, W. Chen, et al. Lora: Low-rank adaptation of large language models. In *International Conference on Learning Representations*, 2021.

[16] S. Huang, T. Wang, H. Xiong, J. Huan, and D. Dou. Semi-supervised active learning with temporal output discrepancy. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3447–3456, 2021.

[17] S. Huang, Y. Xie, S.-C. Zhu, and Y. 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.

 https://doi.org/10.52202/079017-1133

[18] A. J. Joshi, F. Porikli, and N. Papanikolopoulos. Multi-class active learning for image classification. In *2009 ieee conference on computer vision and pattern recognition*, pages 2372–2379. IEEE, 2009.

[19] S. Kato and K. Hotta. Enlarged large margin loss for imbalanced classification. *arXiv preprint arXiv:2306.09132*, 2023.

[20] K. Kim, D. Park, K. I. Kim, and S. Y. Chun. Task-aware variational adversarial active learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8166–8175, 2021.

[21] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[22] A. Krizhevsky, G. Hinton, et al. Learning multiple layers of features from tiny images. 2009.

[23] S. Lei, F. He, Y. Yuan, and D. Tao. Understanding deep learning via decision boundary. *IEEE Transactions on Neural Networks and Learning Systems*, 2023.

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

[25] H. Li. Support vector machine. In *Machine Learning Methods*, pages 127–177. Springer, 2023.

[26] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg. Ssd: Single shot multibox detector. In *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part I 14*, pages 21–37. Springer, 2016.

[27] X. Liu, K. Ji, Y. Fu, W. L. Tam, Z. Du, Z. Yang, and J. Tang. P-tuning v2: Prompt tuning can be comparable to fine-tuning universally across scales and tasks. *arXiv preprint arXiv:2110.07602*, 2021.

[28] Z. Liu, H. Ding, H. Zhong, W. Li, J. Dai, and C. He. Influence selection for active learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9274–9283, 2021.

[29] W. Luo, A. Schwing, and R. Urtasun. Latent structured active learning. *Advances in Neural Information Processing Systems*, 26, 2013.

[30] G. Ortiz-Jimenez, A. Modas, S.-M. Moosavi, and P. Frossard. Hold me tight! influence of discriminative features on deep network boundaries. *Advances in Neural Information Processing Systems*, 33:2935–2946, 2020.

[31] A. Parvaneh, E. Abbasnejad, D. Teney, G. R. Haffari, A. van den Hengel, and J. Q. Shi. Active learning by feature mixing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12237–12246, 2022.

[32] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, et al. Imagenet large scale visual recognition challenge. *International journal of computer vision*, 115(3):211–252, 2015.

[33] O. Sener and S. Savarese. Active learning for convolutional neural networks: A core-set approach. *arXiv preprint arXiv:1708.00489*, 2017.

[34] H. Shah, K. Tamuly, A. Raghunathan, P. Jain, and P. Netrapalli. The pitfalls of simplicity bias in neural networks. *Advances in Neural Information Processing Systems*, 33:9573–9585, 2020.

[35] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[36] S. Sinha, S. Ebrahimi, and T. Darrell. Variational adversarial active learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5972–5981, 2019.

[37] R. Strudel, R. Garcia, I. Laptev, and C. Schmid. Segmenter: Transformer for semantic segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 7262–7272, 2021.

[38] H. Touvron, M. Cord, M. Douze, F. Massa, A. Sablayrolles, and H. Jégou. Training data-efficient image transformers & distillation through attention. In *International Conference on Machine Learning*, pages 10347–10357. PMLR, 2021.

[39] K. Wang, D. Zhang, Y. Li, R. Zhang, and L. Lin. Cost-effective active learning for deep image classification. *IEEE Transactions on Circuits and Systems for Video Technology*, 27(12):2591–2600, 2016.

[40] Y. Xie, M. Ding, M. Tomizuka, and W. Zhan. Towards free data selection with general-purpose models. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023.

[41] Y. Xie, H. Lu, J. Yan, X. Yang, M. Tomizuka, and W. Zhan. Active finetuning: Exploiting annotation budget in the pretraining-finetuning paradigm. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 23715–23724, 2023.

[42] B. Xu, S. Shen, F. Shen, and J. Zhao. Locally linear svms based on boundary anchor points encoding. *Neural Networks*, 117:274–284, 2019.

[43] W. Xu, Z. Hu, Y. Lu, J. Meng, Q. Liu, and Y. Wang. Activedc: Distribution calibration for active finetuning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16996–17005, 2024.

[44] O. Yehuda, A. Dekel, G. Hacohen, and D. Weinshall. Active learning through a covering lens. *Advances in Neural Information Processing Systems*, 35:22354–22367, 2022.

[45] D. Yoo and I. S. Kweon. Learning loss for active learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 93–102, 2019.

[46] B. Zhou, H. Zhao, X. Puig, S. Fidler, A. Barriuso, and A. Torralba. Scene parsing through ade20k dataset. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 633–641, 2017.

[47] J. Zhou, C. Wei, H. Wang, W. Shen, C. Xie, A. Yuille, and T. Kong. ibot: Image bert pre-training with online tokenizer. *arXiv preprint arXiv:2111.07832*, 2021.

 https://doi.org/10.52202/079017-1133

A Related Work

Due to space limitations in the main part, a more detailed account of related work is provided here.

A.1 Active Learning / Finetuning

Active learning aims to maximize the efficacy of a limited annotation budget by strategically selecting the most informative data samples. In pool-based scenarios, existing algorithms generally adopt criteria such as uncertainty or diversity for sample selection. Uncertainty-based methods make selections based on the difficulty of each sample, assessed through various heuristics including posterior probability [24, 39], entropy [18, 29], loss function [16, 45], or their impact on model performance [10, 28]. Conversely, diversity-based algorithms aim to select representative samples that closely approximate the distribution of the entire data pool. This concept of diversity can be measured by the distance between global [33] and local [1, 40] representations, or through other metrics such as gradient directions [2] or adversarial loss [20, 36].

Nevertheless, the algorithms previously discussed are primarily tailored for training from scratch and encounter significant obstacles within the pretraining-finetuning paradigm [12, 41]. In response to this, ActiveFT [41] has been developed specifically for this framework. It operates by aligning the distribution of selected samples with that of the original unlabeled pool within the feature space. However, ActiveFT tends to prioritize high-density areas, frequently neglecting boundary samples. In contrast, our proposed algorithm integrates both diversity and uncertainty into the selection process, which is important for determining the decision boundary in the supervised finetuning.

A recent work ActiveDC [43] employs pseudo-labeling techniques from semi-supervised learning. However, this approach focuses on fine-tuning after sample selection rather than optimizing the sample selection process itself. While it performs exceptionally well under extremely small budgets, our tests show that it falls short when compared to the K-Nearest Neighbors or Linear Probing methods we discuss in Appendix G.

A.2 Decision Boundaries in Neural Networks.

Decision boundaries play a pivotal role in neural network-based classification models, significantly affecting both performance and interpretability [23, 25]. Optimizing these boundaries can substantially improve model generalizability and accuracy, especially in complex, high-dimensional data environments [42, 5]. In the case of support vector machines (SVMs), decision boundaries are integral to defining separating hyperplanes, and maximizing the geometric margin around these boundaries is crucial for robust classification [4]. This principle is equally pertinent to neural networks, where enhancing the margin can similarly boost generalization capabilities. When dealing with imbalanced datasets, adjusting the decision boundary becomes essential for accurately classifying minority classes. To address this, techniques such as Label-Distribution-Aware Margin (LDAM) loss [5] and Enlarged Large Margin (ELM) loss [19] have been developed to refine the decision boundary, thereby improving the balance of generalization errors across different classes. Studies have shown that neural networks tend to utilize the most discriminative or simplest features in constructing decision boundaries [30, 34]. Additionally, a theoretical framework has been proposed for evaluating the complexity of decision boundaries using the novel metric of decision boundary variability, which is inversely related to generalizability [23].

However, in the active finetuning task, the selection of decision boundaries has not traditionally been emphasized. Although challenging in unsupervised scenarios, we effectively leverage features from pretrained models, thereby introducing innovative denoising and selection methods without dependency on labels. This approach addresses a significant gap in the active finetuning domain.

B Details of ActiveFT Method

ActiveFT [41] selects the most useful data samples in the feature space of the pretrained model under the guidance of two basic intuitions: *1) bringing close the distributions between the selected subset $\mathcal{P}^u_{\mathcal{S}}$ and the original pool $\mathcal{P}^u$. 2) maintaining the diversity of $\mathcal{P}^u_{\mathcal{S}}$.* Formally, the goal is to find the

optimal selection strategy $\mathcal{S}$ as follows.

$$\mathcal{S}_{opt} = \arg\min_{\mathcal{S}} D(p_{f_u}, p_{f_\mathcal{S}}) - \lambda R(\mathcal{F}^u_\mathcal{S}) \tag{8}$$

where $D(\cdot,\cdot)$ is some distance metric between distributions p_{f_u} of $\mathcal{P}^u$ and $p_{f_\mathcal{S}}$ of $\mathcal{P}^u_\mathcal{S}$, $R(\cdot)$ is to measure the diversity of a set, and λ is a scale to balance these two terms.

Due to the difficulty in directly optimizing the discrete selection strategy $\mathcal{S}$, $p_{f_\mathcal{S}}$ is alternatively modeled with $p_{\theta_\mathcal{S}}$, where $\theta_\mathcal{S} = \{\theta^j_\mathcal{S}\}_{j\in[K]}$ are the continuous parameters and K is the budget size of core samples. 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 $\mathcal{S}$. The goal of this continuous optimization is written in Eq. 9.

$$\theta_{\mathcal{S},opt} = \arg\min_{\theta_\mathcal{S}} D(p_{f_u}, p_{\theta_\mathcal{S}}) - \lambda R(\theta_\mathcal{S}) \quad s.t. ||\theta^j_\mathcal{S}||_2 = 1 \tag{9}$$

Following the mathematical derivation in [41], Eq. 9 can be solved by optimizing the following loss function through gradient descent.

$$\begin{aligned} L &= D(p_{f_u}, p_{\theta_\mathcal{S}}) - \lambda \cdot R(\theta_\mathcal{S}) \\ &= -\mathop{E}_{\mathbf{f}_i\in\mathcal{F}^u}[sim(\mathbf{f}_i, \theta^{c_i}_\mathcal{S})/\tau] + \mathop{E}_{j\in[B]}\left[\log \sum_{k\neq j, k\in[B]} \exp\Big(sim(\theta^j_\mathcal{S}, \theta^k_\mathcal{S})/\tau\Big)\right] \end{aligned} \tag{10}$$

where $sim(\cdot,\cdot)$ is a cosine similarity between normalized features, the temperature $\tau = 0.07$ and the balance weight λ is empirically set as 1.

After the optimization process, it finds features $\{\mathbf{f}_{s_j}\}_{j\in[K]}$ with the highest similarity to $\theta^j_\mathcal{S}$.

$$\mathbf{f}_{s_j} = \arg\max_{\mathbf{f}_k\in\mathcal{F}^u} sim(\mathbf{f}_k, \theta^j_\mathcal{S}) \tag{11}$$

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

C Denoising Algorithm

We propose three progressive methods to address the challenge of denoising: the distance-guide method, the density-based method, and iterative density-based clustering (IDC), with the latter elaborated in detail in the main body of the paper. In this appendix section, we introduce two additional methods.

Distance-guide method: This method removes the top $N_{i,rm}$ samples that are furthest from the current class center, based on their distance. Although straightforward and simple, it has limitations, particularly in scenarios where the class distribution varies significantly in different directions, resembling an elliptical shape. In such cases, this rudimentary method may inadvertently remove a distant cluster of samples instead of the actual noise.

Density-based method: This approach removes the top $N_{i,rm}$ samples with the highest density distance from their class. The underlying rationale is that noise samples are generally farther away from their neighboring samples. Thus, using the distance to nearby samples as an auxiliary measure to identify noise is reasonable. The definition of density distance is provided in Eq. 2.

D Time Complexity of BiLAF

Theoretical Time Complexity. Given that our approach is a bi-level method, the overall time is influenced by two components: Core Sample Selection and Boundary Sample Selection. Here, we analyse the theoretical complexity of the Boundary Sample Selection component in Algorithm 1.

Define N as the total number of samples, B as the annotation budget, $M = \alpha N$ as the number of core samples, T as the number of cluster iterations, where $\alpha = 0.5\%$ in CIAFR100 and $T =$

https://doi.org/10.52202/079017-1133

$1/P_{in} = 1/10\% = 10$ in all the datasets. Based on Algorithm 1, we divide the Stage 2 Boundary Samples Selection into three steps:

1. Assigning the pseudo-class samples is $O(NM) = O(\alpha N^2)$ because we need to assign each data sample to its corresponding pseudo-center.

2. Iteratively density-based clustering then denoising is $O(TN/\alpha)$. Considering that for each pseudo-center, the expected number of samples is N/M, we need to compute pairwise distances and find the k nearest samples, resulting in a complexity of $O(N^2/M^2)$. Subsequently, we must sort these distances based on density distance, which has a complexity of $O(N/M \log N/M)$. We need to repeat the process for T times in the iterative clustering. Therefore the time is $O(TN^2/M^2 + TN/M \log N/M)$ for each pseudo-center. The total complexity is then multiplied by M, yielding an overall complexity of $O(TN^2/M + TN \log N/M) = O(TN/\alpha + TN \log 1/\alpha) = O(TN/\alpha)$.

3. Selection process is $O(N/\alpha + BN)$. Considering each pseudo-center, the expected number of samples is N/M. We need to calculate the pairwise distances to compute the intra-class distance, d_{intra}, resulting in a complexity of $O(N^2/M^2)$. When considering the inter-class distances, due to the presence of the coefficient δ^{t_l}, adjustments are required each time, thus the complexity becomes $|B_i| \times (N/M \times M) = O(|B_i|N)$, where $|B_i|$ represents the number of selections within this pseudo-category, and $\sum_{i=1}^{M} |B_i| = B$. Therefore, the overall complexity in this process is $O(N^2/M + BN) = O(N/\alpha + BN)$.

Thus, the overall complexity is $O(\alpha N^2 + TN/\alpha + BN)$. In practice, matrix multiplication and constants can impact the speed across different steps.

Experimental Results. Tab. 3 provides the specific execution times of our method (including the time spent on selecting core samples using ActiveFT). Actually, we should not omit T, which is a relatively small constant (e.g., 10), because T/α can be comparable in scale to B when B is small. Therefore, the complexity remains almost consistent according to different B in Tab. 3.

E Experiment Setup

Due to space limitations in the main part, we provide a comprehensive overview of the experiment setup for classification tasks in this section.

Datasets and Evaluation Metrics. Our approach is evaluated using three widely recognized datasets: CIFAR10, CIFAR100 [22], and ImageNet-1k [32]. Both CIFAR10 and CIFAR100 contain 60,000 images with resolutions of 32x32, but they differ in classification complexity, offering 10 and 100 categories, respectively. Each comprises 50,000 images for training and 10,000 for testing. The large-scale dataset ImageNet-1k includes 1,000 categories and a total of 1,281,167 training images along with 50,000 validation images. The raw training data serve as the unlabeled pool $\mathcal{P}^u$ from which selections are made. We employ the *Top-1 Accuracy* metric for performance evaluation.

Baselines. We compare our approach with three heuristic baselines, five active learning methods and the well-designed active finetuning method ActiveFT [41].

Three heuristic baselines are listed as follows:

1. **Random**: The selection of samples for annotation is entirely stochastic.
2. **FDS**: The method is also known as the K-Center-Greedy algorithm, which selects the next sample feature that is farthest from the current selections. As proven in [33], it minimizes the expected loss over the entire pool and the selected subset.
3. **K-Means**: We implement the K-Means method on the feature pool $\mathcal{F}^u$ and select samples nearest to the centroids, where the number K equals to the budget size B.

Five active learning methods are listed as follows:

1. **CoreSet** [33]: CoreSet selects samples that are central to the data distribution, effectively minimizing the maximum distance between any data point and the nearest selected sample, thus representing the core characteristics of the dataset.

2. **VAAL** [36]: The Variational Adversarial Active Learning (VAAL) framework combines variational autoencoders with adversarial learning to facilitate active learning in an unsupervised setting. It utilizes a discriminator to distinguish between labeled and unlabeled data, guiding the selection of samples likely to enhance model performance upon labeling.
3. **LearnLoss** [45]: LearnLoss employs a deep learning strategy to estimate the value of labeling each data sample. It involves training a secondary network alongside the primary model to predict the potential reduction in loss from labeling specific unlabeled samples.
4. **TA-VAAL** [20]: Task-Aware Variational Adversarial Active Learning (TA-VAAL) enhances the VAAL framework by integrating task-specific objectives into the adversarial model, making the active learning process more attuned to the requirements of the specific task, thereby optimizing data selection.
5. **ALFA-Mix** [31]: ALFA-Mix merges active learning with data augmentation techniques to boost model performance. It selects informative samples and employs a mixup strategy to create synthetic samples by blending selected data points, thereby enriching the training dataset and promoting robust feature learning.

These methods has been modified and applied specifically for active finetuning tasks following the instructions in [41]. These approaches utilize a batch-selection strategy for data sampling. Initially, the model is trained on a randomly chosen initial dataset. Subsequently, it employs this trained model to pick a batch of images from the training set. The model is then retrained using the cumulatively selected samples. The chosen active learning techniques encompass strategies based on both diversity and uncertainty in the active learning domain, serving as the baselines in the active finetuning field.

In this scenario, **ActiveFT** [41] emerges as the strongest baseline due to its tailored design for the task. This approach generates a representative subset from the unlabeled pool by aligning its distribution with the entire dataset and enhances diversity through the optimization of parametric models within a continuous space.

Implementation Details. In the unsupervised pretraining phase, we use DeiT-Small [38] model, pretrained using the DINO [6] framework on ImageNet-1k, due to its recognized efficiency and popularity. For all three datasets, we resize images to 224×224 consistent with the pretraining for both data selection and supervised finetuning. In core samples selection stage, we utilize ActiveFT and optimize the parameters $\theta_{\mathcal{S}}$ using the Adam [21] optimizer (learning rate $1e$-3) until convergence. We set the core number K as $50(0.1\%), 250(0.5\%), 6405(0.5\%)$ for CIFAR10, CIFAR100 and ImageNet separately. In the boundary samples selection stage, we set nearest neighbors number k as 10, both removal ratio P_{rm} and clustering fraction P_{in} as 10%, opponent penalty coefficient δ as 1.1. In the supervised finetuning phase, we finetune DeiT-Small model [1] following the setting in [41]. We finetune the models using the SGD optimizer with learning rate as $3e$-3, weight decay as $1e$-4 and momentum as 0.9. We employ cosine learning rate decay with a batch size of 256 distributed across two GPUs. The models are finetuned for 1000 epochs on all datasets with different sampling ratios, except for ImageNet with sampling ratio 5%, where we finetune for 300 epochs. Our experiments are implemented using the mmclassification framework [2].

F More Extensive Experiments

To further investigate the generalizability of our method, we conducted extensive experiments. Firstly, We applied our method to diverse downstream tasks, including Object Detection (F.1) and Semantic Segmentation datasets (F.2). Then, we explored the performance of our method in different scenarios, such as the long-tail distribution scenarios (F.3). Finally, we conduct experiments on fine-grained classification scenarios (F.4).

F.1 Object Detection

Datasets. We conduct our experiments on the PASCAL VOC dataset [9]. Following established protocols from prior studies [45], we merge the training and validation sets of PASCAL VOC 2007 and 2012, forming a training data pool of 16,551 images. We assess the performance of the task

[1] https://github.com/facebookresearch/deit

[2] https://github.com/open-mmlab/mmclassification

https://doi.org/10.52202/079017-1133

model on the PASCAL VOC 2007 test set, utilizing the mAP (mean Average Precision) metric for evaluation.

Implementation Details. To facilitate comparison and model alignment in testing, we only use the DeiT-Small for feature extraction and data selection in this dataset. In the data selection stage, We set the core number K as 500. The other settings are the same as described in Sec. 4.1 in main part. Then in the finetuning process, for alignment with methods previously documented in [45], we train an SSD-300 model [26] with a VGG-16 backbone [35] on the selected samples. Following protocols from prior studies [45], we train the model for 300 epochs with a batch size of 32. We utilize an SGD optimizer with a momentum of 0.9. The initial learning rate is set at $1e$-3, decaying to $1e$-4 after 240 epochs. Our experiments are implemented using the mmdetection framework [3].

F.2 Semantic Segmentation

Datasets. For the segmentation task, we utilize the ADE20K dataset [46], which comprises 20,210 images for training, 2,000 images for validation, and 3,352 images for testing. Each image is annotated with fine-grained labels across 150 semantic classes. We designate the training set as the unlabeled pool $\mathcal{P}^u$ for selection purposes. The performance is assessed using the *mIoU* (mean Intersection over Union) metric.

Implementation Details. We utilize DeiT-Small [38] model pretrained with DINO framework [6] the same as in image classification tasks. In the data selection stage, We set the core number K as 2.5% of data samples. The other settings are the same as described in Sec. 4.1 in main part. We resize the images to 224×224 as well and adopt Segmenter [37] for finetuning in the segmentation task, which is a pure transformer model, using the same DeiT-Small model as its backbone for finetuning. The model is trained for 127 epochs following [37], corresponding to $16\text{k}/32\text{k}/48\text{k}$ iterations on $5\%/10\%/15\%$ of the training data, respectively. Training employs the SGD optimizer (learning rate = $1e$-3, momentum = 0.9) with a batch size of 8 and polynomial decay of the learning rate. Our experiments are implemented using the mmsegmentation framework [4].

F.3 Long-Tailed Classification

Datasets. In the long-tailed datasets, we define the imbalance ratio (IR) as IR = n_{max}/n_{min}, where n_{max} and n_{min} represent the number of training samples in the largest and smallest class, respectively. CIFAR100LT is derived from the original balanced CIFAR100 dataset [22], which includes 50,000 training images and 10,000 test images, each sized 32×32 across 100 classes. Following the methodology of [5], we constructed the long-tailed version, CIFAR100LT, by applying exponential decay to the sampling of training images per class, while keeping the balanced test set unchanged. We utilize CIFAR100LT with imbalance factors (IR) as 100 in our experiments. CIFAR100LT (IR=100) has 10847 training samples.

Implementation Details. We set the core number K as $217(2\%)$ for CIFAR100LT. Other settings are consistent with the main experiment. See Appendix E for reference.

Table 9: **CUB-200-2011 Dataset.** Comparison of Random, ActiveFT, and BiLAF on Different Selection Budget.

Budget	Select Number	Random	ActiveFT	BiLAF (Ours)
20%	1198	46.32	47.83	**48.53**
30%	1798	58.85	59.67	**60.52**
40%	2397	66.75	67.36	**68.31**
50%	2997	72.98	73.25	**74.06**

[3] https://github.com/open-mmlab/mmdetection

[4] https://github.com/open-mmlab/mmsegmentation

F.4 Fine-Grained Classification

We utilized the CUB-200-2011 dataset, which includes 200 bird species with a total of $11,788$ images. According to the default configuration of the dataset, $5,994$ samples are used for training, with the remainder used for testing. Given the large number of classes, we used 10% of the data as Core Samples to select Boundary Samples, while all other parameters were set according to the default values reported in the paper. Tab. 9 clearly demonstrates that our approach continues to hold a leading position in fine-grained classification datasets.

G Extra Finetuning Methods: Linear Probing and K-Nearest Neighbors

Aside from the Full Fine-Tuning, Linear Probing or even K-Nearest Neighbors (KNN) classifiers are also effective approaches. In this section, we explore whether BiLAF remains selecting high-quality data points with these fine-tuning methods. We conducted experiments using Linear Probing and KNN on the CIFAR10 dataset, comparing the results across Random, ActiveFT, and **BiLAF (ours)**.

Table 10: Results on CIFAR10 using KNN, Linear Probing, and Full Fine-Tuning across different budgets.

Finetuning Methods	**Selection Method**	$B = 0.5\%$	$B = 1\%$	$B = 2\%$	$B = 5\%$
K-Nearest Neighbors	Random	82.1	86.7	88.2	90.8
K-Nearest Neighbors	ActiveFT	86.8	87.2	88.5	91.7
K-Nearest Neighbors	BiLAF (ours)	85.7	87.4	88.7	91.8
Linear Probing	Random	85.1	87.6	90.1	92.5
Linear Probing	ActiveFT	**87.8**	88.7	90.6	92.8
Linear Probing	BiLAF (ours)	86.5	89.1	91.0	93.0
Full Fine-Tuning	Random	77.3	82.2	88.9	94.8
Full Fine-Tuning	ActiveFT	85.0	88.2	90.1	95.2
Full Fine-Tuning	BiLAF (ours)	81.0	**89.2**	**92.5**	**95.7**

From the table, the following conclusions can be drawn:

1. At extremely low data volumes, Linear Probing and K-Nearest Neighbors outperform Full Fine-Tuning.
2. As the data volume increases, the performance improvements of Linear Probing and K-Nearest Neighbors start to slow down, which gradually **necessitates the use of Full Fine-Tuning**.
3. Interestingly, the quality of data selected by different methods shows a consistent trend across K-Nearest Neighbors classifiers, Linear Probing, and Full Fine-Tuning. **Our method, compared to competitors, is able to select more suitable data, which is effective across different fine-tuning paradigms.**

H Analysis of Selection Samples to the Decision Boundary.

We conduct linear probing on all the samples with true labels using features from both the pre-trained model and the oracle model which is fully finetuned on all samples. We analyze whether samples selected using different methods—Random, ActiveFT, BiLAF (ours)—tend to be near the decision boundaries. We use two metrics for this analysis: 1) **Entropy**, where a higher value indicates greater uncertainty and a propensity towards boundary samples. 2) **ProbDiff**, calculated as the difference between the highest and second-highest probabilities. A smaller value indicates that the sample is closer to the boundary between these two classes.

From the Tab 11, 12, 13 and 14, the results demonstrate that our method can effectively select boundary samples, maintaining consistency across models with various capabilities.

https://doi.org/10.52202/079017-1133

Table 11: Results of Entropy and ProbDiff on CIFAR10 using Pretrained Model.

CIFAR10 (Pretrained)	Selected Nums	Entropy ↑	ProbDiff ↓	Entropy (Top50) ↑	ProbDiff (Top50) ↓
Random	250	0.0935	0.9486	0.4260	0.7536
ActiveFT	250	0.0424	0.9747	0.2073	0.8743
BiLAF (ours)	250	**0.1023**	**0.9366**	**0.4769**	**0.6924**
Random	500	0.0960	0.9416	0.7022	0.5056
ActiveFT	500	0.0433	0.9763	0.3690	0.7799
BiLAF (ours)	500	**0.1149**	**0.9273**	**0.7089**	**0.4500**
Random	1000	0.0955	0.9430	0.9181	0.3081
ActiveFT	1000	0.0849	0.9495	0.8810	0.3560
BiLAF (ours)	1000	**0.1461**	**0.9064**	**1.0262**	**0.2005**

Table 12: Results of Entropy and ProbDiff on CIFAR100 using Pretrained Model.

CIFAR100 (Pretrained)	Selected Nums	Entropy ↑	ProbDiff ↓	Entropy (Top50) ↑	ProbDiff (Top50) ↓
Random	500	**0.6240**	**0.7295**	**2.2516**	**0.0876**
ActiveFT	500	0.2962	0.8664	1.5663	0.2369
BiLAF (ours)	500	0.3933	0.8167	1.7832	0.1423
Random	1000	**0.5317**	**0.7766**	2.2375	0.0594
ActiveFT	1000	0.3650	0.8430	2.0812	0.0833
BiLAF (ours)	1000	0.4751	0.7815	**2.2606**	**0.0542**
Random	2500	0.5253	0.7749	2.6790	0.0232
ActiveFT	2500	0.4851	0.7936	2.6653	0.0206
BiLAF (ours)	2500	**0.5795**	**0.7476**	**2.7196**	**0.0192**
Random	5000	0.5442	0.7652	2.8791	0.0151
ActiveFT	5000	0.5197	0.7768	2.8487	0.0118
BiLAF (ours)	5000	**0.6219**	**0.7336**	**2.9194**	**0.0090**

Table 13: Results of Entropy and ProbDiff on CIFAR10 using Oracle Model.

CIFAR10 (Oracle)	Selected Nums	Entropy ↑	ProbDiff ↓	Entropy (Top50) ↑	ProbDiff (Top50) ↓
Random	250	0.001512	0.999831	0.002437	0.999729
ActiveFT	250	0.001521	0.999830	0.002403	0.999717
BiLAF (ours)	250	**0.001541**	**0.999827**	**0.002473**	**0.999714**
Random	500	0.001483	0.999835	0.002782	0.999676
ActiveFT	500	0.001542	0.999828	0.002958	0.999644
BiLAF (ours)	500	**0.001605**	**0.999820**	**0.003020**	**0.999641**
Random	1000	0.001551	0.999823	0.003543	0.999575
ActiveFT	1000	0.001527	0.999829	0.003472	0.999586
BiLAF (ours)	1000	**0.001598**	**0.999820**	**0.003588**	**0.999567**

Table 14: Results of Entropy and ProbDiff on CIFAR100 using Oracle Model.

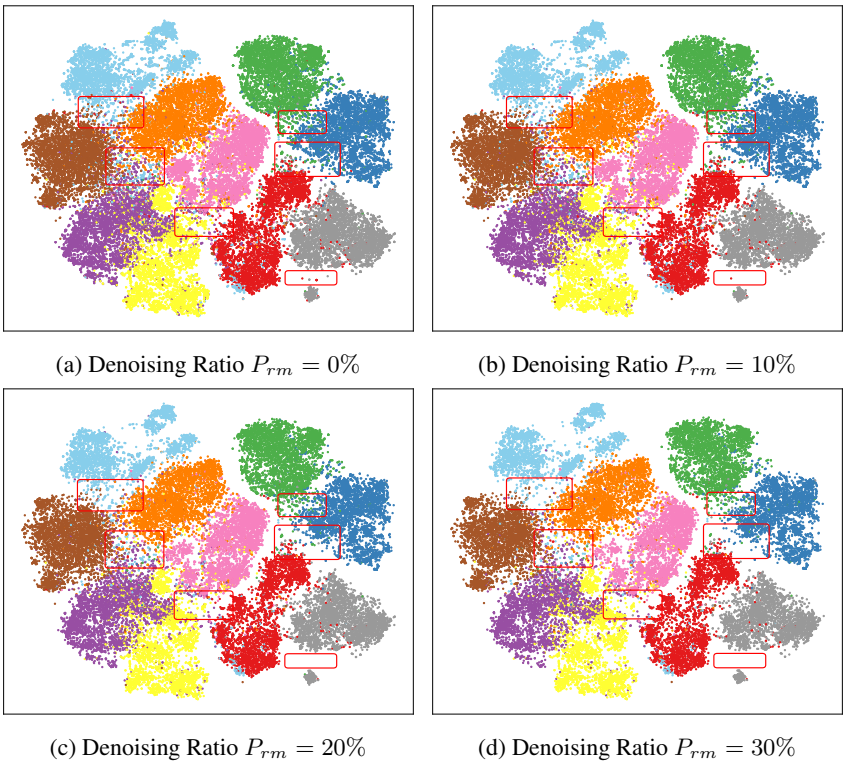
CIFAR100 (Oracle)	Selected Nums	Entropy ↑	ProbDiff ↓	Entropy (Top50) ↑	ProbDiff (Top50) ↓
Random	500	**0.049238**	**0.992216**	**0.176811**	**0.956287**
ActiveFT	500	0.040140	0.993335	0.124594	0.962662
BiLAF (ours)	500	0.042792	0.994406	0.137114	0.965457
Random	1000	**0.047730**	0.993905	0.202417	0.963037
ActiveFT	1000	0.042763	0.993913	0.198896	0.961337
BiLAF (ours)	1000	0.045447	**0.993854**	**0.203469**	**0.960462**
Random	2500	0.046654	0.993407	0.297133	0.919803
ActiveFT	2500	0.047036	0.993245	0.303767	0.919122
BiLAF (ours)	2500	**0.047357**	**0.993103**	**0.309907**	**0.917868**
Random	5000	0.047028	0.993741	0.407092	0.897825
ActiveFT	5000	0.045173	0.994264	0.340326	0.925870
BiLAF (ours)	5000	**0.049476**	**0.992885**	**0.466601**	**0.842029**

I Limitation and Social Impact.

Limitation. In this paper, our approach is based on general features and purposes. For specific tasks, such as object detection task with multi-instances, our method does not show a significant advantage. What's more, in dealing with sample features in long-tail problems, denoising methods might remove outliers that are key samples. However, as a flexible framework, we can make corresponding improvements for different downstream tasks, which can be considered as future work.

Social Impact. We firmly believe that this work has a positive impact on society. By replacing iterative sample selection with one-shot sample selection, our approach not only saves substantial resources that would otherwise be wasted in training, but also enhances labeling efficiency without the need for multiple arrangements. Additionally, our method achieves state-of-the-art performance among similar approaches based on a small number of samples, effectively reducing resource waste caused by ineffective labeling.

J Qualitative Visualization



(a) Denoising Ratio $P_{rm} = 0\%$ (b) Denoising Ratio $P_{rm} = 10\%$

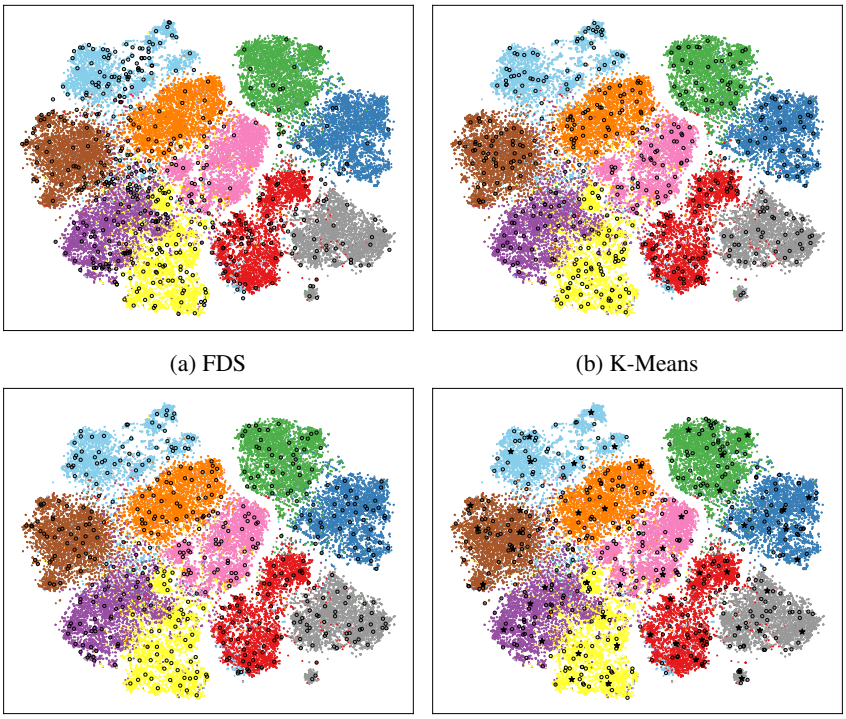
(c) Denoising Ratio $P_{rm} = 20\%$ (d) Denoising Ratio $P_{rm} = 30\%$

Figure 5: **Denoising Strength Visualization using tSNE Embeddings of CIFAR10**.

Denoising Visualization. Fig. 5 illustrates the impact of variations in the removal rate P_{rm} during the denoising process on the retained samples. The red bounding boxes highlight regions with significant changes. These areas include obvious outliers and zones of confusion where multiple class intermingle. As the removal rate increases, the number of samples in these areas tends to decrease gradually, thereby reducing their influence on subsequent boundary point selection, while the relatively dense boundaries are often preserved. However, this represents a trade-off as some important samples might also be removed. Therefore, we have conducted a quantitative analysis in the main text to address this concern.

Selected Samples Visualization. Fig. 6 displays the sample selection by different methods when annotating 1% samples from the CIFAR100 dataset. It is evident that the FDS method has significant

https://doi.org/10.52202/079017-1133



(a) FDS (b) K-Means

(c) ActiveFT (d) BiLAF (ours)

Figure 6: **tSNE Embeddings on CIFAR10 with 1% annotation budget of our BiLAF method and other methods.**

drawbacks, leading to an insufficient selection of sample samples from central classes in feature space, which greatly impedes the model's learning capability. Both the K-Means and ActiveFT methods focus on selecting central samples, differing in their optimization goals and processes. As seen in Fig. 6, both methods achieve their intended purpose, resulting in a relatively uniform selection of samples. Our method further refines this approach. Firstly, we identify the central samples, represented by pentagrams, and then expand to boundary samples, denoted by circles, from each center. Our strategy emphasizes the boundaries between categories, rather than conforming to the entire distribution. For example, considering the 'brown' sample class, our method selects fewer samples in its internal area, far from other categories, and focuses more on locating boundary samples. Extensive experimental results quantitatively demonstrate the substantial improvement our method brings to model performance, validating its effectiveness.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The main claims are clearly presented in the abstract and introduction, and are further elaborated and substantiated throughout the subsequent sections of the paper.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: Refer to Appendix I.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory Assumptions and Proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes]

https://doi.org/10.52202/079017-1133

Justification: Refer to Appendix D.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental Result Reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: In our paper, we provide detailed descriptions of each step of the algorithm, the pseudocode, and all settings required to replicate the experiments. Additionally, we commit to making the code publicly available by the time of the camera-ready submission.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: We provide detailed descriptions of each step of the algorithm, the pseudocode, and all settings required to replicate the experiments. Meanwhile, we make the code publicly available with the GitHub link in the title.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental Setting/Details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: Refer to the main part and Appendix E and F.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment Statistical Significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: We report the mean and variance of the experiment results.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).

https://doi.org/10.52202/079017-1133

- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments Compute Resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: Refer to Sec. 4.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code Of Ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: Our work adheres to the NeurIPS Code of Ethics.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader Impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: Refer to Appendix I.

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.
    - If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.

- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [NA]

    Justification: The paper poses no such risks.

    Guidelines:

    - The answer NA means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: The creators and original owners of all assets used in the paper are properly credited, and the licenses and terms of use are explicitly mentioned and fully respected.

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.

https://doi.org/10.52202/079017-1133

- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New Assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [NA]

    Justification: NA.

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and Research with Human Subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

    Justification: NA.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional Review Board (IRB) Approvals or Equivalent for Research with Human Subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

    Justification: NA.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.