Supplementary Material for the Paper "Active Finetuning: Exploiting Annotation Budget in the Pretraining-Finetuning Paradigm"

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

In this supplementary material, we first analyze the effect of iteration number for the optimization of ActiveFT in Sec. A. Then, we provide more implementation details in Sec. B, including some explanation of N/A results in Tab. 1 of our main paper. Finally, we give a formal proof of the optimal joint distribution in Eq. 15 of our main paper in Sec. C.

A. Ablation Study on Iteration Number

We conduct an additional ablation study of the maximal iteration number T (in Alg. 1 of the main paper) of the parametric model optimization process in ActiveFT. The experiments are conducted on ImageNet [7] with sampling ratio 1%. Results are demonstrated in Tab. 1. The quality of samples selected by ActiveFT continuously improves in the early stage as the optimization of our parametric model p_{θ_S} goes, and then converges in the late stage. This result verifies that our model optimization gradually brings close the distributions of our selected samples to the entire unlabeled pool as well as ensures the diversity of the selected subset in the whole optimization process.

Table 1. **Ablation Study of Iteration Numbers:** Experiments are conducted on ImageNet [7] dataset (1% sampling ratio) with DeiT-Small [11] model pretrained with DINO [1] framework. When iteration number is 0, it is same as random selection.

Sel. Ratio	**Iteration Number**					
	0	**50**	**75**	**100**	**200**	**300**
1%	45.1	46.7	48.4	**50.2**	50.1	50.1

B. Additional Implementation Details

B.1. Unsupervised Pretraining Details

In our main paper, the DeiT-Small model (path size 16x16) [11] is pretrained on ImageNet [7] with DINO framework [1] [1] for 300 epochs using AdamW optimizer [5] and batch size 1024. The learning rate is linearly ramped up to 5e-4×batch_size/256 in the first 10 epochs and decays with a cosine scheduler later.

In Tab. 4 of our main paper, the DeiT-Small model [11] is pretrained with iBOT framework[2] [13] on ImageNet [7] for 800 epochs. The ResNet-50 model [2] is pretrained with DINO framework [1] on ImageNet for 300 epochs. The optimizer is AdamW [5] and the batch size is 1024 in both cases. The learning rate is linearly ramped up to 5e-4×batch_size/256 in the first 10 epochs too.

B.2. Supervised Finetuning Details

We typically follow the protocols in [11] to finetune the DeiT-Small model. For CIFAR10 and CIFAR100 [4] datasets, the pretrained models are supervisedly finetuned for 1000 epochs using SGD optimizer (lr=1e-3, weight-decay=1e-4, momentum=0.9) with batch size 512 and cosine learning rate decay on selected subsets of training data. For ImageNet [7] dataset, to ensure convergence, the models are finetuned for 1000 epochs when the sampling ratio is 1% and for 300 epochs when the sampling ratio is 5%, using the same SGD optimizer as CIFAR. The images are resized to 224x224 in line with the pretraining. The supervised finetuning is implemented based on the official code of DeiT [3]. For ResNet-50 model in Tab. 4 of our main paper, we use the code base of mmclassification [4]. We follow their settings to finetune the model with SGD optimizer (lr=1e-2, weight-decay=1e-4, momentum=0.9) with batch size 512 and cosine learning rate decay on selected subsets of training data for 100 epochs.

On the semantic segmentation task, we follow [10] to train the model for 127 epochs (*i.e.* 16k and 32k iterations on 5% and 10% of training data). The model is trained using SGD optimizer (lr=1e-3, momentum=0.9) with batch size 8 and polynomial learning rate decay. The code base is mmsegmentation [5].

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

[2] https://github.com/bytedance/ibot

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

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

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

B.3. Active Learning Transplantation Details

We transplant three classical active learning methods and two newer algorithms to the pretraining-finetuning paradigm, including CoreSet [8], VAAL [9], LearnLoss [12], TA-VAAL [3], and ALFA-Mix [6].

For all five methods, we apply them to image classification task on CIFAR10, CIFAR100 and ImageNet. These methods select data samples with batch-selection strategy. Firstly, we train the model on a randomly sampled initial set. Then, the model is used to select a batch of images from the training set, and the model is re-trained on all the selected samples. This process repeats until the annotation budget is filled. In the pretraining-finetuning paradigm, for CoreSet, LearnLoss and ALFA-Mix, we use DeiT-Small [11] pretrained with DINO [1] as the backbone of their models for data selection. For VAAL and TA-VAAL, we directly use their original light-weighted VAE to select data. When the data samples have been selected with different sampling ratios, we finetune the DeiT-Small model in the same manner as Sec. B.2 on the selected data samples. The sizes of the initial set and each selection batch are set as 0.5% on CIFAR10, 1% on CIFAR100, and 2.5% on ImageNet separately for all the five algorithms.

B.4. Explanation of N/A Results

There are some N/A results (denoted by "-") in Tab. 1 of our main paper. We explain them from the following three angles.

- **Initial Set of Active Learning:** As described in Sec. B.3, all five active learning methods require to randomly sample a small initial set in the beginning. On this initial set, the performance of these active learning algorithms is same as random sampling. Therefore, we pass the duplicate results on these random initial sets *i.e.* 0.5% of CIFAR10 and 1% of CIFAR100. Since 1% is smaller than the initial set size (2.5%) on ImageNet, we pass this sampling ratio as well.

- **K-Means on ImageNet:** Given the large number of images in training set, it is hard to implement K-Means to ImageNet dataset, which exceeds the capability of our hardware. Since K-Means does not perform well on CIFAR10 and CIFAR100, the N/A results on ImageNet would not affect our conclusions.

C. Proof of the Optimal Distributions for Earth Mover's Distance

In Sec. 3.4 of our main paper, we give an optimal distribution to calculate the earth mover's distance (EMD), *i.e.* each $\mathbf{f}_i \sim p_{f_u}$ transports to their closest $\mathbf{f}_{s_j} \sim p_{f_S}$. The Eq. 15 in the main paper is copied as follows:

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

We will prove it is the optimal joint distribution γ to reach the infimum in Eq. 14 of our main paper, copied as follows:

$$EMD(p_{f_u},p_{f_S}) = \underset{\gamma\in\Pi(p_{f_u},p_{f_S})}{inf} \quad \underset{(\mathbf{f}_i,\mathbf{f}_{s_j})\sim\gamma}{E} \left[||\mathbf{f}_i - \mathbf{f}_{s_j}||_2\right] \tag{2}$$

Suppose there is a general format:

$$\gamma_{f_u,f_S}(\mathbf{f}_i,\mathbf{f}_{s_j}) = p(\mathbf{f}_i,\mathbf{f}_{s_j}) \qquad \mathbf{f}_i \in \mathcal{F}^u, \mathbf{f}_{s_j} \in \mathcal{F}^u_{\mathcal{S}} \tag{3}$$

Because of the uniform distribution of p_{f_u}, $p(\mathbf{f}_i,\mathbf{f}_{s_j})$ satisfies the following constraints.

$$p(\mathbf{f}_i,\mathbf{f}_{s_j}) \geq 0, \qquad \sum_{\mathbf{f}_{s_j}\in\mathcal{F}^u_{\mathcal{S}}} p(\mathbf{f}_i,\mathbf{f}_{s_j}) = p_{f_u}(\mathbf{f}_i) = \frac{1}{N}. \tag{4}$$

The distance expectation for each feature $\mathbf{f}_i$ with the distribution $\mathcal{F}^u$ is

$$\begin{aligned} \underset{\mathbf{f}_{s_j}\in\mathcal{F}^u_{\mathcal{S}}}{E}\left[||\mathbf{f}_i - \mathbf{f}_{s_j}||_2\right] &= \sum_{\mathbf{f}_{s_j}\in\mathcal{F}^u_{\mathcal{S}}} \left[p(\mathbf{f}_{s_j}|\mathbf{f}_i)\cdot||\mathbf{f}_i - \mathbf{f}_{s_j}||_2\right] \\ &= \sum_{\mathbf{f}_{s_j}\in\mathcal{F}^u_{\mathcal{S}}} \left[p(\mathbf{f}_i,\mathbf{f}_{s_j})/p_{f_u}(\mathbf{f}_i)\cdot||\mathbf{f}_i - \mathbf{f}_{s_j}||_2\right] \\ &= N \sum_{\mathbf{f}_{s_j}\in\mathcal{F}^u_{\mathcal{S}}} \left[p(\mathbf{f}_i,\mathbf{f}_{s_j})\cdot||\mathbf{f}_i - \mathbf{f}_{s_j}||_2\right] \end{aligned} \tag{5}$$

Since the $\mathbf{f}_{s_{c_i}}$ is the nearest feature of $\mathbf{f}_i$ in $\mathbf{f}_{s_j} \sim \mathcal{F}^u_{\mathcal{S}}$, we can have the inequality

$$||\mathbf{f}_i - \mathbf{f}_{s_j}||_2 \geq ||\mathbf{f}_i - \mathbf{f}_{s_{c_i}}||_2 \tag{6}$$

so

$$\begin{aligned} \underset{\mathbf{f}_{s_j}\in\mathcal{F}^u_{\mathcal{S}}}{E}\left[||\mathbf{f}_i - \mathbf{f}_{s_j}||_2\right] &\geq N \cdot \left[\sum_{\mathbf{f}_{s_j}\in\mathcal{F}^u_{\mathcal{S}}} p(\mathbf{f}_i,\mathbf{f}_{s_j})\right] \cdot ||\mathbf{f}_i - \mathbf{f}_{s_{c_i}}||_2 \\ &= N \cdot \frac{1}{N} ||\mathbf{f}_i - \mathbf{f}_{s_{c_i}}||_2 \\ &= ||\mathbf{f}_i - \mathbf{f}_{s_{c_i}}||_2 \end{aligned} \tag{7}$$

The above minimum is reached when Eq. 1 (Eq. 15 in our main paper) is satisfied. Therefore, for each $\mathbf{f}_i$, we only need to find the nearest feature $\mathbf{f}_{s_{c_i}}$ among $\mathbf{f}_{s_j} \sim \mathcal{F}^u_{\mathcal{S}}$ and assign the joint probability as $\frac{1}{N}$.

References

[1] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9650–9660, 2021. 1, 2

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

[3] Kwanyoung Kim, Dongwon Park, Kwang In Kim, and Se Young Chun. Task-aware variational adversarial active learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8166–8175, 2021. 2

[4] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009. 1

[5] Ilya Loshchilov and Frank Hutter. Fixing weight decay regularization in adam. 2018. 1

[6] Amin Parvaneh, Ehsan Abbasnejad, Damien Teney, Gholamreza Reza Haffari, Anton van den Hengel, and Javen Qinfeng Shi. Active learning by feature mixing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12237–12246, 2022. 2

[7] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *International journal of computer vision*, 115(3):211–252, 2015. 1

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

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

[10] Robin Strudel, Ricardo Garcia, Ivan Laptev, and Cordelia Schmid. Segmenter: Transformer for semantic segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 7262–7272, 2021. 1

[11] Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Hervé Jégou. Training data-efficient image transformers & distillation through attention. In *International Conference on Machine Learning*, pages 10347–10357. PMLR, 2021. 1, 2

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

[13] Jinghao Zhou, Chen Wei, Huiyu Wang, Wei Shen, Cihang Xie, Alan Yuille, and Tao Kong. ibot: Image bert pre-training with online tokenizer. *arXiv preprint arXiv:2111.07832*, 2021. 1