Supplementary Material for AMU-Tuning: Effective Logit Bias for CLIP-based Few-shot Learning

Yuwei Tang*, Zhenyi Lin*, Qilong Wang†, Pengfei Zhu, Qinghua Hu
Tianjin Key Lab of Machine Learning, College of Intelligence and Computing, Tianjin University, China
{tangyuwei, linzhenyi, qlwang, zhupengfei, huqinghua}@tju.edu.cn

In the supplementary material, we conduct more experiments to further investigate the effectiveness of our AMU-Tuning method. Specifically, we first analyze the computation complexity of AMU-Tuning, and then compare the MTFi predictor with LP by using various backbone models. Subsequently, we conduct the analysis on the effect of hyper-parameters λ and ρ in the AMU-Tuning method. Finally, we compare different methods to compute the confidence κ in uncertainty fusion of Eq. (11). Note that all experiments are conducted on ImageNet-1K.

S1. Analysis on Computation Complexity.

In this section, we compare AMU-Tuning with its counterparts in terms of computation complexity on a single RTX 3090 GPU, including training time over 500 steps with batch size of 16384 and number of trainable parameters. Tab. S1 gives the results under 16-shot setting, where we can see that AMU-Tuning has the fastest training speed and the fewest trainable parameters, since AMU-Tuning only optimizes a lightweight LP. These results verify the efficiency of our AMU-Tuning.

Model	Time (s) ↓	Params. (M)↓
Tip-Adapter-F [7]	22.30	16.38
CaFo [8]	87.88	49.15
AMU-Tuning (Ours)	**6.63**	**2.05**

Table S1. Computational complexity of different methods in terms of training time (Time) and trainable parameters (Params.).

* Equal contributions made by Y. Tang and Z. Lin, † Corresponding author is Q. Wang. This work was supported in part by National Natural Science Foundation of China under Grants 62276186, 61925602, 62222608, in part by CAAI-Huawei MindSpore Open Fund under Grant CAAIXSJLJJ-2022-010 C, in part by Tianjin Natural Science Funds for Distinguished Young Scholar under Grant 23JCJQJC00270, and in part by the Haihe Lab of ITAI under Grant 22HHXCJC00002.

S2. Comparison of MTFi with LP Using Different Backbones

In this section, we conduct an in-depth analysis using extra backbones (i.e., DINO [1] and MAE [4]) to evaluate the effectiveness of MTFi on ImageNet-1K [3]. Specifically, Tab. S2 respectively shows the results of linear probing (LP) and our MTFi predictors with auxiliary features of DINO and MAE, while apparent that the implementation of MTFi leads to a remarkable boost in the model's performance.

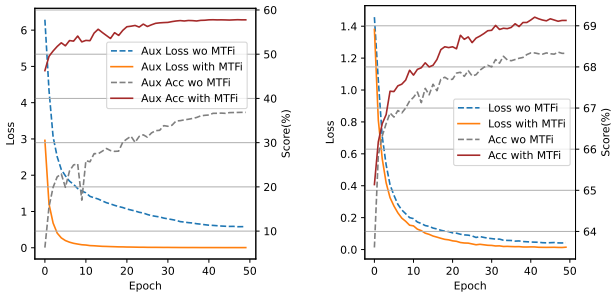
Method	1-shot	4-shot	16-shot
MAE [4]+LP	61.20	62.76	65.49
MAE + MTFi	**61.44**	**63.32**	**65.99**
DINO [1]+LP	61.62	64.43	68.32
DINO+MTFi	**61.97**	**65.05**	**69.21**

Table S2. Comparison(%) of MTFi in extra backbones.

Furthermore, we visualize the loss convergence curves for the DINO mode with MTFi trained in the 16-shot setting. As shown in Fig. S1 both ℓ_{total} and ℓ_{Aux} convergence faster, whlie the individual accuracy (%) of the auxiliary branch increases from 36.24 to 57.79, and the overall model accuracy (%) improves from 68.32 to 69.21. The above results demonstrate that our MTFi can achieve significant performance improvement while benefitting higher training efficiency (e.g., DINO).

S3. Effect of Parameter λ

In Eq. (9), we introduce a hyper-parameter λ that balance the effect between ℓ_{Aux} and ℓ_{Fusion}. In Fig. S2, we examine the performance of the AMU-Tuning method with 4-shot and 16-shot on ImageNet-1K under various values of λ. It is evident that the performance of AMU-Tuning method remains stable when λ is in the range of 0.2 to 0.6, with fluctuations below 0.05%. Therefore, we generally set λ to 0.4 throughout our experiments.



(a) Loss and accuracy of auxiliary branch (b) Loss and accuracy overall model

Figure S1. Comparison of loss and accuracy curves with/without the MTFi method. (a) is auxiliary loss and accuracy curve,while (b) is total loss and accuracy curve.

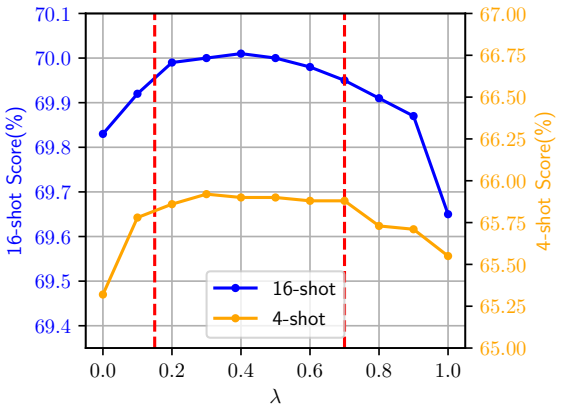


Figure S2. Comparison on AMU-Tuning with different λ.

S4. Effect of Parameter ρ

In Eq. (10), we introduce a hyper-parameter ρ to control the power of uncertainty. In Fig. S2, we examine the performance of the AMU model with 16-shot on ImageNet-1K under various values of ρ. It is evident that the performance of AMU performance stable when ρ is in the range of 0.2 to 0.6. In our experiments, we generally set ρ to 0.4.

S5. Comparison of Different Uncertainty-based Fusion Methods

Accroding to the observation in Sec. 3.2.3, we have that the largest logit value of zero-shot CLIP is consistently high, when the sample is correctly classified. Therefore, we devise various approaches for calculating the confidence parameter κ. First, we compute the confidence score κ based on the largest logit value directly which formularized as

$$\kappa_{\text{Max}} = Max1(\mathbf{s}_0)^{\rho}, \tag{S1}$$

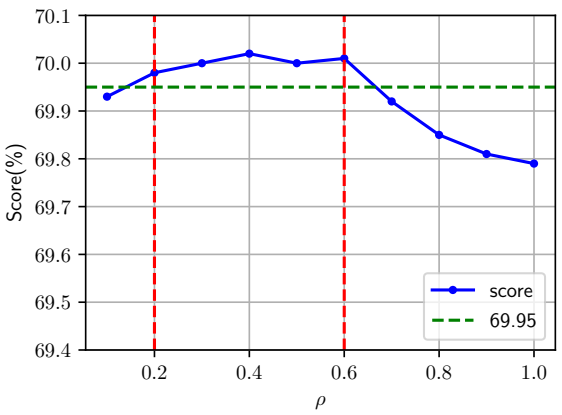


Figure S3. Comparison on AMU-Tuning with different ρ.

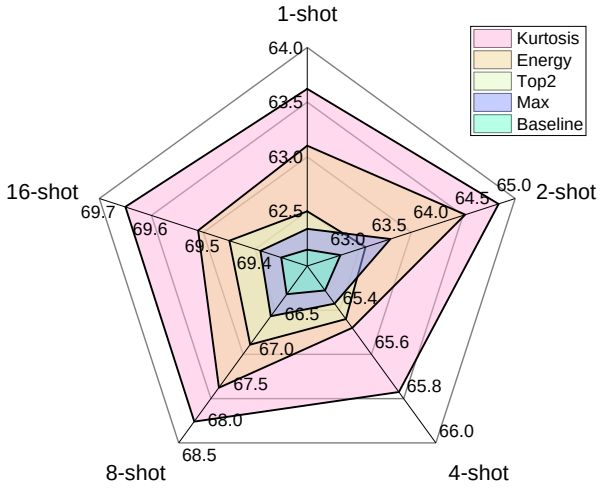


Figure S4. Comparison(%) of different confidence calculation methods.

where $Max1(\cdot)$ computes the largest logit value. Furthermore, we also design a method that simultaneously utilizes the largest and second largest logit values, as suggested in [6]. The formula is as follows:

$$\kappa_{\text{Top2}} = \left(\frac{Max1(\mathbf{s}_0) - Max2(\mathbf{s}_0)}{abs(Max1(\mathbf{s}_0) + Max2(\mathbf{s}_0))} \right)^{\rho}, \tag{S2}$$

$Max2(\cdot)$ computes the second largest logit value. We also explore a confidence calculation method based on energy [5], for C classes κ formulated as follows:

$$\kappa_{\text{Energy}} = \left(\log \sum_{i=1}^{C} e^{\mathbf{s}_0^i} \right)^{\rho}. \tag{S3}$$

We compare different methods in Eqs. (S1) to (S3) and our kurtosis-based confidence in Eq. (10) on ImageNet-1K with the auxiliary features of MoCov3 [2]. As present

in Fig. S4 several confidence computation methods can lead to performance improvement while our kurtosis-based approach achieves the better performance than other compared methods for all cases.

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 *ICCV*, pages 9650–9660, 2021. 1

[2] Xinlei Chen, Saining Xie, and Kaiming He. An empirical study of training self-supervised vision transformers. In *ICCV*, pages 9640–9649, 2021. 2

[3] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. ImageNet: A large-scale hierarchical image database. In *CVPR*, pages 248–255. Ieee, 2009. 1

[4] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. In *CVPR*, pages 16000–16009, 2022. 1

[5] Weitang Liu, Xiaoyun Wang, John Owens, and Yixuan Li. Energy-based out-of-distribution detection. In *Advances in Neural Information Processing Systems*, pages 21464–21475. Curran Associates, Inc., 2020. 2

[6] Abdel Aziz Taha, Leonhard Hennig, and Petr Knoth. Confidence estimation of classification based on the distribution of the neural network output layer. *arXiv preprint arXiv:2210.07745*, 2022. 2

[7] Renrui Zhang, Wei Zhang, Rongyao Fang, Peng Gao, Kunchang Li, Jifeng Dai, Yu Qiao, and Hongsheng Li. Tip-Adapter: Training-free adaption of clip for few-shot classification. In *ECCV*, pages 493–510. Springer, 2022. 1

[8] Renrui Zhang, Xiangfei Hu, Bohao Li, Siyuan Huang, Hanqiu Deng, Yu Qiao, Peng Gao, and Hongsheng Li. Prompt, generate, then cache: Cascade of foundation models makes strong few-shot learners. In *CVPR*, pages 15211–15222, 2023. 1