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,

Abstract

*Recently, pre-trained vision-language models (e.g., CLIP) have shown great potential in few-shot learning and attracted a lot of research interest. Although efforts have been made to improve few-shot ability of CLIP, key factors on the effectiveness of existing methods have not been well studied, limiting further exploration of CLIP's potential in few-shot learning. In this paper, we first introduce a unified formulation to analyze CLIP-based few-shot learning methods from a perspective of logit bias, which encourages us to learn an effective logit bias for further improving performance of CLIP-based few-shot learning methods. To this end, we disassemble three key components involved in computation of logit bias (i.e., logit features, logit predictor, and logit fusion) and empirically analyze the effect on performance of few-shot classification. Based on analysis of key components, this paper proposes a novel AMU-Tuning method to learn effective logit bias for CLIP-based few-shot classification. Specifically, our AMU-Tuning predicts logit bias by exploiting the appropriate **A**uxiliary features, which are fed into an efficient feature-initialized linear classifier with **M**ulti-branch training. Finally, an **U**ncertainty-based fusion is developed to incorporate logit bias into CLIP for few-shot classification. The experiments are conducted on several widely used benchmarks, and the results show AMU-Tuning clearly outperforms its counterparts while achieving state-of-the-art performance of CLIP-based few-shot learning without bells and whistles.*

1. Introduction

In recent years, large-scale vision-language models [1, 11, 27, 38, 43, 59, 62] have attracted large amounts of research attention in computer vision community (especially CLIP [43]), due to the remarkable performance on downstream tasks, e.g., zero-shot generalization ability [18, 56, 61]. Recently, some efforts have been made to improve the transfer learning ability of CLIP given a limited set of training samples [17, 63, 64, 66, 67], i.e., CLIP-based few-shot learning. These methods can be roughly divided into three categories: (1) prompt tuning [66, 67]; (2) adapter-based tuning [17, 63]; (3) cache-based tuning [63, 64]. Specifically, as illustrated in Fig. 1, prompt-tuning methods improve the few-shot learning ability of CLIP by introducing learnable text prompt [66, 67] for the text encoder of CLIP. For adapter-based tuning, some lightweight modules, e.g., multi-layer perceptron (MLP) [17], are built at the end of text and visual encoders to adjust text and visual features for downstream tasks. Subsequently, cache-based tuning methods [63, 64] present "soft" K-nearest neighbor classifiers storing visual features and labels of training samples, which are combined with zero-shot CLIP for final classification.

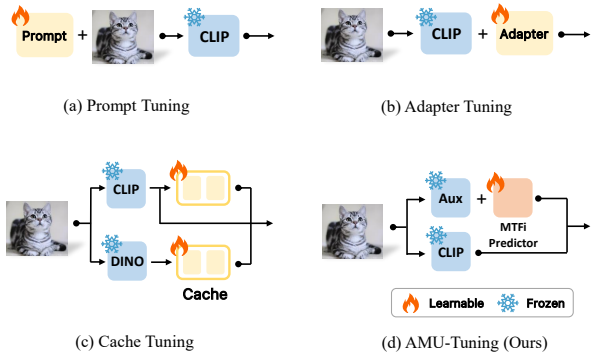


Figure 1. Comparison of the existing CLIP-based few-shot learning methods in terms of architecture design.

Although many works have been studied to improve the few-shot generalization ability of CLIP, the relationship among the existing methods seems a bit loose. More importantly, the key factors on the effectiveness of existing methods have not been well studied, which limits further exploring the potential of CLIP to few-shot learning. Therefore, this paper introduces a unified formulation to analyze CLIP-based few-shot learning methods from a perspective

* 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.

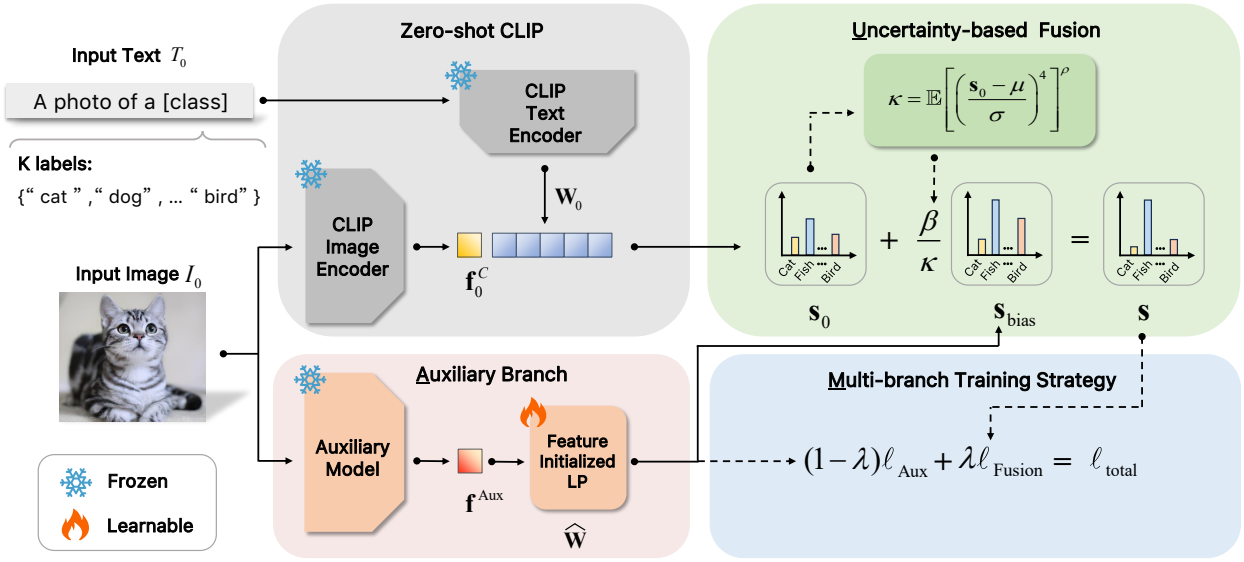


Figure 2. Overview of our proposed AMU-Tuning method for CLIP-based few-shot classification. Specifically, our AMU-Tuning exploits the complementary **A**uxiliary features to compute logit bias. Then, an efficient feature-initialized LP with **M**ulti-branch training is presented to improve performance of logit predictor by better exploring the auxiliary features. Finally, we develop a **U**ncertainty-based fusion by considering prediction confidence of zero-shot CLIP, which adaptively incorporates logit bias into CLIP for few-shot classification.

of logit bias, where we show most of the previous methods can be generally regarded as learning different logit biases for zero-shot CLIP. Meanwhile, logit bias dramatically impacts the performance of few-shot classification. It encourages us to learn an effective logit bias for further improving performance of CLIP-based few-shot learning methods.

According to the observations on previous methods from the perspective of logit bias, we disassemble three key components involved in logit bias (i.e., logit features, logit predictor, and logit fusion), while empirically analyzing the effect on few-shot classification. Specifically, we first compare several auxiliary features to predict logit bias in terms of complementary and superiority, while showing the appropriate features greatly help to learn effective logit bias. Then, we evaluate the effect of various logit predictors, e.g., MLP, cache-based model, and linear probing (LP), while showing feature initialization is helpful for logit predictor, but existing logit predictors do not fully explore the superiority of auxiliary features. Finally, we observe that trade-off parameter of fusion is very sensitive to models and datasets, which is related to prediction confidence of zero-shot CLIP.

Based on above analysis on the key components, we propose a novel AMU-Tuning method to learn effective logit bias for CLIP-based few-shot classification. Specifically, our AMU-Tuning exploits a kind of **A**uxiliary features complementary to CLIP for computing logit bias. Then, an efficient feature-initialized LP with **M**ulti-branch training is presented to improve the performance of logit predictor by better exploring the potential of auxiliary features. Finally, we develop a **U**ncertainty-based fusion by considering prediction confidence of zero-shot CLIP, which adaptively incorporates logit bias into CLIP for effective few-shot classification. The overview of our AMU-Tuning is shown in Fig. 2. To evaluate the effectiveness of our AMU-Tuning method, experiments are conducted on eleven downstream tasks [5, 12, 13, 16, 22, 30, 33, 37, 40, 50, 60], and four out-of-distribution benchmarks [23, 24, 46, 57] by using various backbone models (i.e., ResNets [19] and ViT [15]). The contributions of this work are summarized as follows:

- To our best knowledge, this work makes the first attempt to introduce a unified formulation for CLIP-based few-shot learning methods from a perspective of logit bias. It allows us to further explore the effectiveness of existing methods by analyzing the effect of three key components involved in logit bias, i.e., features, predictor, and fusion.
- Based on the analysis on the key components of logit bias, we propose an efficient AMU-Tuning method for CLIP-based few-shot classification, whose core is to learn effective logit bias by exploiting the appropriate **A**uxiliary features with **M**ulti-branch training of a feature-initialized linear classifier, followed by an **U**ncertainty-based fusion.
- Extensive experiments are conducted on several downstream tasks and out-of-distribution benchmarks, and the results show our proposed AMU-Tuning clearly outperforms its counterparts while achieving state-of-the-art performance of CLIP-based few-shot learning with more efficient computational cost.

Model	Bias	Feature	Predictor	Fusion	16-shot Acc (%)
Zero-shot CLIP [43]	-	-	-	-	60.33
CoOp [67]	$\simeq f_T(T_{\text{bias}})\mathbf{f}_0^C$	T_{bias}	f_T	-	62.95
CLIP-Adapter [17]	$f_T^{\text{Ada}}(\mathbf{W}_0)\mathbf{f}_0^C + \mathbf{W}_0 f_V^{\text{Ada}}(\mathbf{f}_0^C) + f_T^{\text{Ada}}(\mathbf{W}_0) f_V^{\text{Ada}}(\mathbf{f}_0^C)$	CLIP	MLP	Manual Tuning	63.59
Tip-Adapter-F [63]	$\phi\left(\mathbf{F}_{\text{TrC}}^{\top}\mathbf{f}_0^C\right)\mathbf{V}$	CLIP	Cache	Manual Tuning	65.51
CaFo [64]	$\alpha\phi\left(\mathbf{F}_{\text{TrC}}^{\top}\mathbf{f}_0^C\right)\mathbf{V} + (1-\alpha)\phi\left(\mathbf{F}_{\text{TrD}}^{\top}\mathbf{f}_0^D\right)\mathbf{V}$	CLIP+DINO	Cache	Similarity-based	68.79
AMU-Tuning (Ours)	$\widehat{\mathbf{W}}\mathbf{f}^{\text{Aux}}$	Aux	MTFi LP	Uncertainty-based	**70.02**

Table 1. Comparison of existing CLIP-based few-shot learning methods from the perspective of logit bias. Different from previous works, our AMU-Tuning learns logit bias by exploiting the appropriate auxiliary (Aux) features with multi-branch training feature-initialized (MTFi) LP followed by an uncertainty-based fusion, while achieving higher accuracy (Acc) on ImageNet-1K with 16-shot training samples.

2. Related Work

Few-shot Classification Few-shot learning is crucial for limited-sample scenarios, with extensive exploration in this field [3, 4, 29, 36, 39, 47–49]. Recently, there's been a surge in few-shot learning methods that fine-tune large pre-trained models while CLIP [43] makes a seminal work to train a large-scale vision-language model, showing good generalization on various downstream tasks. Subsequently, several works have been studied to improve few-shot generalization ability of CLIP. CoOp [67] first shows text prompt greatly impacts on zero-shot performance of CLIP and introduces the idea of prompt learning to improve few-shot performance of CLIP. After that, a lot of works are proposed to improve the effectiveness of prompt learning [8, 26, 66, 67]. However, prompt learning methods need to compute the gradients throughout text encoder, suffering from the heavy computational cost. Consequently, CLIP-Adapter [17] propose a residual structure and MLP-based adapter modules to fine-tune outputs of text and visual encoders. Tip-Adapter [63] presents a cache structure to perform a "soft" K-nearest neighbor classifier, which is combined with zero-shot CLIP. Going beyond Tip-Adapter, CaFo [64] introduces an extra cache structure with DINO [7], while employing GPT-3 [6] and DALL-E [45] models for text and visual information augmentation. Different from above works, our AMU-Tuning learns logit bias by exploiting the appropriate auxiliary features with multi-branch training feature-initialized LP followed by an uncertainty-based fusion, while achieving better performance.

Large-scale Pre-trained Models Inspired by the success of large-scale pre-trained models in the field of natural language processing [6, 14, 38, 44], many researchers have undertaken various explorations in pre-training visual models in the domain of computer vision, including ResNet [19], ViT [15], Swin Transformers [31], and others [32, 55, 58]. In these large-scale pre-trained models, large-scale vision-language models represented by CLIP [43], SLIP [34], and CoCa [62] have explored training on massive data of images and text from the internet. These models showcase powerful generalization abilities and have achieved remarkable performance in downstream tasks. Recently, a class of methods [9, 10, 20, 53, 54] based on self-supervised learning has garnered significant attention for substantially improving the transfer performance of visual models. For instance, MoCo [20] introduced a momentum-contrast framework, which reduced the constraints on batch size during training, leading to enhanced model transferability. Following that, a class of methods based on Masked Image Modeling (MIM) was devised. These methods involve masking a portion of an image and training the model to reconstruct these masked regions, achieving notable generalization performance. For example, MAE [21] achieves pixel reconstruction through masking, while BEiT [2] enhances model performance by designing a pretext task for visual tokens. Numerous other MIM-based approaches have been proposed [25, 52, 59, 65], significantly advancing the improvement of model transferability. In our work, we aim to leverage large-scale pre-trained models as auxiliary features to collaboratively enhance few-shot generalization of CLIP.

3. Proposed Method

In this section, we first summarize and compare the existing methods from a perspective of logit bias. Then, we empirically analyze three key components involved in computation of logit bias. Based on the analysis, we propose an efficient and effective AMU-Tuning method.

3.1. Perspective of Logit Bias for CLIP-based Few-shot Learning

To analyze the existing methods in a unified framework, we summarize them from a perspective of logit bias. As listed in Tab. 1, we formulate previous works [17, 63, 64, 67] as

$$\mathbf{s} \cong \mathbf{W}_0\mathbf{f}_0^C + \beta \cdot \mathbf{s}_{\text{bias}}, \tag{1}$$

where $\mathbf{W}_0 = f_T(T_0)$ and $\mathbf{f}_0^C = f_V(I_0)$ indicate the outputs of text encoder f_T and visual encoder f_V of CLIP [43], respectively. T_0 and I_0 are text and visual inputs of zero-shot

Model	SUP_{Aux} (%)	CMY_{Aux}	Fusion (%)
ZS-CLIP [43]	N/A	N/A	60.33
CLIP [43]	56.93	0.438	65.34
DINO [7]	55.65	<u>0.816</u>	68.32
MoCov3 [10]	<u>57.68</u>	**0.837**	**69.35**
MAE [21]	38.98	0.722	65.49
SparK [52]	28.31	0.770	63.56
MILAN [25]	**66.36**	0.718	<u>69.24</u>

Table 2. Comparison of different auxiliary features in terms of complementary (CMY_{Aux}), superiority (SUP_{Aux}) and fused results on ImageNet-1K with 16-shot samples. The best and second-best results are highlighted in **bold** and <u>underline</u>, respectively.

CLIP, respectively. $\mathbf{s}_0 = \mathbf{W}_0\mathbf{f}_0^C$ represents the prediction logit of zero-shot CLIP. $\mathbf{s}_{\text{bias}}$ means the logit bias learned by few-shot training samples, which is fused with $\mathbf{s}_0$ to obtain the final prediction $\mathbf{s}$. β is a hyper-parameter of fusion.

Specifically, from Tab. 1 we can see that prompt tuning [66, 67] can be approximated by combining zero-shot CLIP (i.e., $\mathbf{s}_0$) with $f_T(T_{\text{bias}})\mathbf{f}_0^C$, where $\mathbf{s}_{\text{bias}}$ is computed based on the learnable text prompts (i.e., T_{bias}). Clearly, these prompt-tuning methods require to compute the gradients throughout the text encoder f_T, suffering from the heavy computational cost. CLIP-Adapter [17] calculates $\mathbf{s}_{\text{bias}}$ by

$$f_T^{\text{Ada}}(\mathbf{W}_0)\mathbf{f}_0^C + \mathbf{W}_0 f_V^{\text{Ada}}(\mathbf{f}_0^C) + f_T^{\text{Ada}}(\mathbf{W}_0) f_V^{\text{Ada}}(\mathbf{f}_0^C), \quad (2)$$

where $f_T^{\text{Ada}}(\cdot)$ and $f_V^{\text{Ada}}(\cdot)$ are adapters for text and visual features, which are achieved by two MLP. Cache-based Tip-Adapter [63] and CaFo [64] perform a "soft" K-nearest neighbor classifier on a trainable cache of visual CLIP features ($\mathbf{F}_{\text{TrC}}$) to generate $\mathbf{s}_{\text{bias}}$, i.e.

$$\phi\left(\mathbf{F}_{\text{TrC}}^{\mathsf{T}}\mathbf{f}_0^C\right)\mathbf{V}, \quad (3)$$

where $\phi(\cdot)$ is a non-linear function, and $\mathbf{V}$ is the label matrix. T indicates the matrix transposition. Besides, CaFo exploits an extra trainable cache of visual DINO features [7] ($\mathbf{F}_{\text{TrD}}$) to compute $\mathbf{s}_{\text{bias}}$ as

$$\alpha\phi\left(\mathbf{F}_{\text{TrC}}^{\mathsf{T}}\mathbf{f}_0^C\right)\mathbf{V} + (1-\alpha)\phi\left(\mathbf{F}_{\text{TrD}}^{\mathsf{T}}\mathbf{f}_0^D\right)\mathbf{V}, \quad (4)$$

where α is a trade-off parameter computed based on the similarity with $\mathbf{s}_0$. The analysis above shows that most of the existing methods can be regarded as learning different logit biases to adjust the prediction of zero-shot CLIP by using few-shot training samples.

3.2. Analysis on Computation of Logit Bias

By regarding the performance of previous works (the last column of Tab. 1), all methods are superior to zero-shot CLIP by introducing logit bias. Particularly, CLIP-Adapter [17] performs better than prompt tuning [67], while cache-based methods [63, 64] significantly outperform CLIP-Adapter. These comparisons clearly demonstrate that different logit biases greatly influence the performance of few-shot classification. By observing the existing methods in Tab. 1, we disassemble the computation of logit bias into three key components, i.e., logit features, logit predictor, and logit fusion. To deeply analyze the effect of logit bias on the performance of few-shot classification, we conduct comprehensive empirical comparisons on three key components in the following.

3.2.1 Features for Computation of Logit Bias

Existing works mainly exploit CLIP itself to compute logit bias. Recently, CaFo [64] proposes to use the extra features [6, 7, 45] to help generating logit bias, which achieves remarkable performance gain. In this paper, we call such extra features as auxiliary features. To evaluate the effect of auxiliary features, this paper conducts an empirical comparison on several kinds of auxiliary features in terms of complementary and superiority. Specifically, we conduct experiments on ImageNet-1K [13] by using CLIP with the backbone of ResNet-50 (RN50) and 16-shot training samples. For auxiliary features, we compare six pre-trained models, i.e., the visual encoder of CLIP, DINO [7], MoCov3 [10], MAE [21], SparK [52], and MILAN [25]. The backbone for all pre-trained models is RN50, except MAE and MILAN. We utilize VIT-B/16 [15] models for MAE and MILAN, due to the unavailability of pre-trained RN50.

To compute the logit bias, we train a simple LP for all auxiliary features. Then, the logit bias is combined with prediction of zero-shot CLIP by summation for few-shot classification. Particularly, we train an individual LP for all auxiliary features within 50 epochs, whose results represent the superiority of different auxiliary features (indicated by SUP_{Aux}). For measuring the complementarity (CMY_{Aux}) of different auxiliary features, we define CMY_{Aux} by inverse of similarity between LP prediction of auxiliary features ($\mathbf{s}_{\text{Aux}}$) and prediction of zero-shot CLIP ($\mathbf{s}_0$):

$$\begin{aligned} \text{CMY}_{\text{Aux}} &= 1 - \text{SIM}(\mathbf{s}_0, \mathbf{s}_{\text{Aux}}), \\ \text{SIM}(\mathbf{s}_0, \mathbf{s}_{\text{Aux}}) &= \frac{\mathbf{s}_0 \cdot \mathbf{s}_{\text{Aux}}}{\|\mathbf{s}_0\|_2 \cdot \|\mathbf{s}_{\text{Aux}}\|_2}, \end{aligned} \quad (5)$$

where SIM computes the cosine similarity between $\mathbf{s}_{\text{Aux}}$ and $\mathbf{s}_0$. Clearly, smaller similarity means less correlation between $\mathbf{s}_{\text{Aux}}$ and $\mathbf{s}_0$, indicating the auxiliary features may be more complementary to zero-shot CLIP.

As compared in Tab. 2, we can see that all auxiliary features contribute performance gains to zero-shot CLIP (ZS-CLIP). In particular, we observe that complementarity (CMY_{Aux}) is more important than superiority (SUP_{Aux}) for auxiliary features. For example, CLIP is superior to DINO, but the result of fusion with DINO is much better than one of fusing CLIP, because DINO has much higher CMY_{Aux}.

Auxiliary Features	Individual	Joint	Joint+ZO
CLIP [43]	56.93	11.23	65.34
DINO [7]	55.65	36.24	68.32
MoCov3 [10]	57.68	42.82	69.35

Table 3. Comparison (%) of two training strategies (i.e., Individual and Joint) for the bias branch on ImageNet-1K. Joint+ZO indicates the fused results of joint training bias branch with zero-shot CLIP.

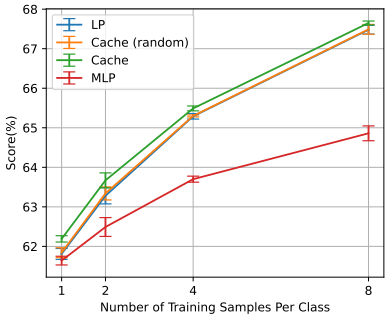
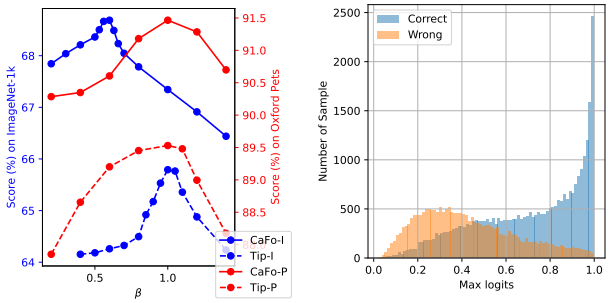


Figure 3. Results of different logit predictors on ImageNet-1K.

Besides, fusion of MAE achieves similar result with fusion of CLIP, although MAE has much lower SUP_{Aux}. Additionally, MILAN is superior to MoCov3, but fusion of MoCov3 achieves better final results due to its higher CMY_{Aux}. Another observation is that the auxiliary features with higher SUP_{Aux} achieve better fusion results, when they have similar CMY_{Aux}. For example, DINO and MoCov3 have similar CMY_{Aux}, and MoCov3 outperforms DINO in terms of fusion result, as MoCov3 has higher SUP_{Aux} than DINO. Based on the above results, we conclude that *the auxiliary features with good complementarity and superiority can greatly help to compute effective logit bias for better performance. Particularly, complementarity is more important than superiority for auxiliary features*.

3.2.2 Logit Predictor

As discussed in Sec. 3.1 and Tab. 1, existing works mainly exploit MLP [17] and cache-based classifiers [63, 64] to predict the logit bias $\mathbf{s}_{\text{bias}}$. To evaluate the effect of logit predictor, we empirically compare with several predictors, including MLP, Cache, Cache with random initialization (Cache-Random), and a simple linear probing (LP) as baseline. Specifically, we conduct experiments on ImageNet-1K by using various shots of training samples by following the same settings in Sec. 3.2.1. All logit predictors exploit the outputs from the RN50 model of MoCov3 [10] as the input features. As shown in Fig. 3, LP achieves similar performance with Cache-Random, and both of them are clearly superior to MLP. Particularly, LP is more efficient



(a) Results with various β (b) Visualization of max logits

Figure 4. (a) Results of Tip-Adapter-F and CaFo with various β on ImageNet-1K and OxfordPets. (b) Visualization of the distribution of max logits for zero-shot CLIP on ImageNet-1K.

than Cache-Random, especially for large-shot settings. The original Cache method with feature initialization achieves better performance than LP and Cache-Random, indicating that feature initialization is helpful for the logit predictor.

Furthermore, we analyze the effect of logit predictor by comparing results of the bias branch under two training strategies, i.e., individual training of bias branch and joint training of bias branch with zero-shot CLIP. Particularly, we use a simple LP as logit predictor for efficiency. The results with 16-shot training samples on ImageNet-1K are given in Tab. 3, where we can see that the individually trained bias branch (Individual) is significantly superior to one jointly trained with zero-shot CLIP (Joint) for all logit features. Clearly, the joint training strategy makes logit bias as a pure supplement to zero-shot CLIP by considering the complementarity of auxiliary features, but it cannot fully explore the superiority of auxiliary features. Based on the above results, we conclude that *feature initialization is helpful for logit predictor, while existing logit predictors do not fully explore the superiority of auxiliary features*.

3.2.3 Logit Fusion

To fuse logit bias with zero-shot CLIP, a manually tuned parameter β is used to control the effect of logit bias. As illustrated in Fig. 4a, we show the 16-shot results of Tip-Adapter [63] and CaFo [64] with various β on ImageNet-1K and OxfordPets datasets, where their performance is heavily affected by β, while β is sensitive to different methods and datasets. To further analyze the effect of logit fusion, we visualize the distribution of max logits for zero-shot CLIP on ImageNet-1K. As shown in Fig. 4b, zero-shot CLIP tends to correctly classify the samples with large max logits, which indicates logits with higher confidence of zero-shot CLIP generally results in more precise classification. On the contrary, zero-shot CLIP usually mis-classifies the samples with logits of lower confidence. Therefore, we can increase

effect of logit bias (i.e., value of β) for samples with low-confidence predictions of zero-shot CLIP. Based on above results, we conclude that *trade-off parameter greatly affects performance of fusion, while prediction confidence of zero-shot CLIP can be regarded as an indicator of logit fusion*.

3.3. Learning Bias via AMU

Based on above analysis, we propose a novel AMU-Tuning method to learn effective logit bias based on appropriate auxiliary features, effective logit predictor and adaptive logit fusion. Particularly, the overview of our AMU-Tuning is illustrated in Fig. 2, whose details are given as follows.

Auxiliary Features $\mathbf{f}^{\text{Aux}}$ According to the conclusion in Sec. 3.2.1, we can seek the optimal auxiliary features $\mathbf{f}^{\text{Aux}}$ from a group of feature candidates based on the metrics of superiority and complementarity (Eq. (5)). Specifically, we employ a certain of features lying in $\Omega_S^{\text{Top-K}} \cap \Omega_C^{\text{Top-M}}$ for various downstream tasks, where $\Omega_S^{\text{Top-K}}$ and $\Omega_C^{\text{Top-M}}$ indicate the sets of features with Top-K superiority and Top-M complementarity, respectively. For efficiency, we adopt MoCov3 model with the backbone of RN50 to obtain the auxiliary features $\mathbf{f}^{\text{Aux}}$ with no special declaration.

Multi-branch Training of Feature-initialized (MTFi) Logit Predictor As shown in Sec. 3.2.2, LP is more efficient than cache-based predictor, but the latter achieves better performance with the help of feature initialization. It encourages us to propose a feature-initialized LP for improving efficiency and effectiveness of logit predictor. Specifically, under C-way-N-shot setting with C classes and N samples of each class, we initialize the weights $\widehat{\mathbf{W}}$ of LP by using the mean of auxiliary features from different classes:

$$\begin{aligned} \widehat{\mathbf{W}}_0 &= [\mathbf{m}_1, \mathbf{m}_2, \cdots, \mathbf{m}_C]^{\mathsf{T}}, \\ \mathbf{m}_i &= \frac{1}{N} \sum_{j=1}^{N} \mathbf{f}_{ij}^{\text{Aux}}, \quad i = \{1, 2, \cdots, C\}, \end{aligned} \tag{6}$$

where $\widehat{\mathbf{W}}_0$ is the initialization of $\widehat{\mathbf{W}}$, and $\mathbf{f}_{ij}^{\text{Aux}}$ is the j-th feature of i-th class. As such, our feature-initialized LP predicts logit bias $\mathbf{s}_{\text{bias}}$ for j-th training sample as

$$\mathbf{s}_{\text{bias}}^{j} = \widehat{\mathbf{W}} \mathbf{f}_{j}^{\text{Aux}}. \tag{7}$$

Furthermore, we propose a multi-branch training strategy to fully explore the superiority of auxiliary features. Specifically, besides the original classification loss (i.e., ℓ_{Fusion}) based on the fused logit $\mathbf{s}$, we introduce an extra training branch to minimize the cross-entropy loss between logit bias $\mathbf{s}_{\text{bias}}$ and the ground-truth label of $\mathbf{y}$ as

$$\ell_{\text{Aux}} = -\sum_{j=1}^{C \times N} \mathbf{y}_j \cdot \log(g(\mathbf{s}_{\text{bias}}^{j})), \tag{8}$$

where $g(\cdot)$ is a softmax function. As such, the total loss of our multi-branch training can be formulated as:

$$\ell_{\text{total}} = (1 - \lambda)\ell_{\text{Aux}} + \lambda \ell_{\text{Fusion}}, \tag{9}$$

where λ is a hyper-parameter to balance effect of ℓ_{Fusion} and ℓ_{Aux}. From Eq. (6) and Eq. (8), we can see than our proposed MTFi logit predictor benifits feature initialization in a more efficient way, while exploring the superiority of auxiliary features by introducing an extra training branch ℓ_{Aux}.

Uncertainty-based Fusion Based on the analysis in Sec. 3.2.3, the hyper-parameter β of bias fusion is very sensitive to models and datasets. Meanwhile, such hyper-parameter is related to prediction confidence of zero-shot CLIP. Therefore, we present an uncertainty-based fusion to adaptively combine zero-shot CLIP with logit bias based on prediction confidence of zero-shot CLIP. Specifically, we introduce an uncertainty (κ) based on Kurtosis (i.e., the fourth moment) [51] to represent prediction confidence as

$$\kappa = \mathbb{E}\left[\left(\frac{\mathbf{s}_0 - \mu}{\sigma}\right)^4\right]^{\rho}, \tag{10}$$

where μ and σ are the mean and the standard deviation of $\mathbf{s}_0$, respectively. ρ is a parameter to control the power of uncertainty. As such, we can adopt κ to balance effect of logit bias. Specifically, we increase effect of logit bias for small κ; otherwise, effect of logit bias is decreased. In conclusion, our AMU-Tuning method can be formulated as

$$\mathbf{s} = \mathbf{s}_0 + \frac{\beta}{\kappa} \widehat{\mathbf{W}} \mathbf{f}^{\text{Aux}}, \tag{11}$$

where only a lightweight LP with the parameters of $\widehat{\mathbf{W}}$ is optimized by the loss ℓ_{total} (Eq. (9)).

4. Experiments

Here, we first describe implementation details of our AMU-Tuning, and then compare with state-of-the-arts (SOTA) on downstream tasks and out-of-distribution (OOD) benchmarks. Finally, we conduct ablation study on ImageNet-1K.

4.1. Implementation Details

Following previous works [63, 64], we evaluate the effectiveness of our AMU-Tuning with [1, 2, 4, 8, 16]-shot training samples on eleven downstream tasks, including ImageNet-1K [13], StandfordCars [30], Caltech101 [16], UCF101 [50], Flowers102 [37], Food101 [5], DTD [12], EuroSAT [22], FGVCAircraft [33], OxfordPets [40], and SUN397 [60]. Specifically, we trained our AMU-Tuning model (i.e., a lightweight LP) on all downstream tasks within 50 epochs, where the AdamW optimizer is used with the initial learning rate of 0.001 and batch size of 8. The

Method	Score				
	1-shot	2-shot	4-shot	8-shot	16-shot
LP-CLIP [43]	22.17	31.90	41.20	49.52	56.13
CoOp [67]	57.15	57.81	59.99	61.56	62.95
CLIP-Adapter [17]	61.20	61.52	61.84	62.68	63.59
VT-CLIP [42]	60.53	61.29	62.02	62.81	63.92
Tip-Adapter-F [63]	61.32	61.69	62.52	64.00	65.51
CaFo [64]	**63.80**	**64.34**	65.64	66.86	68.79
CaFo* [64]	61.58	62.76	64.31	66.25	68.05
AMU-Tuning (Ours)	62.60	64.25	**65.92**	**68.25**	**70.02**

Table 4. Comparison (in %) of different SOTA methods on ImageNet-1K under various few-shot settings.

Dataset	Source	Target			
	IN-1K	v2	-S	-A	-R
ZS-CLIP [43]	60.33	53.27	35.44	21.65	56.00
CoOp [67]	62.95	55.40	34.67	23.06	56.60
CLIP-Adapter [17]	63.59	55.69	35.68	-	-
Tip-Adapter-F [63]	65.51	57.11	36.00	-	-
CaFo [64]	68.79	57.99	39.43	-	-
AMU-Tuning (RN50)	**70.02**	**58.64**	**40.04**	**25.65**	**57.10**
CoCoOp [66]	71.02	64.20	47.99	49.71	75.21
MaPLe [28]	70.72	64.07	49.15	50.90	76.98
AMU-Tuning (ViT)	**74.98**	**65.42**	**50.37**	**52.05**	**78.09**

Table 5. Comparison (%) of different methods under OOD setting.

hyper-parameters of λ in Eq. (8) and ρ in Eq. (10) are decided by cross-validation on the validation sets. All program is implemented by PyTorch [41]/ MindSpore and run on a single RTX 3090 GPU. Source code is available at https://github.com/TJU-sjyj/AMU-Tuning.

4.2. Comparison with SOTA Methods

Results on Downstream Tasks. To verify the effectiveness of our AMU-Tuning method, we first compare with several SOTA CLIP-based few-shot classification methods with backbone of RN50 on ImageNet-1K, including CoOp [67], CLIP-adapter [17], VT-CLIP [42], Tip-Adapter-F [63] and CaFo [64]. Particularly, we implement a CaFo variant (namely CaFo*) by excluding use of the extra DALL-E and GPT-3. The results are shown in Tab. 4, where our AMU-Tuning clearly outperforms all SOTA methods (except CaFo) under various few-shot settings. Comparing with CaFo, our AMU-Tuning achieves better performance under larger-shot settings (i.e., $>$ 4-shot). For smaller-shot settings (i.e., 1-shot and 2-shot), our AMU-Tuning is slightly inferior to CaFo. The reason behind this phenomenon lies in that CaFo uses the extra training samples generated by DALL-E, which contributes to large gains for smaller-shot settings. Comparing with CaFo*, our AMU-Tuning achieves 1.02%, 1.49%, 1.61%, 1.39% and 1.97% for [1, 2, 4, 8, 16]-shot training samples, respectively. Furthermore, we compare with SOTA methods on ten additional downstream tasks, which cover various scenarios, e.g., objects, satellite photos, textures, and scene images. As illustrated in Fig. 5, our AMU-Tuning achieves the best results on most of downstream tasks. Meanwhile, Fig. 6 shows AMU-Tuning clearly outperforms Tip-Adapter [63] and CaFo* on average over eleven downstream tasks. These results above clearly demonstrate the effectiveness and strong generalization of our AMU-Tuning.

Component			Score (%)		
AUX	MTFi	UF	1-shot	4-shot	16-shot
Baseline			61.16	62.33	65.34
✓			62.15	65.31	69.35
	✓		61.83	63.16	66.17
		✓	61.70	63.08	65.90
✓	✓		62.35	65.61	69.72
✓	✓	✓	**62.60**	**65.92**	**70.02**

Table 6. Results of AMU-Tuning with various modules on IN-1K.

Robustness to OOD. Following previous works [66, 67], we further verify the robustness of AMU-Tuning to OOD benchmarks. Specifically, we directly adopt the models fine-tuned on ImageNet-1K (IN-1K) with 16-shot training samples to four OOD benchmarks, including ImageNet-V2 [46], ImageNet-Sketch [57], ImageNet-A [24], and ImageNet-R [23]. As shown in Tab. 5, our AMU-Tuning achieves the best results on all OOD benchmarks, which can transfer performance gains on IN-1K to OOD setting. Since CLIP models are trained on a mass of external image-text pairs, potentially suffering from the risk of data leakage. However, previous works [35, 43] show the overlapping data brings little effect on performance, perhaps the self-supervised settings (without ground-truth labels of samples). Besides, all of our compared methods are built upon CLIP model, these results above verify that our AMU-Tuning is robust to OOD setting.

4.3. Ablation study

Our AMU-Tuning method involves three key components, i.e., appropriate auxiliary features (AUX), multi-branch training feature-initialized (MTFi) logit predictor, and uncertainty-based fusion (UF). To evaluate effect of different modules on AMU-Tuning, we conduct ablation studies on ImageNet-1K (IN-1K) dataset. Specifically, Tab. 6 shows the results of AMU-Tuning with various components, where the baseline is built based on auxiliary features

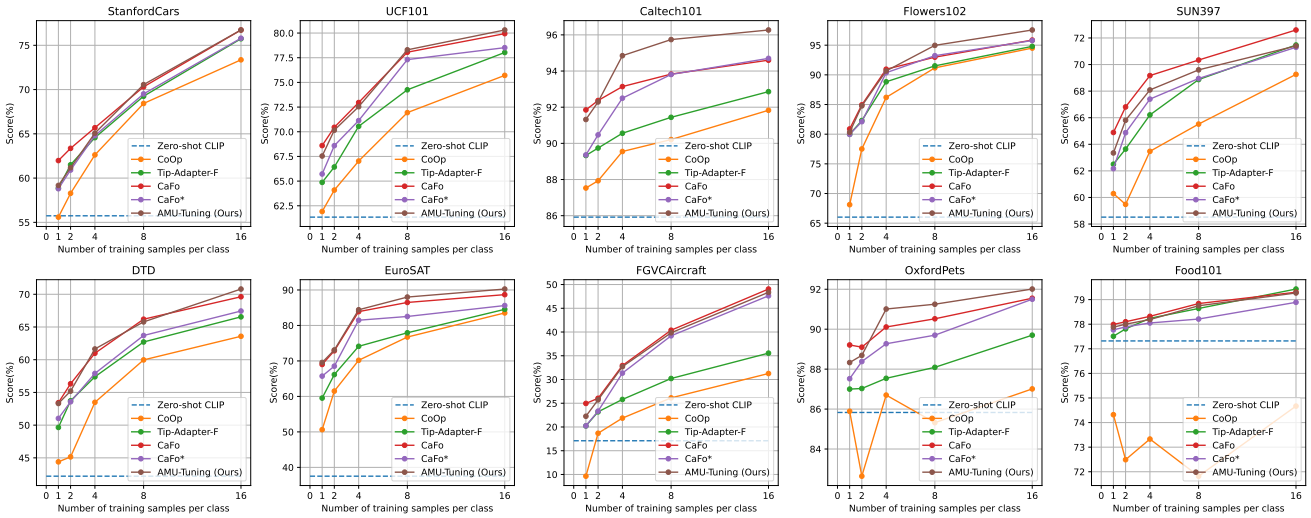


Figure 5. Comparison (in %) of different SOTA methods under various few-shot settings on ten downstream tasks.

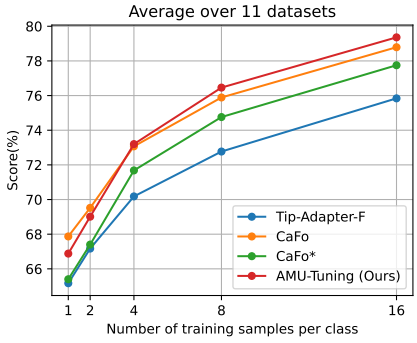


Figure 6. Results on eleven downstream tasks by average.

Models	Backbone			
	RN50	RN101	ViT-B/32	ViT-B/16
ZS-CLIP [43]	60.33	65.53	63.80	68.73
CoOp [67]	62.95	66.60	66.85	71.92
CLIP-Adapter [17]	63.59	65.39	66.19	71.13
Tip-Adapter-F [63]	65.51	68.56	68.65	73.69
CaFo [64]	68.79	70.82	70.82	74.48
CaFo* [64]	68.03	70.21	70.44	74.11
AMU-Tuning (Ours)	**70.02**	**71.58**	**71.65**	**74.98**

Table 7. Comparison (%) of SOTA methods with different visual encoders of CLIP on IN-1K with 16-shot training samples.

of CLIP with simple LP following by β-fusion. From Tab. 6 we can see that all components bring clear performance gains over the baseline. Besides, going beyond our strong baseline with the auxiliary features of MoCov3, MTFi and UF modules further bring 0.5%$\sim$0.7% gains, while achieving SOTA performance.

Furthermore, we compare AMU-Tuning with SOTA methods by using various visual encoders of CLIP, including RN50, RN101, ViT-B/32 and ViT-B/16. Note that our AMU-Tuning utilizes a RN50 pre-trained by MoCov3 to obtain the auxiliary features. As listed in Tab. 7, AMU-Tuning consistently outperforms all compared methods for different visual encoders of CLIP. Particularly, although existing methods (e.g., Tip-Adapter and CaFo) employ stronger visual encoders (e.g., ViT-B) to compute logit bias, AMU-Tuning can obtain better results by using RN50-based auxiliary features. These results above verify strong generalization of our AMU-Tuning again.

5. Conclusion

To better understand the effectiveness of existing CLIP-based few-shot learning methods, this work first introduces a unified formulation from the perspective of logit bias. Then, we disassemble computation of logit bias into three key components, i.e., logit features, logit predictor, and logit fusion, and analyze the effect on performance of few-shot classification. Furthermore, our empirical analysis encourages us to propose AMU-Tuning method for effective CLIP-based few-shot classification, which learns logit bias by exploiting the appropriate auxiliary features with multi-branch training feature-initialized LP following by an uncertainty-based fusion. Extensive experiments on both downstream tasks and out-of-distribution datasets demonstrate the effectiveness of our method. We hope our analysis from the perspective of logit bias could provide a new insight for CLIP-based few-shot learning, while encouraging more effective logit bias learning methods.

References

[1] Jean-Baptiste Alayrac, Jeff Donahue, Pauline Luc, Antoine Miech, Iain Barr, Yana Hasson, Karel Lenc, Arthur Mensch, Katherine Millican, Malcolm Reynolds, Roman Ring, Eliza Rutherford, Serkan Cabi, Tengda Han, Zhitao Gong, Sina Samangooei, Marianne Monteiro, Jacob L. Menick, Sebastian Borgeaud, Andy Brock, Aida Nematzadeh, Sahand Sharifzadeh, Mikolaj Binkowski, Ricardo Barreira, Oriol Vinyals, Andrew Zisserman, and Karén Simonyan. Flamingo: a visual language model for few-shot learning. In *NeurIPS*, 2022. 1

[2] Hangbo Bao, Li Dong, Songhao Piao, and Furu Wei. BEiT: BERT pre-training of image transformers. In *ICLR*, 2022. 3

[3] Peyman Bateni, Raghav Goyal, Vaden Masrani, Frank Wood, and Leonid Sigal. Improved few-shot visual classification. In *CVPR*. IEEE, 2020. 3

[4] Peyman Bateni, Jarred Barber, Jan-Willem van de Meent, and Frank Wood. Enhancing few-shot image classification with unlabelled examples. In *WACV*, 2022. 3

[5] Lukas Bossard, Matthieu Guillaumin, and Luc Van Gool. Food-101–Mining discriminative components with random forests. In *ECCV*, pages 446–461. Springer, 2014. 2, 6

[6] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *NeurIPS*, 33, 2020. 3, 4

[7] 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. 3, 4, 5

[8] Guangyi Chen, Weiran Yao, Xiangchen Song, Xinyue Li, Yongming Rao, and Kun Zhang. PLOT: prompt learning with optimal transport for vision-language models. In *ICLR*, 2023. 3

[9] Xinlei Chen, Haoqi Fan, Ross Girshick, and Kaiming He. Improved baselines with momentum contrastive learning. *arXiv preprint arXiv:2003.04297*, 2020. 3

[10] Xinlei Chen, Saining Xie, and Kaiming He. An empirical study of training self-supervised vision transformers. In *ICCV*, pages 9640–9649, 2021. 3, 4, 5

[11] Xi Chen, Xiao Wang, Soravit Changpinyo, A. J. Piergiovanni, Piotr Padlewski, Daniel Salz, Sebastian Goodman, Adam Grycner, Basil Mustafa, Lucas Beyer, Alexander Kolesnikov, Joan Puigcerver, Nan Ding, Keran Rong, Hassan Akbari, Gaurav Mishra, Linting Xue, Ashish V. Thapliyal, James Bradbury, and Weicheng Kuo. PaLI: A jointly-scaled multilingual language-image model. In *ICLR*, 2023. 1

[12] Mircea Cimpoi, Subhransu Maji, Iasonas Kokkinos, Sammy Mohamed, and Andrea Vedaldi. Describing textures in the wild. In *CVPR*, pages 3606–3613, 2014. 2, 6

[13] 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. 2, 4, 6

[14] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: pre-training of deep bidirectional transformers for language understanding. In *NAACL-HLT*, pages 4171–4186. Association for Computational Linguistics, 2019. 3

[15] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *ICLR*, 2021. 2, 3, 4

[16] Li Fei-Fei, Rob Fergus, and Pietro Perona. Learning generative visual models from few training examples: An incremental bayesian approach tested on 101 object categories. In *CVPR workshop*, pages 178–178. IEEE, 2004. 2, 6

[17] Peng Gao, Shijie Geng, Renrui Zhang, Teli Ma, Rongyao Fang, Yongfeng Zhang, Hongsheng Li, and Yu Qiao. CLIP-Adapter: Better vision-language models with feature adapters. *IJCV*, pages 1–15, 2023. 1, 3, 4, 5, 7, 8

[18] Ziyu Guo, Renrui Zhang, Longtian Qiu, Xianzheng Ma, Xupeng Miao, Xuming He, and Bin Cui. CALIP: zero-shot enhancement of CLIP with parameter-free attention. In *AAAI*, pages 746–754, 2023. 1

[19] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, pages 770–778, 2016. 2, 3

[20] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross B. Girshick. Momentum contrast for unsupervised visual representation learning. In *CVPR*, pages 9726–9735, 2020. 3

[21] 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. 3, 4

[22] Patrick Helber, Benjamin Bischke, Andreas Dengel, and Damian Borth. EuroSAT: A novel dataset and deep learning benchmark for land use and land cover classification. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 12(7):2217–2226, 2019. 2, 6

[23] Dan Hendrycks, Steven Basart, Norman Mu, Saurav Kadavath, Frank Wang, Evan Dorundo, Rahul Desai, Tyler Zhu, Samyak Parajuli, Mike Guo, et al. The many faces of robustness: A critical analysis of out-of-distribution generalization. In *ICCV*, pages 8340–8349, 2021. 2, 7

[24] Dan Hendrycks, Kevin Zhao, Steven Basart, Jacob Steinhardt, and Dawn Song. Natural adversarial examples. In *CVPR*, pages 15262–15271, 2021. 2, 7

[25] Zejiang Hou, Fei Sun, Yen-Kuang Chen, Yuan Xie, and Sun-Yuan Kung. MILAN: Masked image pretraining on language assisted representation. *arXiv preprint arXiv:2208.06049*, 2022. 3, 4

[26] Tony Huang, Jack Chu, and Fangyun Wei. Unsupervised prompt learning for vision-language models. *arXiv preprint arXiv:2204.03649*, 2022. 3

[27] Zhao Jin, Munawar Hayat, Yuwei Yang, Yulan Guo, and Yinjie Lei. Context-aware alignment and mutual masking for 3d-language pre-training. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 10984–10994, 2023. 1

[28] Muhammad Uzair Khattak, Hanoona Rasheed, Muhammad Maaz, Salman Khan, and Fahad Shahbaz Khan. Maple:

Multi-modal prompt learning. In *CVPR*, pages 19113–19122, 2023. 7

[29] Muhammad Uzair Khattak, Hanoona Abdul Rasheed, Muhammad Maaz, Salman H. Khan, and Fahad Shahbaz Khan. Maple: Multi-modal prompt learning. In *CVPR*, pages 19113–19122. IEEE, 2023. 3

[30] Jonathan Krause, Michael Stark, Jia Deng, and Li Fei-Fei. 3D object representations for fine-grained categorization. In *ICCV workshops*, pages 554–561, 2013. 2, 6

[31] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *ICCV*, pages 10012–10022, 2021. 3

[32] Ze Liu, Han Hu, Yutong Lin, Zhuliang Yao, Zhenda Xie, Yixuan Wei, Jia Ning, Yue Cao, Zheng Zhang, Li Dong, et al. Swin transformer v2: Scaling up capacity and resolution. In *CVPR*, pages 12009–12019, 2022. 3

[33] Subhransu Maji, Esa Rahtu, Juho Kannala, Matthew Blaschko, and Andrea Vedaldi. Fine-grained visual classification of aircraft. *arXiv preprint arXiv:1306.5151*, 2013. 2, 6

[34] Norman Mu, Alexander Kirillov, David Wagner, and Saining Xie. SLIP: Self-supervision meets language-image pre-training. In *ECCV*, pages 529–544. Springer, 2022. 3

[35] Thao Nguyen, Gabriel Ilharco, Mitchell Wortsman, Sewoong Oh, and Ludwig Schmidt. Quality not quantity: On the interaction between dataset design and robustness of clip. *NeurIPS*, 35:21455–21469, 2022. 7

[36] Alex Nichol, Joshua Achiam, and John Schulman. On first-order meta-learning algorithms. *CoRR*, abs/1803.02999, 2018. 3

[37] Maria-Elena Nilsback and Andrew Zisserman. Automated flower classification over a large number of classes. In *2008 Sixth Indian conference on computer vision, graphics & image processing*, pages 722–729. IEEE, 2008. 2, 6

[38] R OpenAI. GPT-4 technical report. arxiv 2303.08774. *View in Article*, 2023. 1, 3

[39] Boris Oreshkin, Pau Rodríguez López, and Alexandre Lacoste. Tadam: Task dependent adaptive metric for improved few-shot learning. *NeurIPS*, 31, 2018. 3

[40] Omkar M. Parkhi, Andrea Vedaldi, Andrew Zisserman, and C. V. Jawahar. Cats and dogs. In *CVPR*, pages 3498–3505. IEEE Computer Society, 2012. 2, 6

[41] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. PyTorch: An imperative style, high-performance deep learning library. In *NeurIPS*. Curran Associates, Inc., 2019. 7

[42] Longtian Qiu, Renrui Zhang, Ziyu Guo, Ziyao Zeng, Yafeng Li, and Guangnan Zhang. VT-CLIP: Enhancing vision-language models with visual-guided texts. *arXiv preprint arXiv:2112.02399*, 2021. 7

[43] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *ICML*, pages 8748–8763, 2021. 1, 3, 4, 5, 7, 8

[44] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *JMLR*, 21(1):5485–5551, 2020. 3

[45] Aditya Ramesh, Mikhail Pavlov, Gabriel Goh, Scott Gray, Chelsea Voss, Alec Radford, Mark Chen, and Ilya Sutskever. Zero-shot text-to-image generation. In *ICML*, pages 8821–8831. PMLR, 2021. 3, 4

[46] Benjamin Recht, Rebecca Roelofs, Ludwig Schmidt, and Vaishaal Shankar. Do ImageNet classifiers generalize to ImageNet? In *ICML*, pages 5389–5400. PMLR, 2019. 2, 7

[47] James Requeima, Jonathan Gordon, John Bronskill, Sebastian Nowozin, and Richard E. Turner. Fast and flexible multi-task classification using conditional neural adaptive processes. In *NeurIPS 2019*. 3

[48] Pau Rodríguez, Issam Laradji, Alexandre Drouin, and Alexandre Lacoste. Embedding propagation: Smoother manifold for few-shot classification. In *ECCV*, pages 121–138. Springer, 2020.

[49] Jake Snell, Kevin Swersky, and Richard Zemel. Prototypical networks for few-shot learning. *NeurIPS*, 30, 2017. 3

[50] Khurram Soomro, Amir Roshan Zamir, and Mubarak Shah. UCF101: A dataset of 101 human actions classes from videos in the wild. *arXiv preprint arXiv:1212.0402*, 2012. 2, 6

[51] 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. 6

[52] Keyu Tian, Yi Jiang, Qishuai Diao, Chen Lin, Liwei Wang, and Zehuan Yuan. Designing BERT for convolutional networks: Sparse and hierarchical masked modeling. In *ICLR*, 2023. 3, 4

[53] Yonglong Tian, Dilip Krishnan, and Phillip Isola. Contrastive multiview coding. In *ECCV*, pages 776–794. Springer, 2020. 3

[54] Yonglong Tian, Chen Sun, Ben Poole, Dilip Krishnan, Cordelia Schmid, and Phillip Isola. What makes for good views for contrastive learning? *NeurIPS*, 33, 2020. 3

[55] Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Hervé Jégou. Training data-efficient image transformers & distillation through attention. In *ICML*, pages 10347–10357, 2021. 3

[56] Vishaal Udandarao, Ankush Gupta, and Samuel Albanie. SuS-X: Training-free name-only transfer of vision-language models. In *ICCV*, pages 2725–2736, 2023. 1

[57] Haohan Wang, Songwei Ge, Zachary Lipton, and Eric P Xing. Learning robust global representations by penalizing local predictive power. *NeurIPS*, 32, 2019. 2, 7

[58] Limin Wang, Bingkun Huang, Zhiyu Zhao, Zhan Tong, Yinan He, Yi Wang, Yali Wang, and Yu Qiao. VideoMAE v2: Scaling video masked autoencoders with dual masking. In *CVPR*, pages 14549–14560, 2023. 3

[59] Wenhui Wang, Hangbo Bao, Li Dong, Johan Bjorck, Zhiliang Peng, Qiang Liu, Kriti Aggarwal, Owais Khan Mohammed, Saksham Singhal, Subhojit Som, and Furu Wei. Image as a foreign language: BEiT pretraining for vision and vision-language tasks. In *CVPR*, pages 19175–19186. IEEE, 2023. 1, 3

[60] Jianxiong Xiao, James Hays, Krista A Ehinger, Aude Oliva, and Antonio Torralba. SUN database: Large-scale scene recognition from abbey to zoo. In *CVPR*, pages 3485–3492. IEEE, 2010. 2, 6

[61] Yuwei Yang, Munawar Hayat, Zhao Jin, Hongyuan Zhu, and Yinjie Lei. Zero-shot point cloud segmentation by semantic-visual aware synthesis. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 11586–11596, 2023. 1

[62] Jiahui Yu, Zirui Wang, Vijay Vasudevan, Legg Yeung, Mojtaba Seyedhosseini, and Yonghui Wu. CoCa: Contrastive captioners are image-text foundation models. *Trans. Mach. Learn. Res.*, 2022. 1, 3

[63] 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, 3, 4, 5, 6, 7, 8

[64] 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, 3, 4, 5, 6, 7, 8

[65] 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. 3

[66] Kaiyang Zhou, Jingkang Yang, Chen Change Loy, and Ziwei Liu. Conditional prompt learning for vision-language models. In *CVPR*, pages 16816–16825, 2022. 1, 3, 4, 7

[67] Kaiyang Zhou, Jingkang Yang, Chen Change Loy, and Ziwei Liu. Learning to prompt for vision-language models. *IJCV*, 130(9):2337–2348, 2022. 1, 3, 4, 7, 8