Vision-Language Model Fine-Tuning via Simple Parameter-Efficient Modification

Anonymous ACL submission

Abstract

Recent advances in fine-tuning Vision-Language Models (VLMs) have witnessed the success of prompt tuning and adapter tuning, while the classic model fine-tuning on inherent parameters seems to be overlooked. It is believed that fine-tuning the parameters of VLMs with few-shot samples corrupts the pre-trained knowledge since fine-tuning the CLIP model even degrades performance. In this paper, we revisit this viewpoint, and propose a new perspective: fine-tuning the specific parameters instead of all will uncover the power of classic model fine-tuning on VLMs. Through our meticulous study, we propose ClipFit, a simple yet effective method to fine-tune CLIP without introducing any overhead of extra parameters. We demonstrate that by only fine-tuning the specific bias terms and normalization layers, ClipFit can improve the performance of zero-shot CLIP by 7.27% average harmonic mean accuracy. Lastly, to understand how fine-tuning in CLIPFit affects the pre-trained models, we conducted extensive experimental analyses w.r.t. changes in internal parameters and representations. We found that low-level text bias layers and the first layer normalization layer change much more than other layers. The code will be released.

1 Introduction

Large pre-trained Visual-Language Models (VLMs) have been developed a lot in recent years. For example, CLIP (Radford et al., 2021) and ALIG (Jia et al., 2021) demonstrated remarkable performance for various tasks, e.g., image recognition in a zero-shot fashion. To further improve the performance on the specific downstream tasks, prompt tuning (Lester et al., 2021; Yao et al., 2023; Zhu et al., 2023; Zhou et al., 2022a) and adapter tuning (Gao et al., 2023; Zhang et al., 2021) methods have been proposed. As shown in Fig. 1, prompt tuning methods proposed to introduce a set of learnable prompt vectors as the input of the text encoder while adapter tuning approaches adopted an additional bottleneck layer to learn new features. During the fine-tuning procedure, both of these two strategies keep CLIP's parameters fixed. The performance of prompt tuning and adapter tuning methods are superior on various tasks (Zhou et al., 2022b; Gao et al., 2023), so research on fine-tuning the inherent parameters of VLMs has been barely touched.

For language models, fully fine-tuning with downstream data can achieve promising results (Zaken et al., 2021; Liu et al., 2022). Moreover, recent works in language model fine-tuning (e.g., BitFit (Zaken et al., 2021)) have demonstrated that, without introducing any external parameters, fine-tuning only the bias terms in a pre-trained model can perform competitively on downstream tasks compared with fine-tuning the entire model. For VLMs, however, it is believed that fine-tuning the parameters of VLMs corrupts the inherent pre-trained knowledge as fully fine-tuning degrades performance (Zhou et al., 2022b). In this paper, we revisit this viewpoint and ask if, without introducing any external parameters, fine-tuning the inherent parameters of VLMs can achieve competitive performance compared with prompt tuning.

We start with directly applying BitFit to fine-tuning the CLIP model. We explore two strategies: (i) applying BitFit to the text encoder alone, and (ii) applying BitFit to both the text and image encoder. We found that both two strategies can acquire task-specific knowledge but their performance to unseen class data can be poor (more discussed in Sec. 4.4), implying that directly fine-tuning the bias terms of a text or image encoder may harm the model's generalization ability. These findings motivate us to develop more effective and efficient fine-tuning techniques for VLMs.

In light of this, we propose CLIPFit, a simple yet effective method for efficiently fine-tuning VLMs.

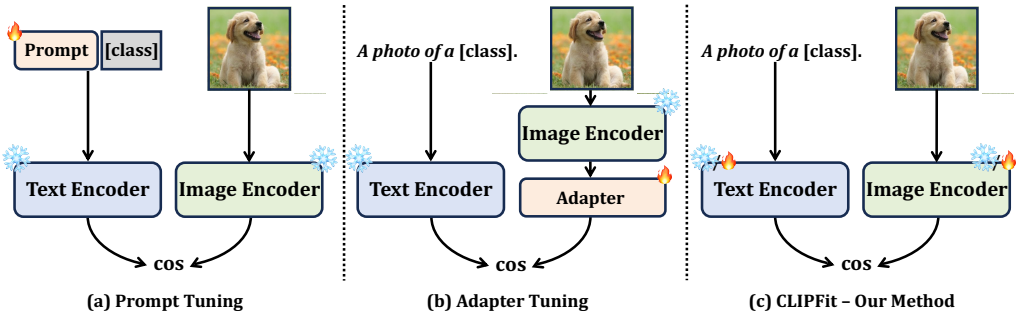

Figure 1: Comparison of (a) prompt tuning methods, (b) adapter tuning methods, and (c) our proposed CLIPFit method. Prompt tuning methods introduce a set of learnable external parameters as input to learn task-specific knowledge. Adapter tuning methods introduce extra learnable networks following the image encoder to learn task-specific features. Unlike these two methods, our CLIPFit does not introduce external parameters and fine-tunes only a small portion of the CLIP model.

CLIPFit is orthogonal to previous prompt tuning and adapter tuning methods, as shown in Fig. 1 (c). For the text encoder, instead of fine-tuning all the bias terms, CLIPFit proposes to tune only the bias terms of projection linear layers in feed-forward networks (FFNs). Fine-tuning only the bias terms of projection linear layers in FFNs will reduce the number of training parameters compared with fine-tuning all the bias terms. Moreover, empirically, we discovered that our bias term tuning strategy can generalize better than BitFit (Zaken et al., 2021), as shown in Sec. 4.4. For the image encoder, as mentioned before, it may harm the model's performance if directly applying BitFit. In the image encoder, layer normalization (LayerNorm) (Ba et al., 2016) aims to normalize the distributions of intermediate layers. Since the distributions of pre-training and downstream data might be divergent, pre-trained LayerNorm might lead to sub-optimal performance for downstream data inference. Therefore, CLIPFit proposes to further update only the parameters of the image encoder's LayerNorm. Updating LayerNorm can yield a better image encoder for downstream data. Lastly, previous studies (Yao et al., 2023) have shown that generic pre-trained knowledge is easily forgotten in the fine-tuning stage. Therefore, we explored two different regularization strategies for alleviating forgetting: (i) using the knowledge distillation (KD) loss (Hinton et al., 2015) to guide CLIPFit to learn from the zero-shot CLIP; (ii) using the mean squared error (MSE) loss in bias terms to penalize changes in text encoder. We empirically found that both two strategies can alleviate forgetting problems and the KD loss performs better, thus we used the KD loss as the final solution for CLIPFit.

Fine-tuning is an empirical and black-box process. So, understanding how fine-tuning affects the pre-trained models is important for uncovering the black-box fine-tuning process. Previous works (Zhou and Srikumar, 2022; De Vries et al., 2020; Merchant et al., 2020) explored this for language models fine-tuning. However, very little work explored the internal black-box fine-tuning process for VLMs. In this paper, we conducted an initial exploration to analyze VLM fine-tuning process of CLIPFit, focusing on changes in internal parameters and representations. We found that for bias terms in the FNN of the text encoder, as the number of layers increases, the change in bias decreases, which means that during the fine-tuning process, low-level features in the text encoder change more than high-level features. For LayerNorm in the image encoder, we found that the first layer (patch embedding) changes much more than other layers. Experimentally, we showed that more changed layers play a more important role in adapting downstream knowledge than less changed layers. Moreover, we explored how KD loss affects the fine-tuning process for alleviating forgetting. We found that KD loss will reduce the changes for the more-changed low-level bias terms and enhance changes in less-changed high-level layers, which implies that penalizing changes for low-level bias terms is important for avoiding overfitting. Lastly, we found that tuning LayerNorm will form a better image feature space compared with zero-shot CLIP.

We conducted extensive experiments on 11 datasets in 4 different settings to show the effectiveness of the proposed CLIPFit. Overall, our main contributions can be summarized as follows:

- We propose a CLIPFit method for efficiently fine-tuning the CLIP model to uncover the power of classic model fine-tuning on VLMs.

Unlike existing prompt tuning or adapter tuning methods, CLIPFit does not introduce any external parameters and only fine-tunes a small specific subset of CLIP's inherent parameters.

- To analyze how CLIPFit affects the pre-trained models, we conducted extensive analyses during the fine-tuning process, focusing on the changes in parameters and representations. These analyses help us better understand the black-box fine-tuning process.

- We conducted extensive experiments on 11 datasets. Results show that CLIPFit brings a 7.33% improvement in harmonic mean accuracy compared with zero-shot CLIP on the 16-shot base-to-new setting, demonstrating that CLIPFit is a promising alternative to prompt tuning and adapter tuning.

2 Related Works

Visual-Language Models (VLMs). With large-scale available web-crawled image-text pairs (Schuhmann et al., 2022), pre-training VLMs have been developed fast in recent years (Xu et al., 2021; Radford et al., 2021; Jia et al., 2021; Wang et al., 2022a) and achieved remarkable zero-shot performance in the downstream tasks, e.g., image classification. Despite the remarkable transfer ability, the potential of VLMs can be further stimulated by fine-tuning it with few-shot downstream data (Song et al., 2022; Zhang et al., 2021; Shen et al., 2021; Wang et al., 2022c,b; Chen et al., 2023).

Parameter-efficient Fine-tuning (PEFT) on VLMs. There are mainly two categories of VLM parameter-efficient fine-tuning methods: prompt tuning (Zhou et al., 2022b,a; Chen et al., 2022; Yao et al., 2023; Zhu et al., 2023; Zhang et al., 2023; Khattak et al., 2023) and adapter tuning (Gao et al., 2023; Zhang et al., 2021). Prompt tuning methods for VLMs introduced a few learnable parameters (prompts) as input, which were inspired by language prompt tuning (Lester et al., 2021). Adapter tuning methods set an additional bottleneck layer following the text or image encoder to learn better features by a residual way. Both prompt tuning and adapter tuning methods boost CLIP's performance, so research on fine-tuning the inherent parameters of CLIP seems to be overlooked. To explore classic model fine-tuning on VLMs, our CLIPFit proposes to fine-tune CLIP by modifying a small portion of the CLIP model's inherent parameters without introducing any external learnable parameters.

PEFT on Large Language Models. Fully fine-tuning language models (Radford et al., 2018; Devlin et al., 2018) can achieve promising results but is expensive. To efficiently fine-tune pre-trained language models, a lot of approaches have sought to fine-tune only a small number of parameters. For example, adapter methods (Bapna et al., 2019; Houlsby et al., 2019; Pfeiffer et al., 2020) and prompt tuning methods (Liu et al., 2023; Lester et al., 2021; Brown et al., 2020; Gao et al., 2020) introduce a set of learnable external parameters for adaptation to downstream tasks. Recently, BitFit (Zaken et al., 2021) demonstrated that, without introducing any new parameters, fine-tuning only the bias terms in pre-trained language models can perform competitively compared with fully fine-tuning. Moreover, to understand how fine-tuning affects pre-trained models, various works (Zhou and Srikumar, 2022; Mosbach et al., 2020; De Vries et al., 2020; Merchant et al., 2020) have explored this with LLM fine-tuning. However, very little work was attempted on the VLM side. In this paper, we attempt to bridge this gap by conducting an initial exploration to analyze the fine-tuning process in CLIPFit for VLMs, focusing on changes in internal parameters and representations.

3 Methodolgy

In this section, we introduce CLIPFit. We first briefly review CLIP and then illustrate CLIPFit.

3.1 Review of CLIP

We first briefly review CLIP (Radford et al., 2021). During pre-training, CLIP aims to align image features and text features in the joint embedding space to capture the relationship between images and texts. Let $D = \{(\boldsymbol{x}_i, \boldsymbol{t}_i)\}_{i=1}^{b}$ be the sampled batch, where $\boldsymbol{x}_i$ is the input image, $\boldsymbol{t}_i$ is the input text and b is the batch size. A CLIP model is comprised of two types of encoders: visual encoder $E_{\mathrm{I}}(\cdot, \boldsymbol{\theta}_{\mathrm{I}})$ and text encoder $E_{\mathrm{T}}(\cdot, \boldsymbol{\theta}_{\mathrm{T}})$. The visual encoder encodes image $\boldsymbol{x}_i$ into $\boldsymbol{f}_i$ and text $\boldsymbol{t}_i$ into $\boldsymbol{g}_i$, i.e.,

$$\boldsymbol{f}_i = E_{\mathrm{I}}(\boldsymbol{x}_i, \boldsymbol{\theta}_{\mathrm{I}}), \quad \boldsymbol{g}_j = E_{\mathrm{T}}(\boldsymbol{t}_i, \boldsymbol{\theta}_{\mathrm{T}}). \quad (1)$$

Then, a contrastive learning loss is applied to them for alignment.

After pre-training, CLIP can perform zero-shot image recognition by comparing the image features with class weights $\{\boldsymbol{w}_i\}_{i=1}^{K}$, where K is the number of classes. The class weight $\boldsymbol{w}_i$ is generated

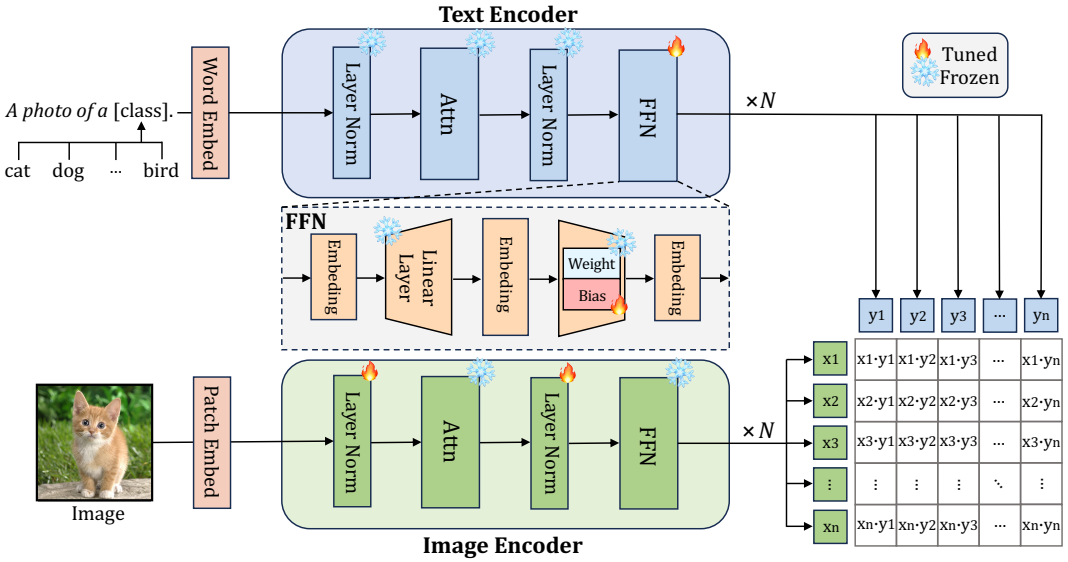

Figure 2: An overview of our CLIPFit. Unlike existing prompt tuning methods or adapter tuning methods, CLIPFit does not introduce any external parameters and fine-tunes specific inherent parameters of CLIP. For the text encoder, as shown in the upper part of the figure, CLIPFit fine-tunes only the bias terms of projection linear layers in feed-forward networks. For the image encoder, as shown in the lower part of the figure, CLIPFit updates LayerNorm.

by text encoder $E_T(\cdot, \boldsymbol{\theta}_T)$ which takes the class descriptions (prompts) as input. These prompts usually take the form "a photo of a [CLASS].", where the class token will be replaced by the i-th class name (e.g., cat) for weight $\boldsymbol{w}_i$. Formally, for an image feature $\boldsymbol{f}$, the probability that it belongs to class i is calculated by

$$p(y=i \mid \boldsymbol{x}) = \frac{\exp\left(\cos\left(\boldsymbol{w}_i, \boldsymbol{f}\right)/\tau\right)}{\sum_{j=1}^{K} \exp\left(\cos\left(\boldsymbol{w}_j, \boldsymbol{f}\right)/\tau\right)}, \quad (2)$$

where τ is a temperature parameter learned by CLIP during pre-training and $\cos(\cdot,\cdot)$ denotes the cosine similarity function.

3.2 CLIPFit

The overall pipeline of the proposed CLIPFit is shown in Fig. 2. Without introducing any external parameters, CLIPFit involves fine-tuning only the bias terms of projection linear layers in FNNs of the text encoder and updating LayerNorm (Ba et al., 2016) in the image encoder.

Text Encoder. For the text encoder, instead of fine-tuning all bias terms, CLIPFit fine-tunes only the bias terms of projection linear layers (i.e., second layers) in the FFNs of the text encoder. Fine-tuning only part of bias terms will reduce the number of training parameters compared with fine-tuning all bias terms. Moreover, Sec. 4.4 will empirically show that our bias tuning method can achieve better performance compared with fine-tuning all bias terms (Zaken et al., 2021).

Image Encoder. As mentioned in Sec. 1, directly applying BitFit (Zaken et al., 2021) to the image encoder may cause a negative impact on the model's performance. Instead of fine-tuning the bias terms of the image encoder, CLIPFit proposes to fine-tune LayerNorm. In LayerNorm, the two learnable parameters gain $\boldsymbol{g}$ and bias $\boldsymbol{b}$ are applied for affine transformation on normalized input vectors $\boldsymbol{x}$ for re-centering and re-scaling, which are expected to enhance the expressive power by re-shaping the distribution (Ba et al., 2016). Different data distributions should produce different gains and biases in LayerNorm for distribution re-shaping during the training process. So, if shifted gains and biases in LayerNorm are applied during inference, it may lead to a sub-optimal solution. Therefore, CLIPFit proposes to fine-tune LayerNorm in the image encoder.

Loss function. Previous works (Yao et al., 2023) have verified that during the fine-tuning stage, generic pre-trained knowledge is easily forgotten. Therefore, we explore two different strategies for alleviating such forgetting. The first one is to use the knowledge distillation (Hinton et al., 2015; Yao et al., 2023) loss to guide CLIPFit to learn from the original zero-shot CLIP. Let $\{\boldsymbol{w}_i^{\text{clip}}\}_{i=1}^{K}$ and $\{\boldsymbol{w}_i\}_{i=1}^{K}$be the text features from original CLIP and text features from CLIPFit. The training loss

and KD loss of CLIPFit are defined by

$$\mathcal{L} = \mathcal{L}_{\text{ce}} + \beta \mathcal{L}_{\text{kg}}, \tag{3}$$

$$\mathcal{L}_{\text{kg}} = \frac{1}{K} \sum_{i=1}^{K} \cos(\boldsymbol{w}_i^{\text{clip}}, \boldsymbol{w}_i), \tag{4}$$

where $\mathcal{L}_{\text{ce}}$ is the cross entropy loss for classification (Zhou et al., 2022b,a) and β is a hyperparameter.

The second strategy is using the MSE loss in bias terms to penalize changes in the text encoder. Let $\{\boldsymbol{b}_i^{\text{clip}}\}_{i=1}^{L}$ and $\{\boldsymbol{b}_i\}_{i=1}^{L}$ be the unfixed text bias terms from pre-trained CLIP and unfixed text bias terms from CLIPFit, where L is the number of unfixed bias layers. The MSE loss is defined as

$$\mathcal{L}_{\text{mse}} = \frac{1}{L} \sum_{i=1}^{L} ||\boldsymbol{b}_i^{\text{clip}} - \boldsymbol{b}_i||^2. \tag{5}$$

We found that both strategies can alleviate the forgetting problems and the KD loss performs better (as discussed in Sec. 4.3), thus we adopted the KD loss as the final solution for CLIPFit.

4 Experiments

In this section, we show and discuss the experimental results. To evaluate the effectiveness of our proposed method, we conducted extensive experiments and analyses on 11 datasets.

4.1 Experimental Setup

Datasets. Following CoOp, we conducted extensive experiments on 11 public classification benchmark datasets to evaluate CLIPFit. The datasets are ImageNet (Deng et al., 2009), Caltech101 (Fei-Fei et al., 2004), OxfordPets (Parkhi et al., 2012), StanfordCars (Krause et al., 2013), Flowers102 (Nilsback and Zisserman, 2008), Food101 (Bossard et al., 2014), FGVCAircraft (Maji et al., 2013), SUN397 (Xiao et al., 2010), DTD (Cimpoi et al., 2014), EuroSAT (Helber et al., 2019), and UCF101 (Soomro et al., 2012).

Implementation details. We implemented our method with PyTorch (Paszke et al., 2019). The experiments were based on the vision backbone with Vit-B/16 (Dosovitskiy et al., 2020). We followed CoOp to preprocess input images. We used a single text prompt for all experiments for a fair comparison. We used SGD optimizer with batch size set as 32, and set the learning rate as 0.002 (Zhou et al., 2022b). All results reported below are the average of three runs with different random seeds. The training epoch was set to 100 for all datasets except ImageNet and Food101. β was set to 8 for all datasets on the base-to-new and cross-dataset setting, and 2 for the distribution shift setting. For the few-shot setting, we set β to 2 for all datasets except SUN397 and DTD. More implementation details are provided in appendix B.

Dataset		CLIP	CoOp	CoCoOp	Adapter	KgCoOp	MaPLe	CLIPFit
Average	Base	69.34	82.69	80.47	82.23	80.73	82.28	**83.72**
	New	74.22	63.22	71.69	70.61	73.60	**75.14**	74.84
	HM	71.70	71.66	75.83	75.98	77.00	78.55	**79.03**
ImageNet	Base	72.43	76.47	75.98	76.13	75.83	**76.66**	76.2
	New	68.14	67.88	70.43	67.17	69.96	**70.54**	70.17
	HM	70.22	71.92	73.10	71.37	72.78	**73.47**	73.06
Caltech101	Base	96.84	98.00	97.96	97.40	97.72	97.74	**98.3**
	New	94.00	89.81	93.81	93.23	93.70	**94.36**	93.7
	HM	95.40	93.73	95.84	95.51	**96.03**	96.02	95.94
OxfordPets	Base	91.17	93.67	95.20	94.33	94.65	**95.43**	95.23
	New	97.26	95.29	97.69	97.10	97.76	**97.76**	97.13
	HM	94.12	94.47	96.43	95.69	96.18	**96.58**	96.17
Stanford Cars	Base	63.37	78.12	70.49	76.10	71.76	72.94	**78.80**
	New	74.89	60.40	68.87	71.20	**75.04**	74.00	73.87
	HM	68.65	68.13	72.01	72.30	73.36	73.47	**76.26**
Flowers102	Base	72.08	97.60	94.87	97.23	95.00	95.92	**96.83**
	New	**77.80**	59.67	71.75	69.27	74.73	72.46	73.53
	HM	74.83	74.06	81.71	80.90	83.65	82.56	**83.59**
Food101	Base	90.10	88.33	90.70	90.37	90.50	**90.71**	90.6
	New	91.22	82.26	91.29	90.83	91.70	**92.05**	91.33
	HM	90.66	85.19	90.99	90.6	91.09	**91.38**	90.96
FGVC Aircraft	Base	27.19	40.44	33.41	38.70	36.21	37.44	**42.47**
	New	**36.29**	22.30	23.71	32.27	33.55	35.61	33.47
	HM	31.09	28.75	27.74	35.19	34.83	36.50	**37.43**
SUN397	Base	69.36	80.60	79.74	81.57	80.29	80.82	**81.97**
	New	75.35	65.89	76.86	74.03	76.53	**78.70**	78.17
	HM	72.23	72.51	78.27	77.62	78.36	79.75	**80.02**
DTD	Base	53.24	79.44	77.01	79.53	77.55	80.36	**81.97**
	New	59.90	41.18	56.00	51.67	54.99	59.18	**63.5**
	HM	56.37	54.24	64.85	62.64	64.35	68.16	**71.56**
EuroSAT	Base	56.48	92.19	87.49	87.70	85.64	**94.07**	93.33
	New	64.05	54.74	60.04	58.83	64.34	**73.23**	71.07
	HM	60.03	68.69	71.21	70.42	73.48	**82.35**	80.69
UCF101	Base	70.53	84.69	82.33	85.47	82.89	83.00	**85.23**
	New	77.50	56.05	73.45	72.97	76.67	**78.66**	77.3
	HM	73.85	67.46	77.64	78.73	79.65	80.77	**81.07**

Table 1: Accuracy comparison on Base-to-new generalization of CLIPFit with previous methods. Adapter: CLIP-Adapter.

Comparisons. We compared our method against state-of-the-art methods: zero-shot CLIP, prompt tuning methods: CoOp, CoCoOp (Zhou et al., 2022a), ProGrad (Zhu et al., 2023), KgCoOp (Yao et al., 2023), MaPLe (Khattak et al., 2023) and adapter tuning methods: CLIP-adapter, Tip-adapter (Zhang et al., 2021). Detailed introductions to these methods can be found in appendix C.

4.2 Comparisons with State-of-the-arts

Results on base-to-new generalization setting Following Zhou et al. (2022b), we split each dataset into two disjoint groups: the base class dataset and the new class dataset. All compared methods and the proposed CLIPFit were trained on the base class dataset and evaluated on the new class dataset. We conducted $4/8/16$-shot experiments,

following Yao et al. (2023). We reported base and new class accuracies (Base and New) and their harmonic mean accuracy (HM). The 16-shot results are shown in Table 1, and $4/8$-shot results are provided in appendix G. As shown in Table 1, CLIPFit achieves 6 best HM accuracies among 11 datasets and the best average HM accuracy, which demonstrates that CLIPFit can not only learn well on seen base class data but also can generalize well to data from unseen new classes. A notable issue with previous methods like CoOp, CoCoOp, ProGrad, and KgCoOp is that they usually perform well only on either the base or new class. To alleviate this issue, MaPLe proposes a multi-modal prompt learning strategy for CLIP tuning which improves a lot over HM compared to previous methods. Compared to MaPLe, our CLIPFit achieves better HM accuracy and average performance on the base class, with slightly lower average performance on the new class. It is important to note that CLIPFit only needs to tune nearly 46K parameters while MaPLe needs to tune nearly 3.55M parameters for each task, which is 77 times more than CLIPFit, meaning that CLIPFit fine-tunes significantly fewer parameters and is much more efficient.

Results on few-shot learning setting. To verify whether the proposed CLIPFit can learn task-specific knowledge, we also compared CLIPFit with other existing methods on the few-shot learning setting. Following Zhou et al. (2022a), we used 1, 2, 4, 8, and 16-shot sets for training and reported accuracy performance. We report the results of the average accuracy of 11 datasets in Table. 2, and report all results on each dataset in appendix F. As shown in Table 2, compared with other methods, CLIPFit shows overall consistent improvements among all $1/2/4/8/16$-shot settings. This demonstrates that CLIPFit can successfully learn task-specific knowledge. It is worth noting that CLIPFit outperforms other methods by a large margin in $1/2/4$-shot settings, demonstrating CLIPFit's robust ability to learn with extremely few samples.

Results on the robustness to distribution shift setting. Following Zhang et al. (2021), we evaluated the robustness under distribution shift of CLIPFit and other methods by first training models on the 16-shot ImageNet dataset and then evaluating on ImageNet-V2 (Recht et al., 2019) and ImageNet-Sketch (Wang et al., 2019). The label sets of two evaluating datasets are subsets of the label set of ImageNet. Although the label sets are compatible, the distributions of these three datasets are different from each other. The results are shown in Table 3. As shown in Table 3, while TIP-adapter achieves the best performance on the ImageNet dataset, CLIPFit can achieve better average performance compared to existing methods, effectively underlining the robustness of CLIPFit.

Results on cross-dataset transfer setting is provided in appendix D.

Table 2: Comparison of CLIPFit and other methods on the few-shot learning setting. We report average accuracy on 11 datasets for the $1/2/4/8/16$-shot setting.

Method	shot				
	1	2	4	8	16
CoOP	68.09	70.13	73.59	76.45	79.01
CLIP-adapter	67.87	70.20	72.65	76.92	79.86
CoCoOp	66.95	67.63	71.98	72.92	75.02
ProGrad	68.2	71.78	74.21	77.93	79.2
KgCoOp	69.51	71.57	74.48	75.82	77.26
Tip-adapter	70.62	73.08	75.75	78.51	81.15
CLIPFit	**72.32**	**74.39**	**77.18**	**79.03**	**81.27**

Table 3: Comparison of our method against other methods on robustness to distribution shift.

Method	Souce	Target		Average
	ImageNet	-V2	-Sketch	
CLIP	66.73	60.83	46.15	57.90
CoOP	71.51	64.20	47.99	61.23
CLIP-adapter	71.60	63.67	46.52	60.60
Tip-adapter	**73.10**	64.82	46.73	61.55
CoCoOp	71.02	64.07	48.75	61.28
ProGrad	72.24	64.73	47.61	61.53
KgCoOp	71.20	64.10	**48.97**	61.42
CLIPFit	71.53	**64.83**	48.87	**61.74**

4.3 Fine-tuning Analysis

Analyzing parameter change. To understand the black-box fine-tuning process in CLIPFit, we first analyzed changes in the parameters of both the text encoder and image encoder. We computed the squared difference $||\boldsymbol{p}^{\text{pre}} - \boldsymbol{p}||^2$ for each layer, where $\boldsymbol{p}^{\text{pre}}$ is the pre-trained parameter vector and $\boldsymbol{p}$ is the fine-tuned parameter vector. We conduct experiments on the DTD dataset. The results are shown in Fig. 3. As observed Fig. 3 (a), for bias terms in the FNN of the text encoder, when the number of layers increases, the change in bias decreases, which implies that low-level features in the text encoder change more than high-level features during the fine-tuning process of CLIPFit. From Fig. 3 (b), we found that for LayerNorm in the image encoder, the first layer (i.e., patch embedding layer) changes much more compared with other layers for both bias and gain, showing that tuning patch embedding LayerNorm is crucial for shifted downstream tasks. Moreover, the gain of the last several LayerNorm layers has much changed and the most intermediate layers change much less. The

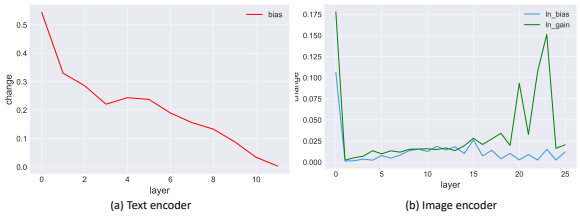

Figure 3: Visualization of changes in different layers.

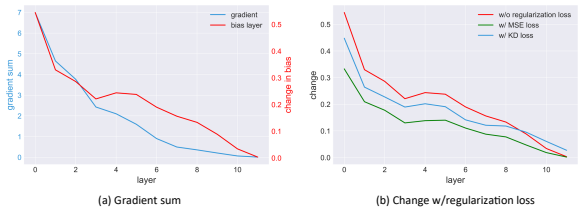

Figure 4: Left: visualization of squared gradient sum. Right: visualization of change w/ regularization loss.

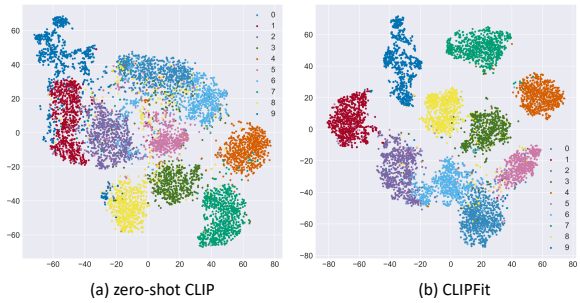

Figure 5: Visualization of learned image feature space from zero-shot CLIP and CLIPFit via t-SNE.

difference in change between different layers may be caused by gradient difference. We visualize the squared sum of gradient from each text bias layer in Fig. 4 (a). As observed, the curve of the gradient sum is very similar to changes in parameters.

To verify whether more changed layers are more important in fine-tuning, we conducted experiments by freezing less (or more) changed LayerNorm layers on the 4-shot setting. We found that when only updating the first LayerNorm layer and freezing other LayerNorm layers, the average accuracy is 76.22%. For comparison, the average accuracy is 74.93% when only updating the twelfth LayerNorm, and 75.06% when updating the last LayerNorm. Both are much less than the first layer and these two layers change much less than the first layer, as shown in Fig. 3 (b). Moreover, when updating the top 6 most changed LayerNorm layers, the average accuracy is 77.03%, which is only a 0.15% drop, while only tuning 23% parameters of CLIPFit. The phenomenon for text encoder is similar and can be found in appendix H. These results demonstrate that the more changed layers are more important for knowledge adapting.

Analyzing regularization loss. We then analyze the two regularization losses: KD loss and bias term MSE loss. We found that both two losses can avoid overfitting and boost performance during fine-tuning. For the 4-shot learning task, fine-tuning w/ KD loss leads to a 77.18% average accuracy, and fine-tuning w/ KD loss leads to a 76.23% average accuracy. Both two performances are better than fine-tuning w/o regularization loss (76.13% average accuracy) and KD loss performs better. We then analyze how these two losses affect changes in parameters during fine-tuning of CLIPFit. The results are shown in Fig. 4 (b). As observed, KD loss will reduce the changes for the more-changed low-level bias terms and enhance changes in less-changed high-level layers, which implies that penalizing changes for low-level bias terms is important in avoiding overfitting. Compared with KD loss, MSE loss directly applying to text bias terms reduces more changes in low-level layers.

Analysing image encoder representations. We used t-SNE (Van der Maaten and Hinton, 2008) to visualize the image representation space of zero-shot CLIP and CLIPFit to analyze image encoder representations. We visualize the data from EuroSAT dataset. The visualization results are presented in Fig. 5. As observed, in high-dimensional classification feature space, CLIPFit has a much clearer separation of different class image features compared with zero-shot CLIP, which demonstrates that CLIPFit can better detect the similarities among images. These results verify that updating LayerNorm in the image encoder during fine-tuning will lead to a more separated and better similarity-detected image feature space.

Updating LayerNorm can also benefit other methods. We show that updating LayerNorm can also benefit prompt tuning methods and adapter tuning methods. We re-implemented CoOp, KgCoOp, and CLIP-adapter with updating LayerNorm. The results are shown in Table 4. Table 4 shows that training with LayerNorm updating can boost the base class, new class, and harmonic mean performance for all three methods. For example, training KgCoOp with updating LayerNorm can bring 1.4%, 1.36%, and 1.38% improvements in the base class, new class, and Harmonic Mean (HM) accuracy, which demonstrates the effectiveness and wide validity of the proposed updating LayerNorm.

Table 4: Comparison of prompt tuning and adapter tuning methods w/ and w/o updating LayerNorm on the 16-shot base-to-new setting. +UL means training updating LayerNorm.

Method	Base	New	H
CoOp	82.63	67.99	74.60
+UL	82.96(+0.33)	69.09(+1.10)	75.39(+0.79)
KgCoOp	80.73	73.60	77.00
+UL	82.13(+1.40)	74.96(+1.36)	78.38(+1.38)
CLIP-adapter	82.23	70.61	75.98
+UL	83.63(+1.40)	71.87(+1.26)	77.31(+1.33)

Table 5: Comparison of different strategies of fine-tuning bias terms in CLIP.

Strategy	Base	New	H	# param.
(a) Text+Image bias	84.15	64.35	72.93	0.17M
(b) Text bias	83.33	64.43	72.67	67.6K
(c) FFNs bias (Text)	83.25	67.60	74.61	30.7K
(d) Projection bias (Text)	83.23	67.58	74.59	6.1K

More detailed analyses about other datasets and other aspects are provided in appendix H.

4.4 Ablation Study

Comparison of different strategies of fine-tuning bias terms. We give an in-depth exploration of how to apply BitFit to fine-tune the CLIP model. Original BitFit fine-tunes all bias terms in language models. We conduct 4 strategies for fine-tuning bias terms of CLIP: (a) fine-tuning all bias terms of the text and image encoder; (b) fine-tuning all bias terms of the text encoder; (c) fine-tuning bias terms of FFNs of the text encoder; (d) fine-tuning bias terms of projection linear layers in FFNs of the text encoder. We trained these four strategies on the 16-shot base-to-new setting with $\mathcal{L}_{\text{ce}}$ and reported average accuracy. The results are shown in Table 5. As shown in Table 5, both strategy (a) and strategy (b) can boost seen base class performance but will decrease significantly unseen new class performance, which implies that directly applying BitFit to CLIP may be harmful to model's generalization ability. Moreover, strategy (c) and strategy (d) can have similar performance among both the base and new class data, but strategy (d) fine-tunes only one-fifth of parameters compared with strategy (c), which speeds up training.

Effectiveness of proposed components. We validated the effects of updating LayerNorm and KD loss by ablating them. The results are shown in Table 6. Fine-tuning bias terms with KD loss brings 2.35%, 1.22%, and 0.09% improvements for $1/4/16$-shot setting, respectively. Fine-tuning bias terms in the text encoder and LayerNorm in the image encoder brings 1.89%, 1.41%, and 1.51% improvements for $1/4/16$-shot setting, respectively. Together, CLIPFit brings 3.46%, 2.46% and 1.74% improvements for $1/4/16$-shot setting, respectively. These results demonstrate the effectiveness of each CLIPFit components.

Configurations	Accuracy(%)		
	1-shot	4-shot	16-shot
Porjection Bias	68.86	74.72	79.53
w/ LayerNorm	70.75	76.13	81.04
w/ KD loss	71.21	75.94	79.62
w/ KD loss + LayerNorm	72.32	77.18	81.27

Table 6: Ablation study of CLIPFit on the few-shot setting with $1/4/16$-shot. Projection Bias: fine-tuning bias terms of projection layer in FNNs of the text encoder. LayerNorm: updating LayerNorm in the image encoder.

Method	Params	time	Base	New	HM
CoOp	2048	0.44ms	82.69	63.22	71.66
CoCoOp	35k	25.59ms	80.47	71.69	75.83
KgCoOp	2048	0.44ms	80.73	73.60	77.00
MaPLe	3.55M	2.1ds	82.28	75.14	78.55
CLIPFit	44k	0.96ms	83.72	74.84	79.03

Table 7: Comparison of training efficiency with other methods over 11 datasets.

Training efficiency. We compare the training efficiency of CLIPFit and other methods w.r.t. parameters and training time per image (Yao et al., 2023). The results are shown in Table 7. It is noticed that CoOp and KgCoOp have the lowest number of training parameters and time. However, the performance of these two methods is not satisfactory. MaPLe improves accuracy performance compared with other methods but also increases the required tuning parameters to 3.55M, which is very time-consuming. CLIPFit achieves the best harmonic mean accuracy with only 44k parameters, which is much less than MaPLe. Also, the training time of CLIPFit is slightly higher than CoOp and KgCoOp. Given the large improvement of CLIPFit, a slight increase in training time is acceptable.

5 Conclusion

In this paper, we presented CLIPFit for fine-tuning visual-language models. Unlike existing prompt tuning and adapter tuning methods, CLIPFit does not introduce any external parameters and fine-tunes CLIP by updating only bias terms of projection layers in FFNs of the text encoder and the image encoder's LayerNorm. To understand the effect of CLIPFit fine-tuning on the pre-trained model, we conducted various analyses focusing on changes in internal parameters and representations. We conducted extensive experiments and analysis to evaluate CLIPFit on 11 datasets, whose performances show the superiority of our method.

6 Limitations

In this paper, we presented CLIPFit for VLM fine-tuning and conducted an exploration of how CLIPFit affects the pre-trained CLIP model. Our analyses found some interesting phenomena after fine-tuning, i.e., low-level bias terms in the text encoder change much more than high-level bias terms and the change in the first LayerNorm layer is much bigger than other LayerNorm layers in the image encoders. Moreover, we found that this may be caused by the difference in the magnitude of the gradient. Nevertheless, our analysis does not reveal why the difference in the magnitude of the gradient happens during fine-tuning. A deeper analysis of gradient back-propagation during fine-tuning is needed to understand this for future work.

Furthermore, following previous works (Zhou et al., 2022a; Yao et al., 2023; Zhou et al., 2022b; Khattak et al., 2023), this paper focused on image classification for VLMs, so our study was constrained to classification tasks. Expanding CLIPFit for VLM fine-tuning to a broader range of tasks (e.g., image retrieval) could be the future work.

References

Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. 2016. Layer normalization. *arXiv preprint arXiv:1607.06450*.

Ankur Bapna, Naveen Arivazhagan, and Orhan Firat. 2019. Simple, scalable adaptation for neural machine translation. *arXiv preprint arXiv:1909.08478*.

Lukas Bossard, Matthieu Guillaumin, and Luc Van Gool. 2014. Food-101–mining discriminative components with random forests. In *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part VI 13*, pages 446–461. Springer.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

Guangyi Chen, Weiran Yao, Xiangchen Song, Xinyue Li, Yongming Rao, and Kun Zhang. 2022. Prompt learning with optimal transport for vision-language models. *arXiv preprint arXiv:2210.01253*.

Guanhua Chen, Lu Hou, Yun Chen, Wenliang Dai, Lifeng Shang, Xin Jiang, Qun Liu, Jia Pan, and Wenping Wang. 2023. mCLIP: Multilingual CLIP via cross-lingual transfer. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 13028–13043, Toronto, Canada. Association for Computational Linguistics.

Mircea Cimpoi, Subhransu Maji, Iasonas Kokkinos, Sammy Mohamed, and Andrea Vedaldi. 2014. Describing textures in the wild. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3606–3613.

Wietse De Vries, Andreas Van Cranenburgh, and Malvina Nissim. 2020. What's so special about bert's layers? a closer look at the nlp pipeline in monolingual and multilingual models. *arXiv preprint arXiv:2004.06499*.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. 2009. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. 2020. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*.

Li Fei-Fei, Rob Fergus, and Pietro Perona. 2004. Learning generative visual models from few training examples: An incremental bayesian approach tested on 101 object categories. In *2004 conference on computer vision and pattern recognition workshop*, pages 178–178. IEEE.

Peng Gao, Shijie Geng, Renrui Zhang, Teli Ma, Rongyao Fang, Yongfeng Zhang, Hongsheng Li, and Yu Qiao. 2023. Clip-adapter: Better vision-language models with feature adapters. *International Journal of Computer Vision*, pages 1–15.

Tianyu Gao, Adam Fisch, and Danqi Chen. 2020. Making pre-trained language models better few-shot learners. *arXiv preprint arXiv:2012.15723*.

Patrick Helber, Benjamin Bischke, Andreas Dengel, and Damian Borth. 2019. 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.

Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. 2015. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*.

Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin De Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. 2019.

Parameter-efficient transfer learning for nlp. In *International Conference on Machine Learning*, pages 2790–2799. PMLR.

Chao Jia, Yinfei Yang, Ye Xia, Yi-Ting Chen, Zarana Parekh, Hieu Pham, Quoc Le, Yun-Hsuan Sung, Zhen Li, and Tom Duerig. 2021. Scaling up visual and vision-language representation learning with noisy text supervision. In *International conference on machine learning*, pages 4904–4916. PMLR.

Muhammad Uzair Khattak, Hanoona Rasheed, Muhammad Maaz, Salman Khan, and Fahad Shahbaz Khan. 2023. Maple: Multi-modal prompt learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 19113–19122.

Jonathan Krause, Michael Stark, Jia Deng, and Li Fei-Fei. 2013. 3d object representations for fine-grained categorization. In *Proceedings of the IEEE international conference on computer vision workshops*, pages 554–561.

Brian Lester, Rami Al-Rfou, and Noah Constant. 2021. The power of scale for parameter-efficient prompt tuning. *arXiv preprint arXiv:2104.08691*.

Pengfei Liu, Weizhe Yuan, Jinlan Fu, Zhengbao Jiang, Hiroaki Hayashi, and Graham Neubig. 2023. Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing. *ACM Computing Surveys*, 55(9):1–35.

Xiao Liu, Kaixuan Ji, Yicheng Fu, Weng Tam, Zhengxiao Du, Zhilin Yang, and Jie Tang. 2022. P-tuning: Prompt tuning can be comparable to fine-tuning across scales and tasks. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 61–68.

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

Amil Merchant, Elahe Rahimtoroghi, Ellie Pavlick, and Ian Tenney. 2020. What happens to bert embeddings during fine-tuning? *arXiv preprint arXiv:2004.14448*.

Marius Mosbach, Anna Khokhlova, Michael A Hedderich, and Dietrich Klakow. 2020. On the interplay between fine-tuning and sentence-level probing for linguistic knowledge in pre-trained transformers. *arXiv preprint arXiv:2010.02616*.

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

Omkar M Parkhi, Andrea Vedaldi, Andrew Zisserman, and CV Jawahar. 2012. Cats and dogs. In *2012 IEEE conference on computer vision and pattern recognition*, pages 3498–3505. IEEE.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. 2019. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32.

Jonas Pfeiffer, Ivan Vulić, Iryna Gurevych, and Sebastian Ruder. 2020. Mad-x: An adapter-based framework for multi-task cross-lingual transfer. *arXiv preprint arXiv:2005.00052*.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. 2021. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PMLR.

Alec Radford, Karthik Narasimhan, Tim Salimans, Ilya Sutskever, et al. 2018. Improving language understanding by generative pre-training.

Benjamin Recht, Rebecca Roelofs, Ludwig Schmidt, and Vaishaal Shankar. 2019. Do imagenet classifiers generalize to imagenet? In *International conference on machine learning*, pages 5389–5400. PMLR.

Christoph Schuhmann, Romain Beaumont, Richard Vencu, Cade Gordon, Ross Wightman, Mehdi Cherti, Theo Coombes, Aarush Katta, Clayton Mullis, Mitchell Wortsman, et al. 2022. Laion-5b: An open large-scale dataset for training next generation image-text models. *Advances in Neural Information Processing Systems*, 35:25278–25294.

Sheng Shen, Liunian Harold Li, Hao Tan, Mohit Bansal, Anna Rohrbach, Kai-Wei Chang, Zhewei Yao, and Kurt Keutzer. 2021. How much can clip benefit vision-and-language tasks? *arXiv preprint arXiv:2107.06383*.

Haoyu Song, Li Dong, Wei-Nan Zhang, Ting Liu, and Furu Wei. 2022. Clip models are few-shot learners: Empirical studies on vqa and visual entailment. *arXiv preprint arXiv:2203.07190*.

Khurram Soomro, Amir Roshan Zamir, and Mubarak Shah. 2012. Ucf101: A dataset of 101 human actions classes from videos in the wild. *arXiv preprint arXiv:1212.0402*.

Laurens Van der Maaten and Geoffrey Hinton. 2008. Visualizing data using t-sne. *Journal of machine learning research*, 9(11).

Haohan Wang, Songwei Ge, Zachary Lipton, and Eric P Xing. 2019. Learning robust global representations by penalizing local predictive power. *Advances in Neural Information Processing Systems*, 32.

Zifeng Wang, Zhenbang Wu, Dinesh Agarwal, and Jimeng Sun. 2022a. Medclip: Contrastive learning from unpaired medical images and text. In *2022 Conference on Empirical Methods in Natural Language Processing, EMNLP 2022*.

Zifeng Wang, Zizhao Zhang, Sayna Ebrahimi, Ruoxi Sun, Han Zhang, Chen-Yu Lee, Xiaoqi Ren, Guolong Su, Vincent Perot, Jennifer Dy, et al. 2022b. Dualprompt: Complementary prompting for rehearsal-free continual learning. In *European Conference on Computer Vision*, pages 631–648. Springer.

Zifeng Wang, Zizhao Zhang, Chen-Yu Lee, Han Zhang, Ruoxi Sun, Xiaoqi Ren, Guolong Su, Vincent Perot, Jennifer Dy, and Tomas Pfister. 2022c. Learning to prompt for continual learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 139–149.

Jianxiong Xiao, James Hays, Krista A Ehinger, Aude Oliva, and Antonio Torralba. 2010. Sun database: Large-scale scene recognition from abbey to zoo. In *2010 IEEE computer society conference on computer vision and pattern recognition*, pages 3485–3492. IEEE.

Hu Xu, Gargi Ghosh, Po-Yao Huang, Dmytro Okhonko, Armen Aghajanyan, Florian Metze, Luke Zettlemoyer, and Christoph Feichtenhofer. 2021. Videoclip: Contrastive pre-training for zero-shot video-text understanding. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 6787–6800.

Hantao Yao, Rui Zhang, and Changsheng Xu. 2023. Visual-language prompt tuning with knowledge-guided context optimization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6757–6767.

Elad Ben Zaken, Shauli Ravfogel, and Yoav Goldberg. 2021. Bitfit: Simple parameter-efficient fine-tuning for transformer-based masked language-models. *arXiv preprint arXiv:2106.10199*.

Renrui Zhang, Rongyao Fang, Wei Zhang, Peng Gao, Kunchang Li, Jifeng Dai, Yu Qiao, and Hongsheng Li. 2021. Tip-adapter: Training-free clip-adapter for better vision-language modeling. *arXiv preprint arXiv:2111.03930*.

Renrui Zhang, Xiangfei Hu, Bohao Li, Siyuan Huang, Hanqiu Deng, Yu Qiao, Peng Gao, and Hongsheng Li. 2023. Prompt, generate, then cache: Cascade of foundation models makes strong few-shot learners. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15211–15222.

Kaiyang Zhou, Jingkang Yang, Chen Change Loy, and Ziwei Liu. 2022a. Conditional prompt learning for vision-language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16816–16825.

Kaiyang Zhou, Jingkang Yang, Chen Change Loy, and Ziwei Liu. 2022b. Learning to prompt for vision-language models. *International Journal of Computer Vision*, 130(9):2337–2348.

Yichu Zhou and Vivek Srikumar. 2022. A closer look at how fine-tuning changes bert. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1046–1061.

Beier Zhu, Yulei Niu, Yucheng Han, Yue Wu, and Hanwang Zhang. 2023. Prompt-aligned gradient for prompt tuning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 15659–15669.

A Dataset Statistics

Following CoOp (Zhou et al., 2022b), we conducted extensive experiments on 11 public classification benchmark datasets to evaluate the effectiveness of the proposed CLIPFit. The datasets are ImageNet (Deng et al., 2009), Caltech101 (Fei-Fei et al., 2004), OxfordPets (Parkhi et al., 2012), StanfordCars (Krause et al., 2013), Flowers102 (Nilsback and Zisserman, 2008), Food101 (Bossard et al., 2014), FGVCAircraft (Maji et al., 2013), SUN397 (Xiao et al., 2010), DTD (Cimpoi et al., 2014), EuroSAT (Helber et al., 2019), and UCF101 (Soomro et al., 2012). In distribution shift experiments, we also used ImageNet-V2 (Recht et al., 2019), and ImageNet-Sketch (Wang et al., 2019) as the target dataset. The statistics of these datasets can be found in Table 8.

B Implementation Details

We implemented our method with PyTorch (Paszke et al., 2019). All experiments were based on the vision backbone with Vit-B/16 (Dosovitskiy et al., 2020) of CLIP (Radford et al., 2021). We followed CoOp (Zhou et al., 2022b) to preprocess input images: we resized all images to 224×224 and used random cropping, resizing, and random horizontal flipping for data augmentation. Following Radford et al. (2021), we used a single hand-craft prompt as text input for all methods except prompt tuning methods for a fair comparison. The prompt for each dataset can be found in Table 8. We used SGD optimizer with batch size set as 32, and set the learning rate as 0.002 (Zhou et al., 2022b). All results reported below are the average of three runs with different random seeds. The training epoch was set to 100 for all datasets except ImageNet and Food101. The training epoch for ImageNet and Food101 datasets was set to 10. Smoothing parameter α was set to 0.99 for all experiments. β was set to 8 for all datasets on the base-to-new and cross-dataset setting, and 2 for the distribution shift setting. For the few-shot setting, we set β to 2 for all datasets except SUN397 and DTD. β was set to 8 for SUN397 and DTD datasets. All experiments were run on one single NVIDIA A100 GPU.

C Detailed Introduction to Baseline Methods.

We compared our method against state-of-the-art methods: zero-shot CLIP (Radford et al., 2021), prompt tuning methods: CoOp (Zhou et al., 2022b), CoCoOp (Zhou et al., 2022a), ProGrad (Zhu et al., 2023), KgCoOp (Yao et al., 2023) , MaPLe (Khattak et al., 2023) and adapter tuning methods: CLIP-ddapter (Gao et al., 2023), Tip-adapter (Zhang et al., 2021).

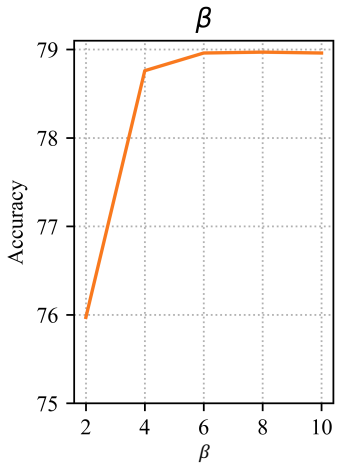

Figure 6: Performance changes of harmonic mean on 16 shot base-to-new setting by varying hyperparameter β.

Zero-shot CLIP (Radford et al., 2021) uses the hand-crafted template "a photo of a []" to generate the prompts and then applies these prompts to predict the class of given images.

CoOp (Zhou et al., 2022b) introduces learnable text prompts instead of hand-crafted prompts to adapt the CLIP model to downstream image recognition tasks.

CoCoOp (Zhou et al., 2022a) proposes to generate an input-conditional token for each image with a lightweight learnable neural network.

KgCoOp (Yao et al., 2023) proposes to use context knowledge distillation to learn from the original CLIP model to avoid overfitting and forgetting.

MaPLe (Khattak et al., 2023) proposes a multi-modal prompt learning strategy to introduce learnable text and image prompts.

CLIP-Adapter (Gao et al., 2023) sets an additional bottleneck layer following the text or image encoder to learn better features by a residual way.

Tip-Adapter(Zhang et al., 2021) does not need training but creates the weights by a key-value cache model constructed from the few-shot training set and then uses this cache model for inference.

Table 8: Statistics and prompts for each Dataset.

Dataset	Classes	Train	Val	Test	Hand-crafted prompt
ImageNet	1,000	1.28M	N/A	50,000	"a photo of a [CLASS]."
Caltech101	100	4,128	1,649	2,465	"a photo of a [CLASS]."
OxfordPets	37	2,944	736	3,669	"a photo of a [CLASS], a type of pet."
StanfordCars	196	6,509	1,635	8,041	"a photo of a [CLASS]."
Flowers	102	4,093	1,633	2,463	"a photo of a [CLASS], a type of flower."
Food101	101	50,500	20,200	30,300	"a photo of [CLASS], a type of food."
FGVCAircraft	100	3,334	3,333	3,333	"a photo of a [CLASS], a type of aircraft."
SUN397	397	15,880	3,970	19,850	"a photo of a [CLASS]."
DTD	47	2,820	1,128	1,692	"[CLASS] texture."
EuroSAT	10	13,500	5,400	8,100	"a centered satellite photo of [CLASS]."
UCF101	101	7,639	1,898	3,783	"a photo of a person doing [CLASS]."
ImageNetV2	1,000	N/A	N/A	10,000	"a photo of a [CLASS]."
ImageNet-Sketch	1,000	N/A	N/A	50,889	"a photo of a [CLASS]."

Table 9: Comparison of CLIPFit and other methods in the cross-dataset transfer setting. S.C.: StanfordCars dataset. F.A.: FGVCAircraft dataset.

	Source	Target										
	ImageNet	Caltech101	OxfordPets	S.C.	Flowers102	Food101	F.A.	SUN397	DTD	EuroSAT	UCF101	*Average*
CoOp	71.51	93.70	89.14	64.51	68.71	85.30	18.47	64.15	41.92	46.39	66.55	63.88
CoCoOp	71.02	**94.43**	90.14	65.32	**71.88**	86.06	22.94	67.36	45.73	45.37	68.21	65.74
ProGrad	**72.24**	91.52	89.64	62.39	67.87	85.40	20.61	62.47	39.42	43.46	64.29	62.71
KgCoOp	70.66	93.92	89.83	**65.41**	70.01	**86.36**	22.51	66.16	**46.35**	46.04	68.50	65.51
CLIPFit	71.10	93.77	**90.36**	64.56	71.43	85.76	**24.46**	**67.43**	45.20	**46.40**	**69.17**	**65.85**

D Results on cross-dataset transfer setting

Following Zhou et al. (2022b,a), We also evaluated the cross-dataset generalization ability of CLIPFit and other methods. Models were trained on the 16-shot Imagenet dataset and then tested on other datasets. The results are shown in Table 9. As shown in Table 9, the average performance of CLIPFit is also better than existing methods, which shows that CLIPFit has a good generalization ability.

E Parameter Analysis

In this section, we aim to discuss hyper-parameters β. α is the coefficient parameter to control the weight of knowledge distillation loss. Experiments were conducted on the 16-shot base-to-new setting, and we report harmonic mean accuracy in Fig. 6. As shown in Fig. 6, performances are not sensitive within certain ranges.

F More Few-shot Learning Results

Following Zhou et al. (2022a), we used 1, 2, 4, 8, and 16-shot sets for training and reported accuracy performance to test whether our proposed method can learn task-specific knowledge. The results are reported in Table 10. As shown in Table 10, CLIPFit can bring a consistent improvement in terms of average accuracy on all settings.

G More Results of base-to-new setting

In this section, we give more detailed results on the base-to-new setting. The detailed results for each dataset on 4-shot and 8-shot settings are shown in Table 11 and Table 12. Since MaPLe (Khattak et al., 2023) did not conduct experiments on $4/8$-shot setting, we do not report results from MaPLe. As shown in Table 11 and Table 12, the proposed CLIPFit brings consistent improvement compared with other methods.

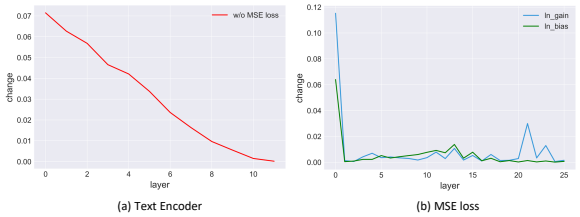

Figure 7: Visualization of changes in different layers on the EuroSAT dataset.

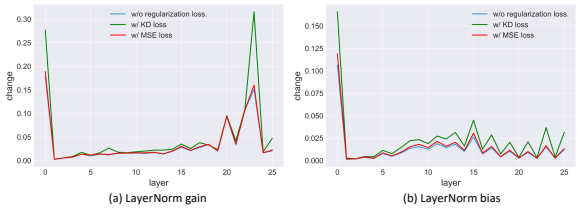

Figure 8: Visualization of LayerNorm changes w/ and w/o regularization loss in the DTD dataset.

H More Fine-tuning Analysis

Sec. 4.3 discussed the changes in unfixed parameters after fine-tuning the DTD dataset and the importance of more changed LayerNorm. In this subsection, we give more detailed analyses of other datasets and other aspects.

Importance of low-level bias terms in text encoder. Sec. 4.3 presented that after the fine-tuning of CLIPFit, for bias terms in the FNN of the text encoder, as the number of layers increases, the change in bias decreases. In this subsection, we conducted experiments to verify whether more changed layers in the text encoder are more important. Similar to Sec. 4.3, we freeze less (or more) changed LayerNorm bias layers in the text encoder on the 4-shot setting. When updating only the first bias layer and freezing other layers, the average accuracy is 74.69%. For comparison, the average accuracy is 73.33% when only updating the sixth bias layer and 70.62% when only updating the last bias layer. Both are much lower than updating the first layer. We also found that when only updating the top-3 bias layers (changed more) and freezing other bias term layers, the average accuracy is 76.13%. For comparison, when only updating the last 3 bias layers (changed less) and freezing other bias term layers, the average accuracy is 70.86%, which is much lower than updating the top 3 bias layers. These results demonstrate that the more changed parameters are crucial for fine-tuning.

Analyzing LayerNorm with regularization loss. Sec. 4.3 analyzed the difference of changes

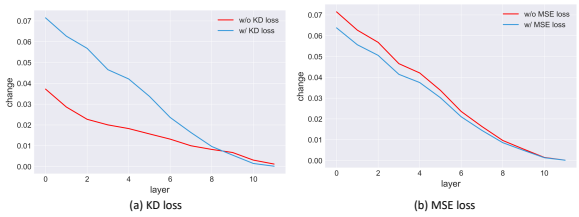

Figure 9: Visualization of bias changes w/ and w/o regularization loss in the EuroSAT dataset.

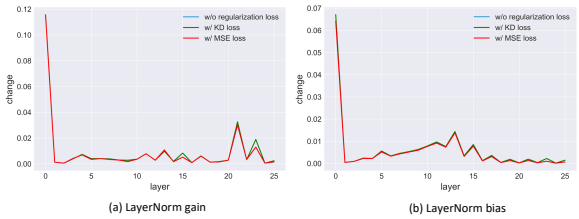

Figure 10: Visualization of LayerNorm changes w/ and w/o regularization loss in the EuroSAT dataset.

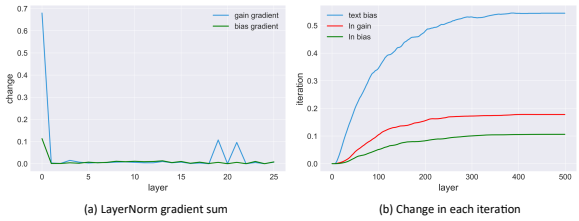

Figure 11: Left: visualization of squared gradient sum in LayerNorm layers. Right: change of the first text bias layer and first LayerNorm layer at each iteration.

in the text encoder bias terms between w/ and w.o regularization loss. In this subsection, we will analyze LayerNorm in the image encoder bias terms between w/ and w.o regularization loss. Noted that although the two regularization losses are applied to text features or text encoder, the image encoder or image features will also be affected since these two encoders are fine-tuned simultaneously. The results on the DTD dataset are shown in Fig. 8. When fine-tuning w/ KD loss, unlike in text encoder, changes in gain and bias increase compared with w/o KD loss. This phenomenon implies that image features will change more w/ KD loss compared with fine-tuning w/o KD loss. Moreover, we also found that the increases are almost in the more changed LayerNorm layers. When fine-tuning w/ MSE loss, changes in gain and bias are equal or slightly higher than fine-tuning w/o KD loss.

LayerNorm gradient. We visualize the squared sum of gradient from each LayerNorm layer in the image encoder in Fig. 11 (a). As observed, the magnitude of gradient in the first LayerNorm layer

Table 10: Comparison with existing methods in the few-shot learning setting. S.C.: StandfordCars dataset. F.A.: FGVCAircraft dataset.

shot	Method	ImageNet	Caltech101	OxfordPets	S.C.	Flowers	Food101	F.A.	Sun397	DTD	EuroSAT	UCF101	AVG
1	CoOp	65.77	92.37	**92.2**	67.1	82.2	82.07	26.73	64.7	49.0	54.8	72.0	68.09
	CoCoOp	**69.51**	93.8	91.17	67.92	71.98	86.1	13.2	68.19	48.51	55.71	70.35	66.95
	CLIP-adapter	67.93	93.3	89.03	67.1	72.03	85.9	27.6	67.1	45.2	61.7	69.67	67.87
	TIP-adapter	67.43	93.56	90.72	67.88	**86.63**	86.01	**29.58**	64.49	53.25	63.95	73.27	70.62
	ProGrad	64.33	90.96	89.01	67.11	83.81	82.75	27.97	64.54	52.79	55.1	71.91	68.2
	KgCoOp	69.03	**94.13**	91.97	67.03	74.63	**86.27**	26.9	68.43	52.5	60.83	72.93	69.51
	CLIPFit	69.37	93.67	91.63	**69.33**	82.83	86.17	27.73	**69.07**	**54.63**	**76.87**	**74.27**	**72.32**
2	CoOp	68.17	92.83	89.2	69.37	88.47	80.8	29.57	66.4	51.7	61.2	73.67	70.13
	CoCoOp	69.84	**94.92**	**92.14**	68.77	76.12	86.21	15.03	69.11	52.02	46.24	73.58	67.63
	CLIP-adapter	68.6	93.67	89.73	68.97	78.53	86.1	29.6	69.0	47.87	66.07	74.1	70.2
	TIP-adapter	68.6	94.22	91.1	71.39	**90.21**	86.26	**32.51**	66.74	56.32	70.38	76.11	73.08
	ProGrad	66.12	93.21	90.55	71.94	88.62	84.81	30.84	68.51	54.35	66.19	74.39	71.78
	KgCoOp	69.63	94.2	92.13	68.13	79.47	86.6	28.07	69.53	55.73	68.97	74.83	71.57
	CLIPFit	**69.93**	94.47	92.03	**72.7**	87.77	**86.63**	30.7	**70.87**	**57.7**	**78.83**	**76.7**	**74.39**
4	CoOp	69.38	94.44	91.3	72.73	91.14	82.58	33.18	70.13	58.57	68.62	77.41	73.59
	CoCoOp	**70.55**	94.98	93.01	69.1	82.56	**86.64**	30.87	70.5	54.79	63.83	74.99	71.98
	CLIP-adapter	69.56	94.0	90.87	71.13	86.77	86.47	31.1	71.3	53.83	66.8	77.3	72.65
	TIP-adapter	69.86	**95.06**	91.58	74.59	91.5	86.48	35.15	70.29	62.09	76.43	**80.24**	75.75
	ProGrad	70.21	94.93	**93.21**	71.75	89.98	85.77	32.93	71.17	57.72	70.84	77.82	74.21
	KgCoOp	70.19	94.65	93.2	71.98	90.69	86.59	32.47	71.79	58.31	71.06	78.4	74.48
	CLIPFit	70.4	95.0	93.07	**76.43**	**92.03**	86.73	**35.8**	**73.0**	**63.2**	**83.17**	80.13	**77.18**
8	CoOp	70.83	94.1	90.83	76.57	94.37	83.37	37.63	72.5	64.7	75.53	80.57	76.45
	CoCoOp	70.77	95.11	**93.44**	70.19	84.17	86.92	26.53	70.62	58.92	68.26	77.19	72.92
	CLIP-adapter	70.2	94.27	91.77	76.47	94.9	86.67	37.2	73.13	66.4	73.23	81.87	76.92
	TIP-adapter	**71.4**	95.2	92.09	78.34	**94.98**	86.74	**40.61**	73.6	**67.28**	81.11	82.24	78.51
	ProGrad	71.1	94.92	92.18	78.78	93.51	85.91	37.89	72.91	62.13	79.22	**88.64**	77.93
	KgCoOp	70.23	94.97	93.1	73.53	89.53	86.9	34.97	72.5	65.87	72.37	80.03	75.82
	CLIPFit	71.0	**95.43**	93.13	**79.57**	94.7	**87.0**	39.93	**74.27**	67.17	**84.87**	82.17	**79.03**
16	CoOp	71.51	95.5	91.8	78.89	96.1	85.17	40.93	74.5	68.63	83.6	82.43	79.01
	CoCoOp	71.02	95.19	93.25	71.68	87.64	87.19	31.29	72.05	63.78	73.82	78.34	75.02
	CLIP-adapter	71.6	94.57	92.03	80.9	**97.0**	86.83	42.67	75.3	71.17	81.87	**84.53**	79.86
	TIP-adapter	**73.1**	95.79	92.7	**83.09**	96.18	**87.24**	**45.59**	74.99	**72.05**	87.46	84.5	81.15
	ProGrad	72.68	95.8	92.13	81.46	94.87	87.01	40.39	75.0	65.92	84.38	81.59	79.2
	KgCoOp	71.2	95.03	93.23	74.87	92.9	87.03	36.27	73.4	69.37	74.93	81.43	77.26
	CLIPFit	71.53	**96.13**	**93.5**	82.43	96.37	87.37	45.47	**75.67**	71.57	**90.13**	83.83	**81.27**

is much bigger than other layers. So the difference in change may be caused by the difference in gradient.

Change in each iteration. We visualize the change in first-layer text bias terms, first-layer LayerNorm gain, and first-layer LayerNorm bias for each iteration in Fig. 11 (b). As observed, the change will increase smoothly and converge to some values.

Analyses on other datasets. We also conducted analyses on other datasets. The results for the EuroSAT dataset are shown in Fig. 7, Fig. 9, and Fig. 10. The phenomena in the EuroSAT dataset are very similar to the DTD dataset.

Table 11: Comparison with existing methods in the base-to-new generalization based on the **4-shot** settings. H: Harmonic mean.

Datasets	metric	CoOp	CLIP-adapter	CoCoOp	ProGrad	KgCoOp	CLIPFit
ImageNet	Base	73.6	74.23	75.46	74.24	74.87	75.03
	New	63.29	67.93	69.58	65.47	69.09	69.87
	H	68.06	70.94	72.4	69.58	71.86	72.36
Caltech101	Base	97.27	97.23	97.25	97.37	97.53	97.57
	New	93.01	94.17	94.9	93.92	94.43	94.23
	H	95.09	95.68	96.06	95.61	95.95	95.87
OxfordPets	Base	93.33	93.8	94.59	94.08	94.68	94.93
	New	95.69	97.0	96.75	97.63	97.58	96.97
	H	94.5	95.37	95.66	95.82	96.11	95.94
StandfordCars	Base	70.92	69.43	67.71	72.69	69.25	73.77
	New	69.38	73.0	75.37	69.88	74.98	73.77
	H	70.14	71.17	71.33	71.26	72.0	73.77
Flowers	Base	92.5	87.93	84.75	92.46	91.3	91.03
	New	70.12	71.9	73.85	72.69	75.34	74.47
	H	79.77	79.11	78.93	81.39	82.56	81.92
Food101	Base	86.79	90.2	89.79	88.91	90.3	90.2
	New	89.06	90.97	90.99	90.18	91.39	91.23
	H	87.91	90.58	90.39	89.54	90.84	90.71
FGVCAircraft	Base	33.21	32.43	32.07	33.73	34.21	34.53
	New	28.57	33.77	33.93	30.09	32.81	31.47
	H	30.72	33.09	32.97	31.81	33.5	32.93
Sun397	Base	76.49	77.7	77.57	77.72	78.87	79.5
	New	64.56	75.67	76.96	71.93	75.64	77.77
	H	70.02	76.67	77.26	74.71	77.22	78.63
DTD	Base	71.26	67.43	67.44	71.06	73.65	74.37
	New	50.93	55.43	56.0	52.58	57.21	64.1
	H	59.4	60.84	61.19	60.44	64.4	68.85
EuroSAT	Base	82.56	81.9	79.27	82.48	82.63	88.57
	New	53.04	59.67	65.44	56.43	59.98	76.7
	H	64.59	69.04	71.69	67.01	69.51	82.21
UCF101	Base	79.97	80.4	78.01	81.3	80.8	82.77
	New	65.98	76.17	73.07	76.02	75.77	76.43
	H	72.3	78.23	75.46	78.57	78.2	79.47
AVG	Base	78.43	77.52	76.72	79.18	78.92	80.21
	New	68.03	72.33	73.35	71.14	73.11	75.18
	H	72.44	74.84	74.85	74.62	75.9	77.61

Table 12: Comparison with existing methods in the base-to-new generalization based on the **8-shot** settings. H: Harmonic mean.

Datasets	metric	CoOp	CLIP-adapter	CoCoOp	ProGrad	KgCoOp	CLIPFit
ImageNet	Base	75.22	75.07	75.52	75.72	75.84	75.73
	New	65.91	67.6	70.28	66.76	69.33	70.07
	H	70.26	71.14	72.81	70.96	72.44	72.79
Caltech101	Base	97.81	97.3	97.76	98.0	97.68	97.83
	New	92.58	93.83	93.63	93.38	94.1	93.8
	H	95.12	95.53	95.65	95.63	95.86	95.77
OxfordPets	Base	94.19	94.33	95.5	94.47	94.81	94.83
	New	96.11	96.83	97.69	97.03	97.58	97.03
	H	95.14	95.56	96.58	95.73	96.18	95.92
StandfordCars	Base	73.2	72.13	69.7	75.08	69.66	76.63
	New	67.44	71.37	74.13	70.63	75.4	74.23
	H	70.2	71.75	71.85	72.79	72.42	75.41
Flowers	Base	96.17	94.27	92.24	93.8	87.72	94.17
	New	69.41	70.67	72.77	72.2	74.75	74.47
	H	80.63	80.78	81.36	81.59	80.72	83.17
Food101	Base	87.27	90.27	89.6	89.48	90.46	90.33
	New	86.96	90.7	90.79	89.9	91.63	91.5
	H	87.11	90.48	90.19	89.69	91.04	90.91
FGVCAircraft	Base	37.01	35.47	33.71	36.89	34.53	38.9
	New	38.45	33.03	32.15	31.67	34.95	32.43
	H	37.72	34.2	32.91	34.08	34.74	35.37
Sun397	Base	78.61	79.53	78.05	79.21	79.37	80.57
	New	66.25	74.9	76.29	70.77	76.85	77.77
	H	71.9	77.1	77.16	74.75	78.09	79.15
DTD	Base	76.97	74.43	73.03	74.42	69.72	77.87
	New	51.81	52.77	57.24	52.38	56.44	62.63
	H	61.93	61.75	64.18	61.48	62.38	69.42
EuroSAT	Base	83.27	80.23	78.68	82.27	81.07	90.3
	New	50.59	59.87	56.03	58.52	63.13	73.0
	H	62.94	68.57	65.45	68.39	70.98	80.73
UCF101	Base	82.85	82.83	80.4	82.61	81.16	84.4
	New	64.32	74.53	71.68	73.75	78.65	77.57
	H	72.42	78.46	75.79	77.93	79.89	80.84
AVG	Base	80.74	79.62	78.56	80.62	78.37	81.96
	New	68.39	71.46	72.06	71.02	73.75	74.95
	H	73.51	75.32	74.9	75.21	76.06	78.3