Skip Tuning: Pre-trained Vision-Language Models are Effective and Efficient Adapters Themselves

Shihan Wu[1] Ji Zhang[2*] Pengpeng Zeng[3] Lianli Gao[3] Jingkuan Song[4] Heng Tao Shen[4]
[1] University of Electronic Science and Technology of China (UESTC)
[2] Southwest Jiaotong University
[3] Shenzhen Institute for Advanced Study, UESTC
[4] Tongji University
{shihan.wu.koorye@outlook.com,jizhang.jim@gmail.com}

Abstract

Prompt tuning (PT) has long been recognized as an effective and efficient paradigm for transferring large pre-trained vision-language models (VLMs) to downstream tasks by learning a tiny set of context vectors. Nevertheless, in this work, we reveal that freezing the parameters of VLMs during learning the context vectors neither facilitates the transferability of pre-trained knowledge nor improves the memory and time efficiency significantly. Upon further investigation, we find that reducing both the length and width of the feature-gradient propagation flows of the full fine-tuning (FT) baseline is key to achieving effective and efficient knowledge transfer. Motivated by this, we propose Skip Tuning, a novel paradigm for adapting VLMs to downstream tasks. Unlike existing PT or adapter-based methods, Skip Tuning applies Layer-wise Skipping (LSkip) and Class-wise Skipping (CSkip) upon the FT baseline without introducing extra context vectors or adapter modules. Extensive experiments across a wide spectrum of benchmarks demonstrate the superior effectiveness and efficiency of our Skip Tuning over both PT and adapter-based methods. Code: https://github.com/Koorye/SkipTuning.

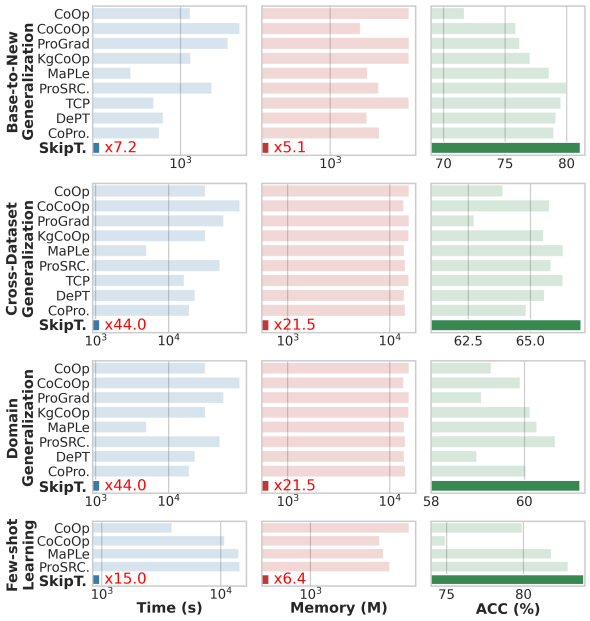

Figure 1. Comparison of our devised Skip Tuning with state-of-the-art prompt tuning methods in terms of training time (seconds), memory cost (M), and classification accuracy (%) across base-to-new generalization, cross-dataset generalization, domain generalization, and few-shot learning benchmarks. × indicates the performance improvement over the state-of-the-art. Comparison results with the adapter-based methods are reported in Table 5.

1. Introduction

There have recently been significant advancements in large pre-trained vision-language models (VLMs) [1, 24, 29]. One notable achievement is the CLIP model [24], which leverages an image-text matching (ITM) loss to align images with their corresponding textual descriptions in a common feature space. While VLMs have proven impressive capabilities in recognizing open-set visual concepts, their zero-shot generalization performance declines significantly when encountering category, distribution, or domain shifts between upstream training data and downstream tasks.

Prompt tuning (PT) has long been recognized as an effective and efficient paradigm for transferring large pre-trained vision-language models (VLMs) to downstream tasks. The core concept of PT is to learn a task-specific prompt (i.e., a small number of context vectors) for the target task, using a limited amount of training data, while keeping the pre-trained VLM parameters fixed. Although many PT ap-

*Corresponding author.

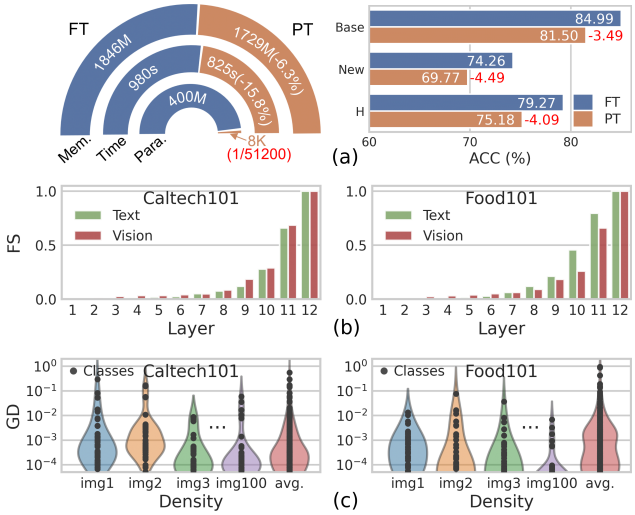

Figure 2. **Motivations**. **(a)** Comparison between the prompt tuning (PT) method CoOp [42] and the full fine-tuning (FT) baseline in terms of i) the number of learnable parameters, ii) memory usage, iii) time cost, and iv) base-to-new generalization performance. **(b)** Feature Sensitivity (**FS**) of CLIP's network layers, averaged over 100 randomly-sampled training images. **(c)** Gradient Dependence (**GD**) of class tokens for different training images.

proaches have reported improved performance and parameter efficiency over the full fine-tuning (FT) baseline, the discrepancy of implementation details among those PT approaches obscure the actual performance enhancement. For example, the FT performance can be significantly underestimated by training with coarsely tuned hyper-parameters. Quantitative evidence is presented in Figure 2 **(a)**, where we make comparisons between the PT method CoOp [42] and the FT baseline in terms of the number of learnable parameters, memory usage, time cost as well as base-to-new generalization performance (see §2.2). As observed, although PT significantly improves the parameter efficiency of FT (using **1/51200** parameters of FT), the improvements in memory and time efficiency are relatively insignificant (reducing only **6.3**% memory usage and **15.8**% time cost). Besides, compared to FT, the classification accuracy of PT decreases by **3.49**% and **4.49**% on base and new tasks, respectively. This suggests that pursuing higher parameter efficiency by freezing the overwhelming majority weights of VLMs during learning the context vectors neither facilitates the transferability of pre-trained knowledge nor improves memory and time efficiency considerably. Besides, in many real-world applications, memory and time efficiency often take precedence over parameter efficiency in terms of practical importance. We therefore raise the following question:

Can we optimize the memory and time efficiency of the FT baseline and adapt VLMs to downstream tasks in an effective and efficient manner?

To answer the above question, we scrutinize the Feature-Gradient Propagation Flows (FGPFs) in the vision encoder and text encoder of the CLIP model when performing FT on the base (or target) task. Interestingly, we observe that, for each training image, the majority of shallow network layers and class tokens contribute minimally to capturing task-specific knowledge for the base task (see §2.2). Motivated by this, we propose Skip Tuning, a novel paradigm for adapting VLMs to downstream tasks without introducing extra context vectors or adapter modules. Concretely, Skip Tuning incorporates two strategies, Layer-wise Skipping (LSkip) and Class-wise Skipping (CSkip), to simultaneously reduce the length and width of FGPFs in the FT baseline, thereby establishing effective and efficient knowledge transfer of VLMs, as shown in Figure 3.

Effectiveness and Efficiency. We conduct extensive experiments across a wide range of benchmarks to validate the effectiveness and efficiency of our Skip Tuning scheme. An overview of the achieved results is shown in Figure 1. As seen, our Skip Tuning demonstrates superiority over existing PT methods, e.g., on the few-shot learning benchmark, Skip Tuning achieves $\times$**15** time efficiency, $\times$**6.4** memory efficiency, while yielding a **1.04**% improvement in ACC compared to the state-of-the-art. Furthermore, we also show the advantages of Skip Tuning over existing adapter-based methods in Table 5, where Skip Tuning achieves $\times$**3.8** time efficiency, $\times$**3.9** memory efficiency, along with a **3.59**% H ACC enhancement over the strong rival LoRA [15].

Main Contributions. The main contributions are threefold.

- We reveal that reducing both the width and length of the feature-gradient propagation flows (FGPFs) of the full fine-tuning (FT) baseline is key to establishing effective and efficient knowledge transfer.
- We devise Skip Tuning, an effective and efficient method for transferring VLMs to downstream tasks without relying on extra context vectors or adapter modules.
- We evaluate our method on a wide spectrum of benchmarks, demonstrating the superiority of Skip Tuning over both prompt tuning and adapter-based approaches.

2. Methodology

In this section, we elaborate on our Skip Tuning approach. We start with an introduction of preliminaries.

2.1. Preliminaries

Contrastive Language-Image Pre-training (CLIP). Following the common practice of existing paradigms for adapting PVLMs, we adopt CLIP [24] as the testbed in this work. CLIP aims to learn an alignment between image and text features generated by an image encoder and a text encoder, respectively. By being exposed to 400 million image-text association pairs and employing a contrastive learning paradigm within a shared feature space, CLIP acquires a di-

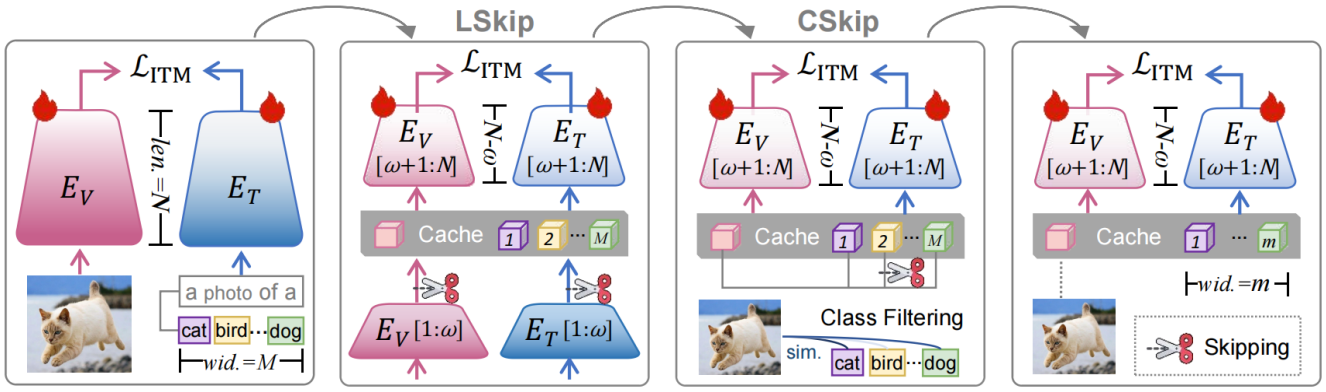

Figure 3. **Overview of our proposed Skip Tuning**. Skip Tuning performs Layer-wise Skipping (**LSkip**) and Class-wise Skipping (**CSkip**)) to enhance the memory and time efficiency of the FT baseline. Specifically, LSkip reduces the *length* of feature-gradient propagation flows (FGPFs) by caching intermediate features produced by the ω-th layers of CLIP's vision encoder E_V and text encoder E_T before FT begins. In contrast, CSkip reduces the *width* of FGPFs by filtering out unimportant class tokens in the text encoder E_T for every training image.

verse array of open-set visual concepts that can be readily applied to downstream tasks. For instance, zero-shot classification can be achieved by framing the classification task as an image-text matching problem. Initially, a prompt (e.g. "`a photo of a`") is crafted to extract text features of all intra-task classes by presenting the class-extended prompt ("`a photo of a [CLS]`") to the text encoder. Subsequently, the image encoder is utilized to derive the image feature of a given input example, and class prediction is performed by comparing cosine distances between the image feature and text features of classes.

Prompt Tuning with an Image-Text Matching Loss. PT aims to learn a task-specific prompt with few-labeled samples from the base task. Concretely, a prompt is formulated as κ context vectors: $\{\boldsymbol{v}_1, \boldsymbol{v}_2, ..., \boldsymbol{v}_\kappa\}$. During training, we produce the text feature of the j-th class by inputting the class-token-extended prompt $\boldsymbol{c}_j = \{\boldsymbol{v}_1, \boldsymbol{v}_2, ..., \boldsymbol{v}_\kappa, [\texttt{CLS}]\}$ to the text encoder $g(\cdot)$, where $[\texttt{CLS}]$ indicates the class token of the j-th class. Denote $\boldsymbol{f}_i \in \mathbb{R}^d$ as the image feature of a sample $\boldsymbol{x}_i$ extracted by the image encoder, the task-specific prompt can be updated via back-propagating the image-text matching (ITM) loss through the frozen CLIP model. The ITM loss can be expressed as

$$\mathcal{L}_{\text{ITM}} = -\sum_j \boldsymbol{y}_j \log \text{p}(\boldsymbol{c}_j|\boldsymbol{x}_i), \tag{1}$$

where $\boldsymbol{y}$ is the one-hot label, and

$$\text{p}(\boldsymbol{c}_j|\boldsymbol{x}_i) = \frac{e^{<g(\boldsymbol{c}_j), \boldsymbol{f}_i>/\tau}}{\sum_{t=1}^{M} e^{<g(\boldsymbol{c}_t), \boldsymbol{f}_i>/\tau}}, \tag{2}$$

$<\cdot>$ denotes cosine similarity, M is the number of classes, and τ is the temperature learned by CLIP.

2.2. Rethinking the FT Baseline

While many PT schemes have reported improved performance and efficiency over the full fine-tuning (FT) baseline, the discrepancy of the implementation details among those schemes obscures the actual performance enhancement.

Comparison of PT and FT. To comprehensively evaluate the progress established by PT, we perform a comparison between the representative PT method CoOp [42] with the full FT baseline in terms of i) the number of learnable parameters, ii) memory usage, iii) time cost, and iv) base-to-new generalization results. The experimental setting is shown in **Sup. Mat. (A)**. From the obtained results in Figure 2 **(a)**, we observe that although PT significantly reduces the number of learnable parameters compared with FT (using **1/51200** parameters of FT), the improvements in memory efficiency and time efficiency are relatively insignificant (reducing only **6.3**% memory usage and **15.8**% time cost). Notably, in comparison to FT, the classification accuracy of PT decreases by **3.49**% and **4.49**% on base and new tasks, respectively. This means that pursuing higher parameter efficiency by freezing the vast majority of VLM weights during PT neither enhances the transferability of pre-trained knowledge nor improves memory and time efficiency significantly. In other words, existing PT methods involve a trade-off between parameter efficiency and performance.

Cost Analysis of FT. The computational cost (i.e., memory and time cost) for fine-tuning VLMs is mainly caused by Feature-Gradient Propagation Flows (FGPFs)—the forward propagation flows of image/text features and the backward propagation flows of gradients. For example, on the CLIP model, each propagation flow is required to traverse all the N layers of the vision encoder E_V and the text encoder E_T. For simplicity, we assume that E_V and E_T have identical network structures (e.g. ViT-B/16). Denote the number of class tokens as M, and the costs of the FGPFs in the l-th layer of E_V and E_T as C_V and C_T respectively. Thus, the total cost $C_{\text{total}} = N \times (C_V + C_T \times M^1)$. In this work,

[1]For each training image, we need to obtain the text features of all M

we refer to the number of network layers and the number of class tokens *used for each training image* as the *length* and *width* of FGPFs, respectively. In particular, *length* $= N$, *width* $= M$ for existing PT methods, as shown in Figure 3. We therefore raise the following question:

> *Can we reduce both the length and width of FGPFs without compromising the FT performance?*

Towards Effective and Efficient Knowledge Transfer. To answer the above question, we design two metrics of Feature Sensitivity (**FS**) and Gradient Dependence (**GD**) to estimate the contributions of different network layers and class tokens for each training image $\boldsymbol{x}$, respectively.

More specifically, let $\boldsymbol{f}_l$ and $\boldsymbol{f}_l'$ be a vision (or a text) feature before and after fine-tuning the l-th layer of the CLIP's vision (or text) encoder, the **FS** of the l-th layer for the training image $\boldsymbol{x}$ can be expressed as

$$\mathbf{FS}_l(\boldsymbol{x}) = \psi(\boldsymbol{f}_l, \boldsymbol{f}_l'), \tag{3}$$

where $\psi(\cdot)$ is the Euclidean distance. Hence, the bigger the $\mathbf{FS}_l(\boldsymbol{x})$ value, the more important the l-th layer for the input image $\boldsymbol{x}$. Let $\nabla \boldsymbol{f}_l$ indicate the computed feature gradients passing through the last layer of the vision encoder for the training image $\boldsymbol{x}$ and $\nabla \boldsymbol{f}_l^{(c)}$ be the feature gradients after the c-th class token is removed from Eq. 2. The **GD** of the c-th class for $\boldsymbol{x}$ can be computed as

$$\mathbf{GD}_c(\boldsymbol{x}) = \psi(\nabla \boldsymbol{f}_l, \nabla \boldsymbol{f}_l^{(c)}). \tag{4}$$

Therefore, the bigger the $\mathbf{GD}_c(\boldsymbol{x})$ value, the more important the c-th class for the input image $\boldsymbol{x}$.

Figure 2 **(b)** illustrates the obtained **FS** values of different image/text encoder layers when adapting CLIP to the two datasets Caltech101 and Food101. As can be observed, the FS scores of the vast majority of shallow layers are close to 0, and only the last few layers contribute significantly to adapting CLIP to the two datasets. Moreover, Figure 2 **(c)** presents the frequency distributions of the **GD** values of class tokens for 100 training images randomly sampled from the two datasets. As shown, most class tokens contribute minimally to capturing task-specific knowledge from each training image. In a nutshell, the above observations reveal that we can achieve effective and efficient FT by filtering out unimportant network layers and class tokens in the text encoder for every training image.

2.3. Skip Tuning

Motivated by the previous section, we propose Skip Tuning, which performs Layer-wise Skipping (**LSkip**) and Class-wise Skipping (**CSkip**) to adapt VLMs to downstream tasks in an effective and efficient manner, as shown in Figure 3.

classes to calculate the image-text matching loss, as illustrated in Eq. (2).

Layer-wise Skipping (LSkip). LSkip aims to reduce the length of FGPFs without compromising the performance of the FT baseline. To this end, LSkip first saves the image and text features before the ω-th shallow layers of E_V and E_T in a cache, and then uses the saved intermediate features as input to update the parameters of the remaining $N - \omega$ deep layers. Denote $\boldsymbol{v}_i^{\omega}$ and $\boldsymbol{t}_j^{\omega}$ as the extracted image and text features of the image $\boldsymbol{x}_i$ and the class token `CLS` ($j = 1, ..., M$) in the ω-th layers of E_V and E_T, respectively, i.e.,

$$\boldsymbol{v}_i^{\omega} = E_V[1:\omega](\boldsymbol{x}_i), \boldsymbol{t}_j^{\omega} = E_T[1:\omega](\boldsymbol{c}_j), \tag{5}$$

where $\boldsymbol{c}_j$ is constructed using a hand-crafted prompt, e.g., $\boldsymbol{c}_j =$ `a photo of a [CLS]`.

After that, the $[1:\omega]$ layers of both E_V and E_T are discarded. During FT, we input $\boldsymbol{v}_i^{\omega}$ and $\boldsymbol{t}_j^{\omega}$ to the remaining $N - \omega$ deep layers of E_V and E_T, and obtain:

$$\tilde{\boldsymbol{v}}_i^{N} = E_V[\omega+1:N](\boldsymbol{v}_i^{\omega}), \tilde{\boldsymbol{t}}_j^{N} = E_T[\omega+1:N](\boldsymbol{t}_j^{\omega}), \tag{6}$$

which are used to calculate the loss $\mathcal{L}_{\text{ITM}}$ in Eq. 1. In this way, the image and text features, along with the calculated gradients, propagate through only $N - \omega$ layers, effectively reducing the length of FGPFs.

Class-wise Skipping (CSkip). Given a training image, the goal of CSkip is to filter out unimportant class tokens in the text encoder E_T during the construction of the image-text matching loss $\mathcal{L}_{\text{ITM}}$ (Eq. 1). Intuitively, the direct way is to select the top k closest class tokens to compute the loss for each image, based on the similarities between the image feature and the text features of the M inner-task classes. Nevertheless, we empirically find that this strategy leads to overfitting by selecting the same subset of classes for each image across different training epochs. Therefore, we propose an exponential image-conditioned class filtering strategy to overcome this limitation.

Concretely, before FT begins, we use the text encoder E_T to produce *M* class features with a hand-crafted prompt, e.g. "`a photo of a [CLS]`". Then, we sort the cosine similarities between the vision feature of a training image and the *M* class features in descending order. The probability of sampling the j-th class token for the current training image can be expressed as:

$$p_j = \begin{cases} 1 & o_j \leq r \times M, \\ \zeta(o_j - r \times M, \lambda) & o_j > r \times M, \end{cases} \tag{7}$$

where $\zeta(\iota, \lambda) = e^{-\lambda \iota}$, $\lambda > 0$ is the exponential decay coefficient, r is the sampling ratio, and o_j is the sorting index of the j-th class. In this way, for each image, we can maintain the top $r \times M$ closest class tokens while also sampling the remaining $(1 - r) \times M$ classes with a certain probability for constructing $\mathcal{L}_{\text{ITM}}$, improving the diversity of training data at different epochs. Denote m as the number of

class tokens sampled for the i-th training image, we have $m \ll M$. This means we can flexibly reduce the width of FGPFs in the text encoder for every training image. Surprisingly, our experimental results in the next section reveal that ignoring the majority of class tokens for each training image can enhance the generalization performance of the learned model. This improvement may stem from CSkip's capability to filter out redundant and distracting text features when performing image-text matching with the loss of $\mathcal{L}_{\text{ITM}}$.

3. Experiments

3.1. Experimental Setup

Datasets. We conduct experiments using 11 datasets from diverse sources. In particular, for base-to-new generalization, cross-dataset transfer and few-shot learning, we use 11 datasets including ImageNet [4], Caltech101 [5], OxfordPets [23], StanfordCars [19], Flowers102 [22], Food101 [2], FGVCAircraft [21], SUN397 [31], UCF101 [28], DTD [3] and EuroSAT [12]. For domain generalization, we use ImageNet [4] as the source dataset and its four variants as target datasets including ImageNetV2 [26], ImageNet-Sketch [30], ImageNet-A [14], and ImageNet-R [13].

Evaluation Metric. We report base-task accuracy (%, denoted as *Base*), new-task accuracy (%, denoted as *New*), and their harmonic mean (%, denoted as *H*) to compare the performance/effectiveness of different methods. We also report the time cost (seconds, denoted as *Time*) and memory usage (M, denoted as *Memory*) for efficiency evaluation.

Implementation details. Our implementation of Skip Tuning is based on the open-source Github repository of DePT [39][2]. In concrete terms, we leverage pre-trained ViT-B/16 as the backbone of the CLIP model. We employ the SGD optimizer to train the model with a learning rate of 2e-5 and a batch size of 4. By default, the number of training epochs is set to 20 for the base-to-new generalization, cross-dataset generalization, and domain generalization benchmarks, and 40 for the few-shot learning benchmark. The above hyper-parameters are fixed across all datasets. We adjust the LSkip hyper-parameter ω, CSkip hyper-parameters r, and λ in § 3.3. All experimental results are the average of 3 runs with different seeds. We conduct experiments using an NVIDIA V100 GPU. For more details and additional results, please refer to **Sup. Mat.**

3.2. Experimental Results

Base-to-New Generalization. The base-to-new generalization setting evaluates whether the models learned on a base task can generalize to new tasks with unseen classes. Following the comparison methods, for each dataset, we first construct a base task and a new task by equally dividing the dataset into two sets of classes, then we perform prompt tuning on the base task and test the learned model on both the base and new tasks. Table 1 presents the obtained results of Skip Tuning and other state-of-the-art prompt tuning methods on 11 datasets. As shown, Skip Tuning achieves the best base-to-new generalization performance with the lowest memory usage and time cost. Concretely, Skip Tuning achieves $\times$**7.2** time efficiency, and $\times$**5.1** memory efficiency, while maintaining a **1.14**% improvement in H ACC compared to the previous state-of-the-art method PromptSRC. This demonstrates the effectiveness and efficiency of our proposed Skip Tuning method. Moreover, we observe a tradeoff between base-task and new-task accuracies for most competitors. For instance, CoPrompt outperforms DePT on new tasks but lags behind DePT on base tasks. Notably, our Skip Tuning achieves the best performance on both base and new tasks simultaneously. This suggests that Skip Tuning effectively mitigates the overfitting issue when transferring VLMs to the base (or target) task.

Cross-Dataset Generalization. The cross-dataset generalization setting assesses whether models trained on a source dataset/distribution can generalize to unseen target datasets/distributions. We follow the common setup of those comparison methods to use ImageNet as the source dataset and the other 10 datasets as target datasets. Table 2 presents the obtained cross-dataset generalization results of our Skip Tuning and other state-of-the-art methods on the source and target datasets. As seen, compared to the previous state-of-the-art method PromptSRC, our Skip Tuning achieves $\times$**44** time efficiency, and $\times$**21.5** memory efficiency, while establishing **1.5**% and **1.19**% ACC improvements on the source and target distributions respectively. From the average results, Skip Tuning consistently shows superior effectiveness and efficiency over the nine competitors on the 10 target datasets, without compromising the results of the tuned model on the source dataset. This demonstrates the effectiveness of our Skip Tuning scheme for improving the robustness of the tuned model to distribution shifts.

Domain Generalization. The domain generalization setting assesses whether models trained on a source domain can generalize to unseen/target domains. In line with those comparison methods, we consider the ImageNet as the source domain and the other four ImageNet variants as target domains. As shown in Table 2, compared to the previous state-of-the-art method PromptSRC, Skip Tuning achieves $\times$**44** time efficiency, and $\times$**21.5** memory efficiency, while establishing **1.5**% and **0.55**% ACC improvements on the source and target domains, respectively. Besides, Skip Tuning consistently outperforms those competitors on the four target domains without compromising the performance of the tuned model on the source domain, which proves the effectiveness of our Skip Tuning scheme in improving the robustness of the tuned model to domain shifts.

[2] https://github.com/Koorye/DePT

Table 1. Base-to-new generalization results over 11 datasets. * indicates our reproduced results.

Datasets	Metric	CoOp (IJCV'22)	CoCoOp (CVPR'22)	ProGrad (ICCV'23)	KgCoOp (CVPR'23)	MaPLe (CVPR'23)	ProSRC. (ICCV'23)	TCP (CVPR'24)	DePT (CVPR'24)	CoPro.* (ICLR'24)	**SkipT.** (Ours)
ImgNet	Base	76.47	75.98	77.02	75.83	76.66	77.60	77.27	77.03	76.53	**77.73**
	New	67.88	70.43	66.66	69.96	70.54	70.73	69.87	70.13	**71.30**	70.40
	H	71.92	73.10	71.46	72.78	73.47	**74.01**	73.38	73.42	73.82	73.89
Caltech	Base	98.00	97.96	98.02	97.72	97.74	98.10	98.23	98.30	**98.60**	98.50
	New	89.31	93.81	93.89	94.39	94.36	94.03	94.67	94.60	95.17	**95.33**
	H	93.73	95.84	95.91	96.03	96.02	96.02	96.42	96.41	96.85	**96.89**
Pets	Base	93.67	95.20	95.07	94.65	95.43	95.33	94.67	94.33	94.73	**95.70**
	New	95.29	97.69	97.63	97.76	97.76	97.30	97.20	97.23	96.70	**97.87**
	H	94.47	96.43	96.33	96.18	96.58	96.30	95.92	95.76	95.71	**96.77**
Cars	Base	78.12	70.49	77.68	71.76	72.94	78.27	80.80	79.13	73.17	**82.93**
	New	60.40	73.59	68.63	75.04	74.00	74.97	74.13	**75.47**	70.63	72.50
	H	68.13	72.01	72.88	73.36	73.47	76.58	77.32	77.26	71.88	**77.37**
Flowers	Base	97.60	94.87	95.54	95.00	95.92	98.07	97.73	98.00	96.93	**98.57**
	New	59.67	71.75	71.87	74.73	72.46	**76.50**	75.57	76.37	75.50	75.80
	H	74.06	81.71	82.03	83.65	82.56	**85.95**	85.23	85.84	84.88	85.70
Food101	Base	88.33	90.70	90.37	90.50	**90.71**	90.67	90.57	90.50	90.37	90.67
	New	82.26	91.29	89.59	91.70	**92.05**	91.53	91.37	91.60	91.53	92.03
	H	85.19	90.99	89.98	91.09	**91.38**	91.10	90.97	91.05	90.95	91.34
Aircraft	Base	40.44	33.41	40.54	36.21	37.44	42.73	41.97	43.20	36.17	**45.37**
	New	22.30	23.71	27.57	33.55	35.61	**37.87**	34.43	34.83	34.47	37.13
	H	28.75	27.74	32.82	34.83	36.50	40.15	37.83	38.57	35.30	**40.84**
SUN397	Base	80.60	79.74	81.26	80.29	80.82	**82.67**	82.63	82.33	82.30	82.40
	New	65.89	76.86	74.17	76.53	78.70	78.47	78.20	77.80	**79.63**	79.03
	H	72.51	78.57	77.55	78.36	79.75	80.52	80.35	80.00	**80.94**	80.68
DTD	Base	79.44	77.01	77.35	77.55	80.36	83.37	82.77	82.20	83.00	**83.77**
	New	41.18	56.00	52.35	54.99	59.18	62.97	58.07	59.13	63.20	**67.23**
	H	54.24	64.85	62.45	64.35	68.16	71.15	68.25	68.78	71.76	**74.59**
EuroSAT	Base	92.19	87.49	90.11	85.64	**94.07**	92.90	91.63	89.03	93.77	92.47
	New	54.74	60.04	60.89	64.34	73.23	73.90	74.73	71.07	71.73	**83.00**
	H	68.69	71.21	72.67	73.48	82.35	82.32	82.32	79.04	81.28	**87.48**
UCF101	Base	84.69	82.33	84.33	82.89	83.00	87.10	87.13	85.80	86.20	**87.30**
	New	56.05	73.45	74.94	76.67	78.66	78.80	80.77	77.23	78.70	**82.47**
	H	67.46	77.64	79.35	79.65	80.77	82.74	83.83	81.29	82.28	**84.81**
Avg ACC	Base	82.69	80.47	82.48	80.73	82.28	84.26	84.13	83.62	82.89	**85.04**
	New	63.22	71.69	70.75	73.60	75.14	76.10	75.36	75.04	75.32	**77.53**
	H	71.66	75.83	76.16	77.00	78.55	79.97	79.51	79.10	78.93	**81.11**
Cost	Time (s)	1186	2851	2311	1191	413	1735	619	733	684	**239**
	Mem. (M)	3204	1556	3204	3188	1729	2041	3189	1714	2060	**404**

Table 2. Cross-dataset generalization results on 11 datasets. * indicates our reproduced results. The detailed results on the 10 target datasets (i.e, Caltech, Pets, Cars, ... , and UCF101) are reported in **Sup. Mat. (B)**.

Datasets	Metric	CoOp (IJCV'22)	CoCoOp (CVPR'22)	ProGrad (ICCV'23)	KgCoOp (CVPR'23)	MaPLe (CVPR'23)	ProSRC. (ICCV'23)	TCP (CVPR'24)	DePT (CVPR'24)	CoPro.* (ICLR'24)	**SkipT.** (Ours)
ImageNet	ACC	71.51	71.02	72.24	70.66	70.72	71.27	71.40	**72.77**	72.53	**72.77**
10 Datasets	Avg ACC	63.88	65.74	62.71	65.51	66.30	65.81	66.29	65.55	64.81	**67.00**
Cost	Time (s)	31632	93917	56223	31636	4942	50091	16174	22796	19161	**1139**
	Mem. (M)	15412	13622	15412	15254	13786	14107	15263	13783	14131	**656**

Table 3. Domain generalization results on ImageNet. * indicates our reproduced results. The detailed results on the 4 ImgNet variants (i.e., ImgNet-V2, ImgNet-S, ImgNet-A, and ImgNet-R) are presented in **Sup. Mat. (B)**.

Datasets	Metric	CoOp (IJCV'22)	CoCoOp (CVPR'22)	ProGrad (ICCV'23)	KgCoOp (CVPR'23)	MaPLe (CVPR'23)	ProSRC. (ICCV'23)	DePT (CVPR'24)	CoPro.* (ICLR'24)	**SkipT.** (Ours)
ImageNet	ACC	71.51	71.02	72.24	70.66	70.72	71.27	**72.77**	72.53	**72.77**
4 Imag.Variants	Avg ACC	59.28	59.90	59.07	60.11	60.26	60.65	58.97	60.02	**61.20**
Cost	Time (s)	31632	93917	56223	31636	4942	50091	22796	19161	**1139**
	Mem. (M)	15412	13622	15412	15254	13786	14107	13783	14131	**656**

3.3. Ablation Studies

In this section, we conduct ablative studies to further scrutinize our devised Skip Tuning method.

Effectiveness of the Designed Components. Our proposed Skip Tuning approach simultaneously performs Layer-wise Skipping (LSkip) and Class-wise Skipping (CSkip) to enhance the memory and time efficiency of the full fine-tuning (FT) baseline. In this experiment, we investigate the effectiveness of LSkip and CSkip by gradually adding them to the FT baseline. From the results in Table 4, we have the following observations. **i)** Both LSkip and CSkip contribute to performance improvement. **ii)** By combining all those components, Skip Tuning improves the effectiveness and efficiency of the baseline method remarkably. **iii)** The memory and time efficiency gains are more pronounced when CSkip is applied to the FT baseline, compared to the performance improvements it yields. **iv)** CSkip improves the memory and time efficiency of the baseline without com-

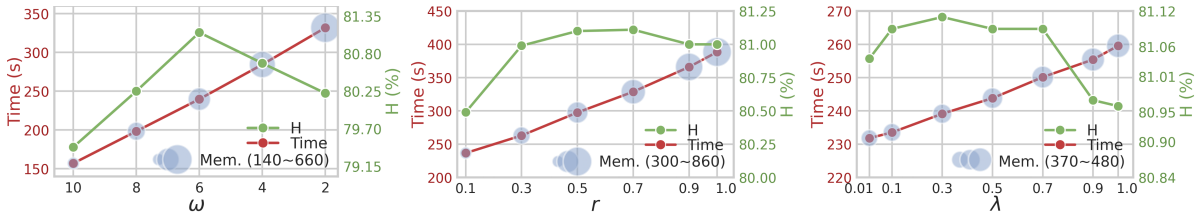

Figure 4. Ablation studies of the number of skipped layers ω in LSkip, and the sampling rate r, the decay coefficient λ in CSkip.

Table 4. Ablation study on the components of Skip Tuning.

LSP	CSP	Base	New	H	Time (s)	Mem. (M)
✗	✗	84.99	74.26	79.27	1002	1846
✓	✗	**85.08**	77.30	81.00	388	861
✗	✓	84.77	74.48	79.29	440	919
✓	✓	85.04	**77.53**	**81.11**	**239**	**404**

Table 5. Comparison of our Skip Tuning and adapter-based methods. † denotes the efficiency-optimized versions (details are illustrated in **Sup. Mat. (C)**).

Method	Base	New	H	Time (s)	Mem. (M)
FT	84.99	74.26	79.27	1002	1846
CLIP-adapter [10]	74.48	73.81	74.14	888	1784
LoRA [15]	80.53	74.73	77.52	910	1580
SkipT. (Ours)	**85.04**	**77.53**	**81.11**	**239**	**404**
CLIP-adapter†	76.75	73.56	75.12	137	175
SkipT. (Ours)†	**82.66**	**75.56**	**78.95**	**115**	**67**

promising the generalization performance of the learned model. One possible reason is that CSkip can filter out redundant and distracting text features when performing image-text matching with the loss of $\mathcal{L}_{\text{ITM}}$.

Impact of the LSkip Hyper-parameter ω. Our Skip Tuning approach drops out the $1 \sim \omega$ layers of the CLIP's vision and text encoders in the LSkip step. Here, we investigate the impact of ω on performance by setting ω to the values of $\{2, 4, 6, 8, 10\}$. The obtained average testing results on the 11 datasets are reported in Figure 4 (*Left*). As shown, the obtained H ACCs gradually increase as the ω value decreases from 10 to 6, after which the performance gradually decreases. But, we also see that as the value of ω becomes smaller, both memory and time costs increase. Hence, we set $\omega = 6$ for LSKip in this work.

Impact of the CSkip Hyper-parameter r. In the CSkip step of Skip Tuning, we devise an image-conditioned class sampling strategy to filter out irrelevant class tokens for each training image. The larger the sampling ratio r value, the more class tokens will be used to construct the training loss $\mathcal{L}_{\text{ITM}}$ for each image. It is necessary to scrutinize the impact of r on performance. To this end, we respectively set r to $\{0.1, 0.3, 0.5, 0.7, 0.9, 1.0\}$, and report the average testing results on the 11 datasets in Figure 4 (*Mid.*). From the obtained H ACCs, our method is in general not sensitive to the change of r within a certain range (from 0.4 to 1.0). We also see a gradual decrease in H ACC when $r > 0.7$, suggesting that for each training image, not all class tokens are beneficial for capturing task-specific knowledge. Besides, we can observe that as the value of r becomes larger, both memory and time costs increase. Therefore, we set $r = 0.5$ for CSKip in this work.

Impact of the CSkip Hyper-parameter λ. Our Skip Tuning introduces an exponential decay coefficient λ for the devised image-conditioned class sampling strategy in the CSkip step. We study the impact of λ by setting λ to the values of $\{0.01, 0.1, 0.3, 0.5, 0.7, 0.9, 1.0\}$, and report the average testing results over the 11 datasets in Figure 4 (*Right*). As can be observed from the obtained H ACCs, our method is in general not sensitive to λ when it takes the values from 0.1 to 0.7. Particularly, the H ACCs gradually decrease as λ increases from 0.7 to 1.0. Also, we can see that as the value of λ becomes larger, both memory and time costs increase. Thus, we set $\lambda = 0.3$ for CSkip in this work.

3.4. Additional Results

Comparison with Adapter-based Methods. Previous experimental results demonstrate the superior effectiveness and efficiency of our Skip Tuning method over existing prompt tuning methods. To further demonstrate the advantages of Skip Tuning, we also compare it with the representative adapter-based methods LoRA [15] and CLIP-adapter [10]. The base-to-new generalization results averaged on 11 datasets are reported in Table 5, where CLIP-adapter† refers to our re-implementation of CLIP-adapter, i.e., the CLIP's last-layer features are cached once before training and subsequently used to update the weights of the adapter module. As observed, our Skip Tuning offers substantial memory and time efficiency advantages over the two strong competitors, while also delivering better classification performance on both base and new tasks.

Few-shot Learning. In the previous experiments, we follow the comparison approaches to evaluate the performance of different methods on M-way 16-shot tasks—16 training examples are sampled for each of the M inner-task classes. It is interesting to further scrutinize the effectiveness and efficiency of our Skip Tuning method under different shots. To this end, Figure 5 reports the few-shot learning performance of Skip Tuning and other representative competitors. As shown, our Skip Tuning demonstrates superior performance compared to other methods, with significant efficiency advantages in memory usage and time cost across

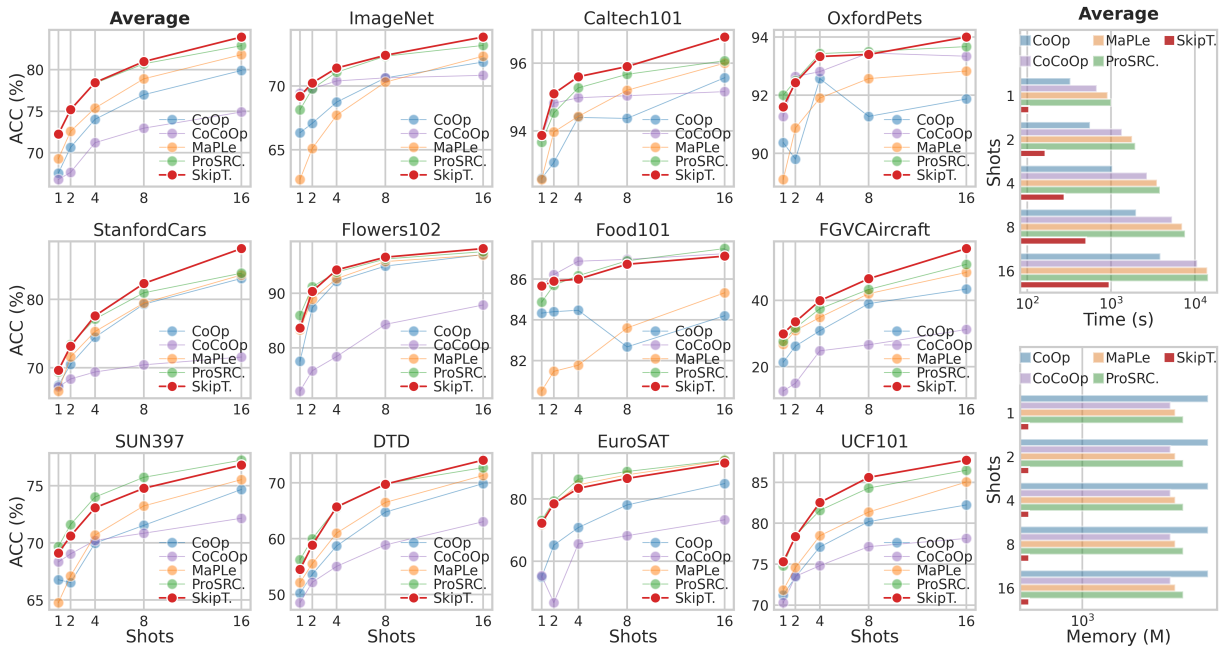

Figure 5. Few-shot learning results on 11 datasets.

1-shot to 16-shot settings. Additionally, while PromptSRC and MaPLe enhance CoOp's performance, both methods incur increased time cost. In contrast, Skip Tuning achieves both effectiveness and efficiency without compromise, further underscoring the superiority of our approach.

4. Related Work

Pre-trained Vision-Language Models. Pre-trained vision-language models (VLMs) have garnered great attention recently. Notable models such as CLIP[24], ALIGN[16], LiT[36] and FILIP[34] have demonstrated exceptional performance across various vision-language tasks. These models are pre-trained on extensive datasets comprising image-text pairs sourced from the internet. For instance, CLIP[24] and ALIGN[16] respectively utilize over 400 million and 1 billion pairs, for training. The large scale of training data enables these models to excel in open-vocabulary image-text retrieval and zero-shot classification. Additionally, they have been successfully applied to downstream tasks including image classification[10, 40], object detection[6, 8, 11, 35, 43], and semantic segmentation[20, 25]. In this work, we focus on the effective adaptation of vision-language models to downstream visual recognition tasks.

Prompt Tuning. Prompt tuning (a.k.a. context optimization [42]) has emerged as a parameter-efficient learning paradigm to adapt powerful VLMs to downstream tasks [7, 9, 37, 38]. As a representative method, CoOp [42] enables task adaptation by optimizing a set of prompt vectors within CLIP's language branch. While effective, CoOp often suffers from limited generalization on new tasks due to overfitting to the base task. A series of schemes are devised to tackle this problem, e.g. CoCoOp[41], KgCoOp[32], ProGrad[44], TCP [33], and DePT[39]. By adding trainable prompts into both the image and text branches of CLIP, MaPLe[17], PromptSRC[18], and CoPrompt[27] demonstrate remarkable performance on both base and new tasks. Despite the advantages, we reveal that freezing the parameters of VLMs during learning the context vectors neither facilitates the generalization of pre-trained knowledge nor significantly improves memory and time efficiency.

5. Conclusion

In this work, we first reveal that freezing the parameters of VLMs during prompt tuning neither facilitates the transferability of pre-trained knowledge nor improves memory and time efficiency considerably. To circumvent this limitation, we propose Skip Tuning, an effective and efficient method for transferring VLMs to downstream tasks without relying on extra context vectors or adapter modules. Extensive experiments across a broad range of benchmarks demonstrate the superiority of our Skip Tuning method over both prompt tuning and adapter-based approaches.

Acknowledgements. This study is supported by grants from the National Natural Science Foundation of China (Grant No. 62425208, No. U22A2097, No. 62122018, No. 62020106008, No. U23A20315, No. 82441006, No. 62402094), the Postdoctoral Fellowship Program of CPSF (Grant No. GZB20240625), the Science and Technology Innovation Committee of Shenzhen Municipality Foundation (Grant No.JCYJ20240813114208012).

References

[1] Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023. 1

[2] Lukas Bossard, Matthieu Guillaumin, and Luc Van Gool. 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, 2014. 5

[3] Mircea Cimpoi, Subhransu Maji, Iasonas Kokkinos, Sammy Mohamed, and Andrea Vedaldi. Describing textures in the wild. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3606–3613, 2014. 5

[4] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009. 5

[5] 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. *Computer vision and Image understanding*, 106(1):59–70, 2007. 5

[6] Chengjian Feng, Yujie Zhong, Zequn Jie, Xiangxiang Chu, Haibing Ren, Xiaolin Wei, Weidi Xie, and Lin Ma. Promptdet: Towards open-vocabulary detection using uncurated images. In *European Conference on Computer Vision*, pages 701–717. Springer, 2022. 8

[7] Yuqian Fu, Yanwei Fu, Jingjing Chen, and Yu-Gang Jiang. Generalized meta-fdmixup: Cross-domain few-shot learning guided by labeled target data. *IEEE Transactions on Image Processing*, 31:7078–7090, 2022. 8

[8] Yuqian Fu, Yu Xie, Yanwei Fu, and Yu-Gang Jiang. Styleadv: Meta style adversarial training for cross-domain few-shot learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 24575–24584, 2023. 8

[9] Yuqian Fu, Yu Wang, Yixuan Pan, Lian Huai, Xingyu Qiu, Zeyu Shangguan, Tong Liu, Yanwei Fu, Luc Van Gool, and Xingqun Jiang. Cross-domain few-shot object detection via enhanced open-set object detector. In *ECCV*, pages 247–264. Springer, 2025. 8

[10] 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. *International Journal of Computer Vision*, 132(2): 581–595, 2024. 7, 8

[11] Xiuye Gu, Tsung-Yi Lin, Weicheng Kuo, and Yin Cui. Open-vocabulary object detection via vision and language knowledge distillation. *arXiv preprint arXiv:2104.13921*, 2021. 8

[12] 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. 5

[13] 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 *Proceedings of the IEEE/CVF international conference on computer vision*, pages 8340–8349, 2021. 5

[14] Dan Hendrycks, Kevin Zhao, Steven Basart, Jacob Steinhardt, and Dawn Song. Natural adversarial examples. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 15262–15271, 2021. 5

[15] Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*, 2021. 2, 7

[16] Chao Jia, Yinfei Yang, Ye Xia, Yi-Ting Chen, Zarana Parekh, Hieu Pham, Quoc Le, Yun-Hsuan Sung, Zhen Li, and Tom Duerig. Scaling up visual and vision-language representation learning with noisy text supervision. In *International conference on machine learning*, pages 4904–4916. PMLR, 2021. 8

[17] Muhammad Uzair Khattak, Hanoona Rasheed, Muhammad Maaz, Salman Khan, and Fahad Shahbaz Khan. Maple: Multi-modal prompt learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 19113–19122, 2023. 8

[18] Muhammad Uzair Khattak, Syed Talal Wasim, Muzammal Naseer, Salman Khan, Ming-Hsuan Yang, and Fahad Shahbaz Khan. Self-regulating prompts: Foundational model adaptation without forgetting. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 15190–15200, 2023. 8

[19] Jonathan Krause, Michael Stark, Jia Deng, and Li Fei-Fei. 3d object representations for fine-grained categorization. In *Proceedings of the IEEE international conference on computer vision workshops*, pages 554–561, 2013. 5

[20] Timo Lüddecke and Alexander Ecker. Image segmentation using text and image prompts. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 7086–7096, 2022. 8

[21] Subhransu Maji, Esa Rahtu, Juho Kannala, Matthew Blaschko, and Andrea Vedaldi. Fine-grained visual classification of aircraft. *arXiv preprint arXiv:1306.5151*, 2013. 5

[22] 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. 5

[23] Omkar M Parkhi, Andrea Vedaldi, Andrew Zisserman, and CV Jawahar. Cats and dogs. In *2012 IEEE conference on computer vision and pattern recognition*, pages 3498–3505. IEEE, 2012. 5

[24] 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 *International conference on machine learning*, pages 8748–8763. PMLR, 2021. 1, 2, 8

[25] Yongming Rao, Wenliang Zhao, Guangyi Chen, Yansong Tang, Zheng Zhu, Guan Huang, Jie Zhou, and Jiwen Lu. Denseclip: Language-guided dense prediction with context-aware prompting. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 18082–18091, 2022. 8

[26] Benjamin Recht, Rebecca Roelofs, Ludwig Schmidt, and Vaishaal Shankar. Do imagenet classifiers generalize to imagenet? In *International conference on machine learning*, pages 5389–5400. PMLR, 2019. 5

[27] Shuvendu Roy and Ali Etemad. Consistency-guided prompt learning for vision-language models. *arXiv preprint arXiv:2306.01195*, 2023. 8

[28] 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. 5

[29] Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*, 2023. 1

[30] Haohan Wang, Songwei Ge, Zachary Lipton, and Eric P Xing. Learning robust global representations by penalizing local predictive power. *Advances in Neural Information Processing Systems*, 32, 2019. 5

[31] Jianxiong Xiao, James Hays, Krista A Ehinger, Aude Oliva, and Antonio Torralba. 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, 2010. 5

[32] Hantao Yao, Rui Zhang, and Changsheng Xu. 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, 2023. 8

[33] Hantao Yao, Rui Zhang, and Changsheng Xu. Tcp: Textual-based class-aware prompt tuning for visual-language model. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 23438–23448, 2024. 8

[34] Lewei Yao, Runhui Huang, Lu Hou, Guansong Lu, Minzhe Niu, Hang Xu, Xiaodan Liang, Zhenguo Li, Xin Jiang, and Chunjing Xu. Filip: Fine-grained interactive language-image pre-training. *arXiv preprint arXiv:2111.07783*, 2021. 8

[35] Yuhang Zang, Wei Li, Kaiyang Zhou, Chen Huang, and Chen Change Loy. Open-vocabulary detr with conditional matching. In *European Conference on Computer Vision*, pages 106–122. Springer, 2022. 8

[36] Xiaohua Zhai, Xiao Wang, Basil Mustafa, Andreas Steiner, Daniel Keysers, Alexander Kolesnikov, and Lucas Beyer. Lit: Zero-shot transfer with locked-image text tuning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 18123–18133, 2022. 8

[37] Ji Zhang, Jingkuan Song, Lianli Gao, and Hengtao Shen. Free-lunch for cross-domain few-shot learning: Style-aware episodic training with robust contrastive learning. In *Proceedings of the 30th ACM International Conference on Multimedia*, pages 2586–2594, 2022. 8

[38] Ji Zhang, Lianli Gao, Xu Luo, Hengtao Shen, and Jingkuan Song. Deta: Denoised task adaptation for few-shot learning. In *ICCV*, pages 11541–11551, 2023. 8

[39] Ji Zhang, Shihan Wu, Lianli Gao, Heng Tao Shen, and Jingkuan Song. Dept: Decoupled prompt tuning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12924–12933, 2024. 5, 8

[40] Renrui Zhang, Rongyao Fang, Wei Zhang, Peng Gao, Kunchang Li, Jifeng Dai, Yu Qiao, and Hongsheng Li. Tip-adapter: Training-free clip-adapter for better vision-language modeling. *arXiv preprint arXiv:2111.03930*, 2021. 8

[41] Kaiyang Zhou, Jingkang Yang, Chen Change Loy, and Ziwei Liu. Conditional prompt learning for vision-language models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 16816–16825, 2022. 8

[42] Kaiyang Zhou, Jingkang Yang, Chen Change Loy, and Ziwei Liu. Learning to prompt for vision-language models. *International Journal of Computer Vision*, 130(9):2337–2348, 2022. 2, 3, 8

[43] Xingyi Zhou, Rohit Girdhar, Armand Joulin, Philipp Krähenbühl, and Ishan Misra. Detecting twenty-thousand classes using image-level supervision. In *European Conference on Computer Vision*, pages 350–368. Springer, 2022. 8

[44] Beier Zhu, Yulei Niu, Yucheng Han, Yue Wu, and Hanwang Zhang. Prompt-aligned gradient for prompt tuning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 15659–15669, 2023. 8