Prompt Tuning based Adapter for Vision-Language Model Adaption

Jingchen Sun
University at Buffalo
jsun39@buffalo.edu

Jiayu Qin
University at Buffalo
jiayuqin@buffalo.edu

Zihao Lin
Duke University
zihao.lin@duke.edu

Changyou Chen
University at Buffalo
cchangyou@gmail.com

Abstract

Large pre-trained vision-language (VL) models have shown significant promise in adapting to various downstream tasks. However, fine-tuning the entire network is challenging due to the massive number of model parameters. To address this issue, efficient adaptation methods such as prompt tuning have been proposed. We explore the idea of prompt tuning with multi-task pre-trained initialization and find it can significantly improve model performance. Based on our findings, we introduce a new model, termed Prompt-Adapter, that combines pre-trained prompt tunning with an efficient adaptation network. Our approach beat the state-of-the-art methods in few-shot image classification on the public 11 datasets, especially in settings with limited data instances such as 1 shot, 2 shots, 4 shots, and 8 shots images. Our proposed method demonstrates the promise of combining prompt tuning and parameter-efficient networks for efficient vision-language model adaptation. The code is publicly available at: https://github.com/Jingchensun/prompt_adapter

1. Introduction

In recent years, there has been a growing interest in developing Vision-Language (VL) models that can perform joint reasoning over visual and textual information. Large-scale VL models, such as CLIP [30] and ALIGN [16], have shown impressive zero-shot transfer learning ability on downstream tasks [12, 18], including image classification [21, 41, 23] and open-vocabulary object detection [13, 38, 42, 23]. These models are pre-trained on web-scale images and text pairs [30] and contain a lot of cross-domain knowledge. However, adapting these large-scale VL models to downstream tasks is still quite challenging due to the size and complexity of the models.

Several methods have been proposed to adapt large pre-trained VL models. One of the most commonly used methods is fine-tuning, which involves updating the model's parameters on the task-specific dataset. However, fine-tuning the entire network is computationally expensive and may lead to overfitting, especially when the target dataset is small. As an alternative, two methods have been proposed: prompt tuning [11, 24, 26] and parameter-efficient [22, 15] tuning. Prompt tuning involves adding an extra set of words or learnable parameters that are fed into the text encoder, allowing the model to obtain task-specific outputs. In contrast, parameter-efficient tuning involves adding an extra network or parameters to learn the representation of the downstream tasks, which reduces the computational cost of model adaptation.

Prompt learning is a technique used to fine-tune pre-trained language models for specific downstream tasks. It involves providing a prompt, which is a natural language statement or question that constrains the model to generate a specific output. Prompt learning has shown great promise in improving the zero-shot transfer learning ability of large-scale VL models. There are mainly three types of prompt tuning methods: text prompt tuning, visual prompt tuning, and unified prompt tuning. The representative work of text prompt tuning is CoOp [41]. CoOp injects additional text as input into the text encoder to help guide the model toward the desired output. Visual prompt tuning works, like VPT [17] and visual prompting [2], by injecting additional parameters into multiple layers of the vision transformer to optimize the image features output. Unified prompt tuning [39] combines text and visual prompts for a better trade-off between the two. However, prompt-based methods are highly impacted by the dataset and need longer training time to achieve optimal results.

Tip-Adapter [40] is a recent method proposed for adapting the CLIP model to new downstream tasks in a training-free manner. The approach appends a non-parametric cache model to the weight-frozen CLIP model, where the cache model stores few-shot visual features encoded by CLIP and their ground-truth labels under one-hot encodings. During inference, the cache model is used to retrieve the few-shot knowledge and incorporate it with CLIP's pre-trained knowledge to achieve high performance on downstream tasks. Tip-Adapter has the advantage of not requiring fine-tuning or additional training for adapting CLIP to new tasks, which significantly reduces computational costs. However,

it has the drawback of relying on manual prompts for the text encoder, which may not fully capture the knowledge of the CLIP text encoder.

In this research, we present a novel approach that integrates prompt tuning and parameter-efficient networks to overcome the limitations of each individual method. Specifically, we propose to generalize the manual prompting of the Tip-Adapter with learnable prompts adapted from CoOp's text prompt. We demonstrate the effectiveness of our approach in few-shot image classification. Our methods have two variants called Prompt-Adapter and Prompt-Adapter-F. Prompt-Adapter is a variant that is only based on the prior knowledge of pre-trained text prompt and cache model, it does not need training and can outperform the previous. Furthermore, we extend Prompt-Adapter by utilizing a multi-task trained prompt to initialize the text prompt and achieve a further 1.55% improvement in classification accuracy. For another variant of our work Prompt-Adapter-F, we train the network with 20 epochs and obtain a 0.49% improvement in accuracy compared to previous methods, setting a new state-of-the-art on few-shot classification. Our results suggest that the integration of prompt tuning and parameter-efficient networks can enhance the efficiency and performance of VL model adaptation for image classification.

In summary, we propose a novel approach that combines multi-task pre-trained prompt learning with parameter-efficient networks to achieve efficient few-shot image classification. Our contributions include 1) showing the effectiveness of the multi-task pre-trained prompt mechanism in improving single-task prompt recognition accuracy, 2) proposing a new network architecture that incorporates prompt learning and a parameter-efficient network, and 3) demonstrating the superiority of our approach through few-shot image classification experiments on 11 datasets. The results show that our approach outperforms state-of-the-art models in terms of accuracy, especially in extreme situations where data are limited available. Our method highlights the potential of combining prompt learning and parameter-efficient networks for efficient vision-language model adaptation.

2. Related Work

Vision Language Model. Vision and language models have become increasingly popular in recent years and have shown remarkable success in various computer vision tasks [41, 13, 38, 42, 12, 18]. These models typically consist of an image encoder and a text encoder, trained using contrastive loss on large-scale image-text pairs. CLIP [30] and ALIGN [16] are some of the most prominent models in this domain. CLIP (Contrastive Language-Image Pre-training) is a large-scale vision language model trained on 400 million image-text pairs that have demonstrated impressive transferability in cross-domain downstream tasks. In this context, our work focuses on transferring CLIP into 11 cross-domain image classification tasks. The integration of prompt learning-based methods and parameter-efficient methods has also shown significant improvement in few-shot performance with small computing resources, making it an area of active research.

Parameter Tuning Large pre-trained vision language models like CLIP have achieved state-of-the-art performance in various downstream tasks, but their high computational requirements make them difficult to deploy in resource-constrained environments. To address this issue, several approaches [22, 10, 40] have been proposed to reduce the number of parameters and computations required for inference while maintaining high accuracy. CLIP-Adapter [10] appends a lightweight two-layer Multi-Layer Perceptron (MLP) to the pre-trained weight-fixed CLIP model and optimizes its parameters via stochastic gradient descent (SGD). Tip Adapter [40] constructs a non-parametric cache model that stores features of training images and their labels as a key-value database. By aggregating the information from the cache model with the text features, Tip-Adapter significantly boosts the classification accuracy without training over Zero-shot CLIP. However, Tip-Adapter still uses the manual Prompt as an image classifier, which can not fully utilize the huge knowledge of the text encoder of the CLIP model.

Prompt Tuning Prompt engineering [11, 24, 26] has been widely used in natural language processing (NLP) to improve the performance of language models on specific tasks. A prompt is a piece of text that is added to the input to guide the model toward a particular output. Prompts can be used to provide additional information to the model, such as a task description or a set of constraints. Prompt engineering involves designing effective prompts that can guide the model toward the desired output. For example, by feeding a manual text prompt "a photo of a {}" to the text encoder [30], the CLIP model has shown strong zero-shot image recognization ability.

Meanwhile, prompt tuning aims to learn an optimal prompt automatically through fine-tuning or meta-learning. In recent studies, several methods have been proposed to improve the efficiency and effectiveness of prompt engineering and learning. For example, CoOp [41] introduced a collaborative optimization method that jointly optimizes prompts and model parameters. On the other hand, Visual Prompt Tuning (VPT) [17] uses a visual prompt that is tailored to the image input, improving the model's performance on image classification tasks. Unified Prompt Tuning (UPT) [39] is a recent method that learns a unified prompt for multiple tasks, which leads to better performance in downstream tasks. These methods have shown promising results in improving the performance of vision

and language models, and we aim to build upon their success in our work.

Few-shot Learning Few-shot learning has been a challenging research topic in image classification, where the objective is to recognize novel classes from only a few training examples [8, 36, 1, 25]. This problem has garnered significant attention in the computer vision community, as it is more realistic and practical in many real-world scenarios where labeled data is scarce or costly to obtain. In recent years, various methods have been proposed for few-shot image classification.

One of the earliest approaches for few-shot learning was Siamese neural networks [19], which used a distance metric to compare images and learn to distinguish between classes. Later, the idea of meta-learning was introduced [31], which trains a model to learn how to learn from few examples. This approach led to the development of popular few-shot learning algorithms like Matching Networks [4], Prototypical Networks [33, 7], and Relation Networks [35]. These methods have shown promising results on various benchmark datasets, but they usually rely on simple feature extractors and do not scale well to large-scale image classification problems.

Recently, CLIP's [30] ability to jointly reason about text and images has inspired a new line of research on few-shot learning for vision and language models. Various adaptation methods, such as CoOp [41], UPT [39], and Tip-Adapter [40] have been proposed to fine-tune the CLIP model on few-shot datasets. These methods have achieved state-of-the-art results on several benchmark datasets, demonstrating the effectiveness of adapting large-scale vision and language models to few-shot learning tasks.

3. Method

We first revisit the CLIP, CoOp, and Tip-Adapter in Section 3.1, then present the technical details of our proposed method in Section 3.2.

3.1. Preliminaries

CLIP [30] model is a large-scale neural network model, which is trained on a diverse set of image-text pairs to learn the relationship between the visual and textual features. Unlike traditional vision models, CLIP leverages the massive amount of textual and visual information available on the internet to learn more robust and accurate representations.

The CLIP model consists of two encoders: an image encoder and a text encoder, which are trained together using contrastive loss to project images and the corresponding text descriptions into a common embedding space. CLIP jointly trains an image encoder ψ and a text encoder ϕ and uses a symmetric contrastive loss to match the batch of image-text pairs. The training objective L_{CLIP} is:

$$L_{CLIP} = L_{i2t} + L_{t2i} \tag{1}$$

with L_{i2t} representing an image-to-text contrastive loss and L_{t2i} a text-to-image contrastive loss. The contrastive loss is calculated as follows:

$$L_{i2t} = -\sum_{i \epsilon ß} \log \frac{\exp\left(\cos\left(u_i, v_i\right)/\tau\right)}{\sum_{j \epsilon ß} \exp\left(\cos\left(u_i, v_j\right)/\tau\right)} \tag{2}$$

$$L_{t2i} = -\sum_{j \epsilon ß} \log \frac{\exp\left(\cos\left(u_j, v_j\right)/\tau\right)}{\sum_{j \epsilon ß} \exp\left(\cos\left(u_i, v_j\right)/\tau\right)} \tag{3}$$

where $u = \psi(x)$ represents the projection of image x to the final hidden space, $v = \phi(y)$ indicates the projection of text y to the final embedding. cos denotes the cosine similarity; τ is a learnable temperature value.

The joint contrastive learning allows the CLIP to learn to associate textual descriptions with visual features, enabling it to perform a variety of tasks such as image classification [21], object detection [42, 13], segmentation [12], etc.

CoOp Context Optimization [41] is a simple approach that is specifically designed for adapting large-scale vision-language models for downstream image recognition tasks. CoOp is built on the principle of optimizing the context, or prompt, to improve the model's ability to recognize images. The method is based on the insight that the context or prompt can significantly impact the performance of the model. CoOp aims to find the optimal prompt by iteratively refining it based on the performance of the model on the downstream task.

CoOp utilized a number of unified contexts to model context words with continuous vectors for downstream image recognition. It is the first prompt-based network used in the CLIP model adapting. They designed the unified context vectors in the text encoder of the CLIP model and shares the same context with all classes. The prompt given to the text encoder $g(\cdot)$ is designed with the following form,

$$t = [\mathrm{V}]_1[\mathrm{V}]_2 ... [\mathrm{V}]_M[\mathrm{C}lass] \tag{4}$$

where each $[\mathrm{V}]_1[\mathrm{V}]_2 ... [\mathrm{V}]_M$ is a vector with the same dimension as word embeddings (i.e., 512 for CLIP), and M is a hyperparameter determining the number of context tokens. The CoOp approach has demonstrated impressive results in adapting CLIP-like vision-language models for downstream image recognition through its use of text prompt learning vectors. Despite this, one of the major drawbacks of CoOp is its relatively long training time, typically requiring 200 epochs to achieve optimal performance compared to other methods. Therefore, there is a need for further research to explore ways to reduce the training time of CoOp while maintaining its effectiveness in adapting large vision-language models.

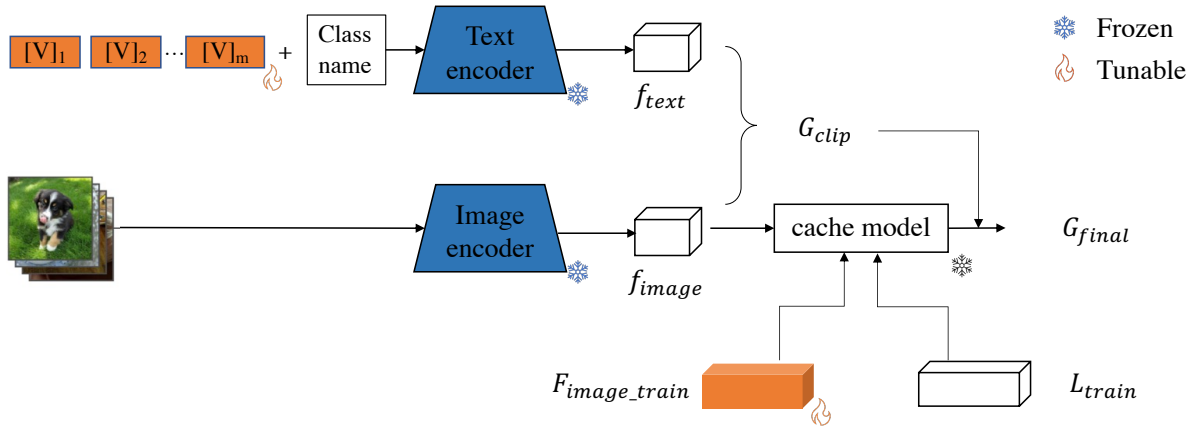

Figure 1. An illustration of our method Prompt Adapter. The learned prompt and the class name are sent to the frozen CLIP text encoder. While the test images are sent to the frozen CLIP image encoder. The G_{clip} is obtained by calculating the cosine similarity between text embeddings and image features. Further, the G_{cache} is obtained by the cache model. The Final logit G_{final} is a linear combination of G_{cache} and G_{clip}. The G_{final} is used to classify the images.

Tip-Adapter [40] is a recent parameter-efficient approach for adapting CLIP-like vision-language models to downstream image recognition tasks. Unlike traditional fine-tuning methods, Tip-Adapter constructs a non-parametric cache model that stores the features of training images and their corresponding labels as a key-value database. Given K-shot images with N-class training samples, we use I_K to represent the image features of K-shot images and L_N to represent their labels. The key in Tip-Adapter is the pre-trained $N * K$ image feature F_{train} extracted by the CLIP image encoder, while the value is an N-dimensional one-hot vector L_{train}, which represents the ground truth label. The F_{train} and L_{train} are represented by:

$$\text{F}_{\text{train}} = \text{VisualEncoder}(I_K) \tag{5}$$

$$\text{L}_{\text{train}} = \text{OneHot}(L_N) \tag{6}$$

After constructing the cache model, the adaption of CLIP can be simply achieved by two matrix-vector multiplications $f_{\text{test}}\text{F}_{\text{train}}$. The term $f_{\text{test}}\text{F}_{\text{train}}^T$ is equivalent to the cosine similarities between test feature f_{test} and all few-shot training features F_{train}. Then, the prediction for the cache model can be obtained via a linear combination of cached values weighted $A\text{L}_{\text{train}}$ and the original CLIP logits. The A is calculated as:

$$A = \exp\left(-\beta(1 - f_{\text{test}}\text{F}_{\text{train}}^T)\right) \tag{7}$$

and the final output logits of the test image by Tip-Adapter are then calculated as:

$$\text{Logits} = \alpha(A)\text{L}_{\text{train}} + f_{\text{test}}W_c^T \tag{8}$$

By aggregating the information from the cache model with the text features, Tip-Adapter can significantly boost classification accuracy without the need for further training on the CLIP model. This approach leverages the prior knowledge encoded in CLIP by feature retrieval and can incorporate new knowledge from the few-shot training set.

When given more shots, Tip-Adapter lags behind the training-required methods without training. Thus the author proposed an augmented version called Tip-Adapter-F [40], which treats the keys in the cache model as learnable parameters, and fine-tunes them via SGD. Tip-Adapter-F achieves state-of-the-art performance on the few-shot adapting of the CLIP model. However, Tip-Adapter still relies on manual prompts as image classifiers, limiting its ability to fully utilize the vast knowledge of the text encoder of the CLIP model.

3.2. Our Method

Tip-Adapter represents the state-of-the-art few-shot learning approach for CLIP-based vision-language models. However, it still relies on manual prompts for image classification and is limited in its ability to fully utilize the vast knowledge of the CLIP text encoder. Our proposed Prompt-Adapter overcomes this limitation by incorporating text prompt learning with Tip-Adapter, which can achieve better few-shot learning performance. As shown in Figure 1, our Prompt-Adapter adopts a similar network structure as Tip-Adapter but replaces the original manual prompt "a photo of a " with learnable vectors $[\text{V}]_1[\text{V}]_2...[\text{V}]_M$. By doing so, our approach can optimize the text prompt to better capture the essential features of the few-shot training set and improve classification accuracy.

Our Prompt-Adapter has two variants. One is the training-free variant, denoted as Prompt-Adapter; and the other is the learnable variant, denoted as Prompt-Adapter-F.

In **Prompt-Adapter**, we do not need the training phase,

and only rely on the cache model and fixed prompt. We first construct the cache model with the few-shot train images and the one-hot ground truth label. And then we directly use the learned prompt from CoOp.

When testing the model, the learned prompt and class names are sent to the text encoder to obtain the text embeddings f_{text}. And in the image branch, the test images are sent to the CLIP image encoder and obtained image features $f_{\text{image\_test}}$. Then we calculate the clip logits G_{clip}, which is the cosine similarity between f_{text} and $f_{\text{image\_test}}$, given by:

$$G_{\text{clip}} = f_{\text{text}} \cdot f_{\text{image\_test}} \tag{9}$$

The clip logits G_{clip} can be used to select the most matched image and label pairs for classification. In practice, we do not only rely on the clip logits to classify images. Thus, we further calculate the cache logits G_{cache}, which is the cosine similarity between the test image features and the cached image features. The cache logits is given by:

$$G_{\text{cache}} = \exp\left(-\beta(1 - f_{\text{image\_test}} \cdot \mathrm{F}^{T}_{\text{image\_train}})\right)\mathrm{L}_{\text{train}} \tag{10}$$

where $\mathrm{F}_{\text{image\_train}}$ and $\mathrm{L}_{\text{train}}$ represent the train images and the one-hot labels in the cache model. And $f_{\text{image\_test}} \cdot \mathrm{F}_{\text{image\_train}}$ represent the cosine similarity between test image features and trained image features; an exponential function is applied to convert similarities into positive values; and β stands for a modulating hyperparameter to control the degree of sharpness [40]. The final logits are a linear combination of cache logits and clip logits, given by:

$$G_{\text{final}} = \alpha \cdot G_{\text{cache}} + G_{\text{clip}} \tag{11}$$

where α is the weight coefficient to balance the information from the training data and the prior knowledge of the CLIP model [40]. The final Logits can be used to select the most matched image and text pairs. Thus we can use the final Logits to realize image classification.

In **Prompt-Adapter-F** , the image features in the cache model are learnable. There are training phases and test phases in the method. In the training phase, training images are sent to the image encoder, and the output of the image encoder is denoted as $f_{\text{image\_train}}$. The cache logits is adapted to:

$$G_{\text{cache}} = \exp\left(-\beta(1 - f_{\text{image\_train}} \cdot \mathrm{F}^{T}_{\theta})\right)\mathrm{L}_{\text{train}} \tag{12}$$

where F_θ represents the learned parameters of cache features. The final logits of the Prompt Adapter F are the same as equation 11. Thus, the learned parameters can be optimized by the cross-entropy loss, here G_{Target} refers to ground truth label logits:

$$L_{\text{prompt\_adapter}} = CE(G_{\text{final}}, G_{\text{Target}}) \tag{13}$$

where CE represents the cross entropy loss function. When training Prompt-Adapter-F, two learnable parameters are present in the network: the text prompt $[\mathrm{V}]_1[\mathrm{V}]_2...[\mathrm{V}]_M$ and the cache features F_θ. To optimize these parameters, we employ two different strategies: the separately optimized strategy and the joint optimization strategy. In the separately optimized strategy, we first optimize the text prompt. Specifically, the learnable vectors $[\mathrm{V}]_1[\mathrm{V}]_2...[\mathrm{V}]_M$ are optimized by the cross entropy loss function between clip logits and ground truth labels. The loss function L_{prompt} is given by:

$$L_{\text{prompt}} = CE(G_{\text{clip}}, G_{\text{Target}}) \tag{14}$$

After we trained the prompt, we froze the parameters of the text prompt, and then optimize F_θ until we obtain the best performance. On the other hand, the joint optimization strategy directly optimizes the final output logits G_{final}. Because G_{final} is a linear combination with the clip logits G_{clip} and the cache logits G_{cache}. When we use the cross-entropy function to optimize the final output logits, the optimizer will automatically update the parameter in the text prompt and the cache features at the same time. Results from our ablation study will show that the separately optimized strategy performs better than the joint optimization strategy. Hence, we use the separately optimized strategy as the default strategy in our experiment.

Multi-task Initialization An important phenomenon we found is that the prompt initialization method has a significant impact on the performance of the model. Our default initialization approach involves randomly initializing the text prompt and using few-shot images and labels to train the prompt. To improve model performance, inspired by previous works [32], we first use 11 datasets as the source tasks, we train the shareable prompt among the 11 datasets. And then we use the shareable prompt as initialization and adapt the network to a single task. For the single-task prompt learning, we directly optimize the prompt with the cross-entropy loss function of each task. Once the sign-task prompt is learned, the single-task prompt will be used as the final text prompt in our network.

4. Experiment

To comprehensively compare our methods with other works, we first conducted detailed experiments for 20-shot image classification, where there are only 20 labeled examples available for each class. Then, we extended the experiment set to include results for 1, 2, 4, 8, and 16 shots. Finally, we performed an ablation study to investigate the effects of different initialization methods and training strategies on the performance of the network.

4.1. Few Shot Image Classfication

Datasets. Following previous works such as CoOp [41] and Tip-Adapter [40], we use 11 publicly available im-

age classification datasets. These datasets include ImageNet [6], Caltech101 [9], OxfordPets [29], StanfordCars [20], Flowers102 [28], Food101 [3], FGVCAircraft [27], SUN397 [37], DTD [5], EuroSAT [14], and UCF101 [14]. By selecting a diverse range of datasets, we aimed to ensure that the evaluation was comprehensive and our methods can generalize across different domains.

ImageNet [6] is a massive dataset containing over 14 million images with more than 20,000 object categories. Caltech101 [9] is a smaller dataset consisting of 101 object categories, containing about 9,000 images. The OxfordPets [29] dataset includes over 7,000 images of pets in various categories, such as cats, dogs, and birds. StanfordCars [20] is a dataset of cars consisting of over 16,000 images of 196 car models. Flowers102 [28] is a dataset with over 8,000 flower images in 102 categories. Food101 [3] is a dataset consisting of 101 food categories, including pizza, sushi, and burgers. FGVCAircraft [27] is a dataset containing over 10,000 images of aircraft, including commercial and military planes. SUN397 [37] is a scene recognition dataset that includes over 130,000 images of indoor and outdoor scenes. DTD (Descriptive Textures) [5] is a dataset with 47 texture categories, including fabrics, tiles, and plants. EuroSAT [14] is a dataset with 27,000 satellite images of ten land-use classes. Finally, UCF101 [34] is a dataset with 13,000 videos in 101 action categories, such as playing basketball or brushing teeth. Table 1 is the statistics of these 11 datasets.

Table 1. The statistics of these 11 image classification datasets.

	classes	train	val	test
Oxford_Pets	37	2,944	736	3,669
Flowers102	102	4,093	1,633	2,463
Fgvc_Aircraft	100	3,334	3,333	3,333
Describable Textures	47	2,820	1,128	1,692
Eurosat	10	13,500	5,400	8,100
Stanford_Cars	196	6,509	1,635	8,041
Food101	101	50,500	20,200	30,300
Sun397	397	15,880	3,970	19,850
Caltech101	100	4,128	1,649	2,465
Ucf101	101	7,639	1,898	3,783
ImageNet	1,000	1.28M	N/A	50,000

Training Details. When training the learned prompt $[\mathrm{V}]_1[\mathrm{V}]_2...[\mathrm{V}]_M$, the maximum training epoch of the ImageNet dataset is fixed at 50. While for the other 10 datasets, the maximum training epoch is set to 200 epochs for 16/8 shots, 100 epochs for 4/2 shots, and 50 epochs for 1 shot. The optimizer is SGD and the learning rate is 0.002 with batch size 32. And the learning rate is scheduled by the cosine annealing rule. We follow all other settings as the default settings in CoOp [41] for fairly comparing.

While training for the cache features F_θ, we follow the default settings on Tip-Adapter [40] and set 100-epoch training for the EuroSAT dataset and only 20-epoch training for the other 10 datasets. All the experiments are done on 1, 2, 4, 8, 16, and 20 shots training sets, and test on the full test sets. The initial learning rate is 0.001 with a batch size of 256, and the AdamW optimizer with a cosine scheduler.

Baseline. We compare our method with various existing methods to evaluate its effectiveness. In particular, we use the following five methods as baselines for comparison purposes: 1) Zero-shot CLIP [30]: This baseline method relies solely on the CLIP model without any parameter fine-tuning. The hand-crafted text prompt template "a photo of a {}" is used as the text encoder input. 2) Linear-probe CLIP [30]: This method involves adding an additional linear classifier on top of the feature extraction layer of the frozen CLIP model. The classifier is trained on a few-shot training set to learn the corresponding labels. 3) CoOp [41]: This method aims to optimize the learnable vectors as the text prompt. Specifically, it employs a context optimization technique to improve the performance of the CLIP model. 4) UPT [39]: This method presents a unified text and visual prompt tuning approach. It employs a lightweight self-attention network to generate the prompt for CLIP's text and visual encoders. 5) Tip-Adapter [40]: This method is based on a key-value cache model that stores and retrieves knowledge. It is a training-free adaptation method that enables parameter-efficient adaptation of large vision and language models.

For fair comparisons, we use the same backbone with ViT-B16, and data preprocessing with CLIP. We report the average results of three random seed-running experiments to reduce variance and increase result robustness. In addition, we adopt the default settings for each baseline method as described in the original papers.

Few-shot Learning Results We first report results on the 20-shot image setting for the Prompt-Adapter network. After training, we freeze the prompt and fine-tune the learnable cache features. We also use 20-shot images to train other baseline methods. The evaluation results are presented in Table 2, which shows that our Prompt Adapter achieves an average classification accuracy of 78.44%, which is 1.1% higher than the Tip-Adapter's (77.35%) accuracy. This improvement can be attributed to the pre-trained prompt that contains rich text knowledge. Furthermore, when finetuning the learnable cache features with Prompt-Adapter-F, the classification accuracy reaches 81.52%, which is 0.06% higher than Tip-Adapter-F's (81.46%) and Unified Prompt Learning's (81.44%) accuracy. Remarkably, our Prompt-Adapter-F outperforms the current state-of-the-art methods even when trained on 20-shot images.

For experiments with 1, 2, 4, and 8, 16 shot settings, we plot out the learning curves for different methods in Figure 2. By looking at the average results over 11 datasets, we

Table 2. Few-shots image classification for different methods.

	Oxford_pets	Flowers102	FGVCAircraft	DTD	EuroSAT	StanfordCars	Food101	SUN397	caltech101	ucf101	ImageNet	Average
Zero-Shot Clip	89.13	70.65	24.87	44.03	48.26	65.55	85.88	62.58	93.27	67.70	68.79	65.52
CoCp	91.53	96.40	40.30	69.47	84.00	79.20	85.00	74.43	95.70	82.40	71.60	79.09
Tip-Adapter	91.88	94.60	39.96	66.08	78.01	75.39	86.45	71.94	95.09	78.51	70.32	77.11
Prompt-Adapter	88.12	96.55	43.08	70.33	82.86	79.49	83.76	72.91	94.93	80.23	70.63	**78.44**
Tip-Adapter-F	93.08	96.31	45.45	71.93	87.43	83.82	87.38	76.30	95.94	85.01	73.45	81.46
Unifed Prompt Tunning	92.95	97.11	46.80	70.65	90.51	84.33	85.00	75.92	95.94	84.03	72.63	81.44
Prompt-Adapter-F	92.40	98.05	49.08	71.45	87.56	83.39	86.96	75.81	95.66	83.95	72.36	**81.52**

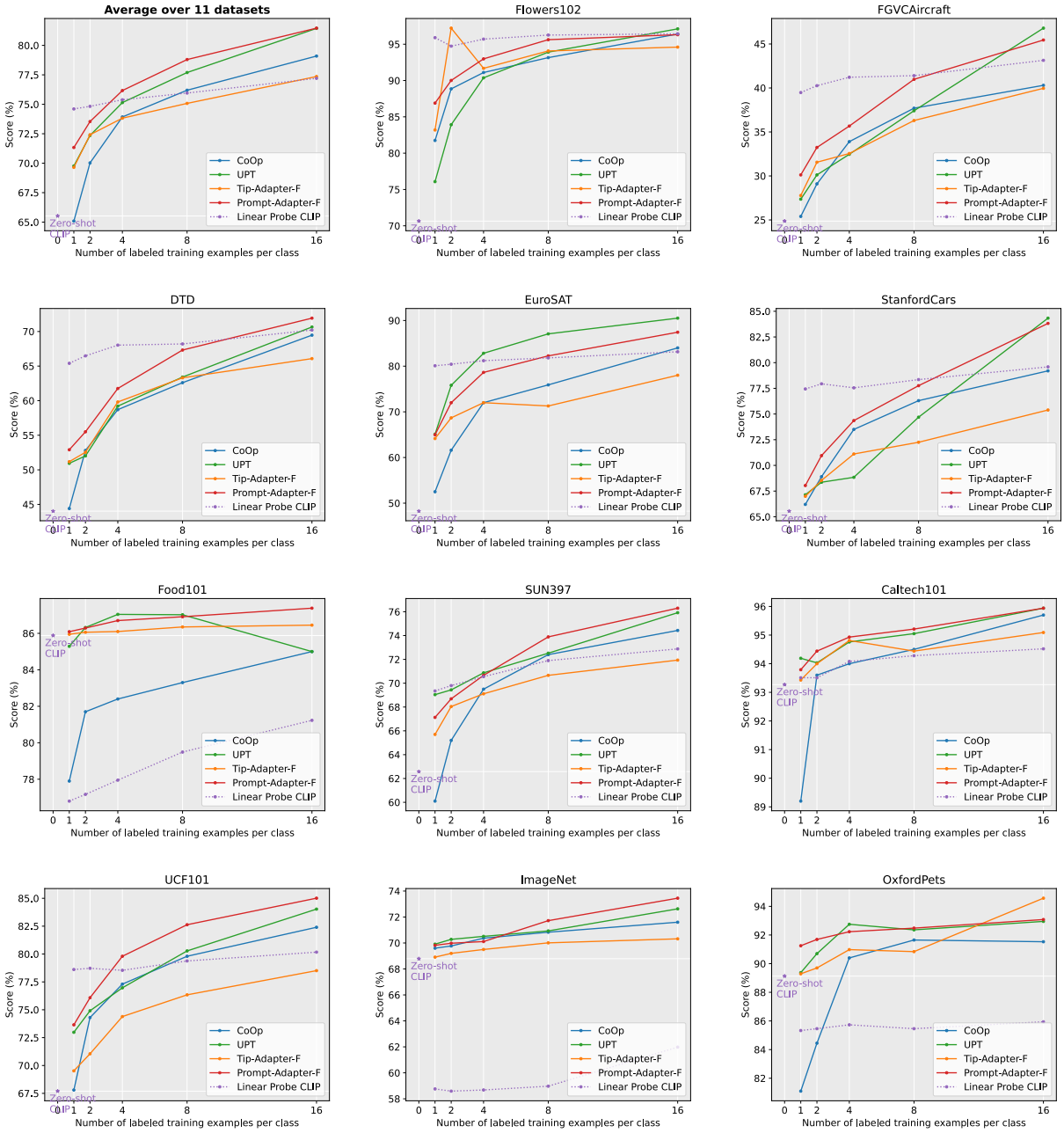

Figure 2. Few-shot image classification of 11 datasets.

Table 3. Ablation study of different initialization ways for prompt learning.

	Oxford_Pets	Flowers102	FGVCAircraft	DTD	EuroSAT	StanfordCars	Food101	SUN397	Caltech101	UCF101	ImageNet	Average
Handcrafted prompt	89.21	71.34	24.72	44.39	47.60	65.32	86.06	62.50	92.94	66.75	66.73	65.23
Random initialization	92.53	96.47	42.91	68.50	80.87	83.09	87.21	75.29	95.77	82.24	71.92	79.71
Mannul initialization	91.53	96.40	40.30	69.47	84.00	79.20	85.00	74.43	95.70	82.40	71.60	79.09
Pretrained initization	91.74	97.20	44.91	70.21	86.02	82.20	87.37	75.78	95.98	83.45	72.23	80.64

Table 4. Ablation study of training strategies.

		Oxford_Pets	Flowers102	FGVCAircraft	DTD	EuroSAT	StanfordCars	Food101	SUN397	Caltech101	UCF101	Average
Joint Training	Prompt-Adapter-F	76.25	84.37	33.61	62.40	76.68	55.42	60.73	63.90	84.13	75.53	67.30
Separate Training	Prompt-Adapter-F	91.66	97.89	48.39	71.28	85.37	82.91	86.35	75.56	95.74	84.43	81.96

find that our method Prompt-Adapter-F achieves the highest accuracy among all the few-shot settings. Specifically, our method outperforms CoOp [41], UPT [39], and Tip-Adapter-F [40] in the 1-shot, 2 shots, 4 shots, and 8 shots experiments. These results indicate that our method can effectively handle scenarios with extremely limited data availability. Notably, our method has demonstrated superior performance compared to UPT [39]. Because UPT needs 200 epochs of training time to achieve such performance while our method only needs 20 epochs of training time. This phenomenon further highlighting the efficacy of our approach. These findings support the claim that our Prompt Adapter approach is a promising technique for few-shot image classification tasks.

In addition to achieving high accuracy on the average classification accuracy across 11 datasets, our method also outperforms other state-of-the-art methods on individual datasets. Specifically, on datasets such as Flowers102 [28], FGVCAircraft [27], DTD [5], StanfordCars [20], Food101 [3], SUN397 [37], Caltech101 [9], and UCF101 [34], the proposed Prompt-Adapter-F demonstrates superior performance compared to other methods by significant margins. Only on some datasets like EuroSAT [14] and OxfordPets [29] our method has some slight decreases in accuracy compared with other methods. We claim the reason may be these two datasets have high intra-class visual feature variance [39], making text prompt adaption difficult.

In summary, the experimental results demonstrate that our Prompt-Adapter approach is effective and robust for few-shot image classification tasks. The pre-trained prompt and learnable cache features synergistically contribute to the superior performance of our method.

4.2. Ablation Study

In this study, we investigate the effect of initialization and training strategies on the performance of a neural network designed for language generation tasks. Specifically, we evaluate three different initialization methods and two training strategies for the network.

Regarding the initialized way of the prompt, we test three methods, including random initialization, manual initialization, and pre-trained prompt initialization. Manual initialization means using the embeddings of "a photo of a" to initialize the context vectors [41]. Both random initialization and manual initialization are single-task initialization ways. And we select the handcraftd prompt as a baseline to compare with. Our results in Table 3 indicate that pre-trained prompt initialization achieves 80.64% accuracy, and outperform the other methods on all three settings. This method involves first training the prompt on multiple datasets to learn the joint representation of the prompt and then adapting it to single-task training. The superiority of this method suggests that pre-training the prompt can effectively capture the underlying features of the language and enhance the model's performance.

In terms of the training strategy of the network, we employ two approaches, namely joint training and separate training. Our results in Tabel 4 reveal that separate training achieved 81.96% accuracy on average, better than joint training on the tested datasets. This finding can be attributed to the fact that joint training tends to balance the learnable prompt and learnable cache features, which may lead to a degradation in prompt learning. On the other hand, separate training can better capture the prompt features and enhance the model's ability to generate high-quality language.

Overall, our study provides insights into the impact of initialization and training strategies on the performance of neural networks for language generation tasks. The results highlight the importance of pre-training the prompt and separate training strategy.

5. Conclusion

Based on the results of our study, we conclude that the proposed prompt-based adaptation method is an effective approach for efficiently adapting large vision and language models to downstream tasks. By leveraging the strong feature prior knowledge from the cache model and learned text prompt, our method outperforms state-of-the-art approaches on 11 few shots image classification tasks. We believe that our findings can contribute to the community's ongoing efforts to improve the efficiency and effectiveness of vision and language models for a wide range of downstream tasks.

References

[1] Arman Afrasiyabi, Hugo Larochelle, Jean-François Lalonde, and Christian Gagné. Matching feature sets for few-shot image classification. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9014–9024, 2022. 3

[2] Hyojin Bahng, Ali Jahanian, Swami Sankaranarayanan, and Phillip Isola. Exploring visual prompts for adapting large-scale models. *arXiv preprint arXiv:2203.17274*, 1(3):4, 2022. 1

[3] 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. 6, 8

[4] Qi Cai, Yingwei Pan, Ting Yao, Chenggang Yan, and Tao Mei. Memory matching networks for one-shot image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4080–4088, 2018. 3

[5] 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. 6, 8

[6] 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. 6

[7] Shumin Deng, Ningyu Zhang, Jiaojian Kang, Yichi Zhang, Wei Zhang, and Huajun Chen. Meta-learning with dynamic-memory-based prototypical network for few-shot event detection. In *Proceedings of the 13th International Conference on Web Search and Data Mining*, pages 151–159, 2020. 3

[8] Guneet S Dhillon, Pratik Chaudhari, Avinash Ravichandran, and Stefano Soatto. A baseline for few-shot image classification. *arXiv preprint arXiv:1909.02729*, 2019. 3

[9] Li Fei-Fei, Rob Fergus, and Pietro Perona. Learning generative visual models from few training examples: An incremental bayesian approach tested on 101 object categories. In *2004 conference on computer vision and pattern recognition workshop*, pages 178–178. IEEE, 2004. 6, 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. *arXiv preprint arXiv:2110.04544*, 2021. 2

[11] Tianyu Gao, Adam Fisch, and Danqi Chen. Making pre-trained language models better few-shot learners. *arXiv preprint arXiv:2012.15723*, 2020. 1, 2

[12] Golnaz Ghiasi, Xiuye Gu, Yin Cui, and Tsung-Yi Lin. Open-vocabulary image segmentation. *arXiv preprint arXiv:2112.12143*, 2021. 1, 2, 3

[13] 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. 1, 2, 3

[14] 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. 6, 8

[15] Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin De Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. Parameter-efficient transfer learning for nlp. In *International Conference on Machine Learning*, pages 2790–2799. PMLR, 2019. 1

[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. 1, 2

[17] Menglin Jia, Luming Tang, Bor-Chun Chen, Claire Cardie, Serge Belongie, Bharath Hariharan, and Ser-Nam Lim. Visual prompt tuning. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXXIII*, pages 709–727. Springer, 2022. 1, 2

[18] Chen Ju, Tengda Han, Kunhao Zheng, Ya Zhang, and Weidi Xie. Prompting visual-language models for efficient video understanding. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXXV*, pages 105–124. Springer, 2022. 1, 2

[19] Gregory Koch, Richard Zemel, Ruslan Salakhutdinov, et al. Siamese neural networks for one-shot image recognition. In *ICML deep learning workshop*, volume 2. Lille, 2015. 3

[20] 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. 6, 8

[21] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Communications of the ACM*, 60(6):84–90, 2017. 1, 3

[22] Brian Lester, Rami Al-Rfou, and Noah Constant. The power of scale for parameter-efficient prompt tuning. *arXiv preprint arXiv:2104.08691*, 2021. 1, 2

[23] Chunyuan Li, Haotian Liu, Liunian Harold Li, Pengchuan Zhang, Jyoti Aneja, Jianwei Yang, Ping Jin, Yong Jae Lee, Houdong Hu, Zicheng Liu, et al. Elevater: A benchmark and toolkit for evaluating language-augmented visual models. *arXiv preprint arXiv:2204.08790*, 2022. 1

[24] Xiang Lisa Li and Percy Liang. Prefix-tuning: Optimizing continuous prompts for generation. *arXiv preprint arXiv:2101.00190*, 2021. 1, 2

[25] Ying Liu, Hengchang Zhang, Weidong Zhang, Guojun Lu, Qi Tian, and Nam Ling. Few-shot image classification: Current status and research trends. *Electronics*, 11(11):1752, 2022. 3

[26] Yuning Lu, Jianzhuang Liu, Yonggang Zhang, Yajing Liu, and Xinmei Tian. Prompt distribution learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5206–5215, 2022. 1, 2

[27] Subhransu Maji, Esa Rahtu, Juho Kannala, Matthew Blaschko, and Andrea Vedaldi. Fine-grained visual classification of aircraft. *arXiv preprint arXiv:1306.5151*, 2013. 6, 8

[28] 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. 6, 8

[29] 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. 6, 8

[30] 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, 3, 6

[31] Mengye Ren, Eleni Triantafillou, Sachin Ravi, Jake Snell, Kevin Swersky, Joshua B Tenenbaum, Hugo Larochelle, and Richard S Zemel. Meta-learning for semi-supervised few-shot classification. *arXiv preprint arXiv:1803.00676*, 2018. 3

[32] Sheng Shen, Shijia Yang, Tianjun Zhang, Bohan Zhai, Joseph E Gonzalez, Kurt Keutzer, and Trevor Darrell. Multitask vision-language prompt tuning. *arXiv preprint arXiv:2211.11720*, 2022. 5

[33] Jake Snell, Kevin Swersky, and Richard Zemel. Prototypical networks for few-shot learning. *Advances in neural information processing systems*, 30, 2017. 3

[34] 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. 6, 8

[35] Flood Sung, Yongxin Yang, Li Zhang, Tao Xiang, Philip HS Torr, and Timothy M Hospedales. Learning to compare: Relation network for few-shot learning. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1199–1208, 2018. 3

[36] Yonglong Tian, Yue Wang, Dilip Krishnan, Joshua B Tenenbaum, and Phillip Isola. Rethinking few-shot image classification: a good embedding is all you need? In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XIV 16*, pages 266–282. Springer, 2020. 3

[37] 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. 6, 8

[38] Yuhang Zang, Wei Li, Kaiyang Zhou, Chen Huang, and Chen Change Loy. Open-vocabulary detr with conditional matching. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part IX*, pages 106–122. Springer, 2022. 1, 2

[39] Yuhang Zang, Wei Li, Kaiyang Zhou, Chen Huang, and Chen Change Loy. Unified vision and language prompt learning. *arXiv preprint arXiv:2210.07225*, 2022. 1, 2, 3, 6, 8

[40] Renrui Zhang, Wei Zhang, Rongyao Fang, Peng Gao, Kunchang Li, Jifeng Dai, Yu Qiao, and Hongsheng Li. Tip-adapter: Training-free adaption of clip for few-shot classification. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXXV*, pages 493–510. Springer, 2022. 1, 2, 3, 4, 5, 6, 8

[41] 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. 1, 2, 3, 5, 6, 8

[42] Xingyi Zhou, Rohit Girdhar, Armand Joulin, Philipp Krähenbühl, and Ishan Misra. Detecting twenty-thousand classes using image-level supervision. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part IX*, pages 350–368. Springer, 2022. 1, 2, 3