Compound Text-Guided Prompt Tuning via Image-Adaptive Cues

Hao Tan[1,2*], **Jun Li**[1,2*], **Yizhuang Zhou**[3], **Jun Wan**[1,2†], **Zhen Lei**[1,2,4], **Xiangyu Zhang**[3]

[1]MAIS, Institute of Automation, Chinese Academy of Sciences, Beijing, China
[2]School of Artificial Intelligence, University of Chinese Academy of Sciences, Beijing, China
[3]MEGVII Technology
[4]CAIR, HKISI, Chinese Academy of Sciences, Hong Kong, China
{tanhao2023, lijun2021, jun.wan, zhen.lei}@ia.ac.cn, {zhouyizhuang, zhangxiangyu}@megvii.com

Abstract

Vision-Language Models (VLMs) such as CLIP have demonstrated remarkable generalization capabilities to downstream tasks. However, existing prompt tuning based frameworks need to parallelize learnable textual inputs for all categories, suffering from massive GPU memory consumption when there is a large number of categories in the target dataset. Moreover, previous works require to include category names within prompts, exhibiting subpar performance when dealing with ambiguous category names. To address these shortcomings, we propose Compound **T**ext-**G**uided **P**rompt **T**uning (TGP-T) that significantly reduces resource demand while achieving superior performance. We introduce text supervision to the optimization of prompts, which enables two benefits: 1) releasing the model reliance on the pre-defined category names during inference, thereby enabling more flexible prompt generation; 2) reducing the number of inputs to the text encoder, which decreases GPU memory consumption significantly. Specifically, we found that compound text supervisions, i.e., category-wise and content-wise, is highly effective, since they provide inter-class separability and capture intra-class variations, respectively. Moreover, we condition the prompt generation on visual features through a module called Bonder, which facilitates the alignment between prompts and visual features. Extensive experiments on few-shot recognition and domain generalization demonstrate that TGP-T achieves superior performance with consistently lower training costs. It reduces GPU memory usage by 93% and attains a 2.5% performance gain on 16-shot ImageNet. The code is available at https://github.com/EricTan7/TGP-T.

Introduction

Large-scale vision-language pre-training (Kim, Son, and Kim 2021; Radford et al. 2021; Jia et al. 2021; Bao et al. 2022) has emerged as a powerful paradigm for tackling a wide range of visual tasks (Gu et al. 2021; Saharia et al. 2022; Alayrac et al. 2022). The vision-language models (VLMs), e.g., CLIP (Radford et al. 2021) and ALIGN (Jia et al. 2021), have demonstrated remarkable generalization capabilities to various downstream tasks (Yao et al. 2021; Guo et al. 2023; Huang et al. 2023; Smith et al. 2023).

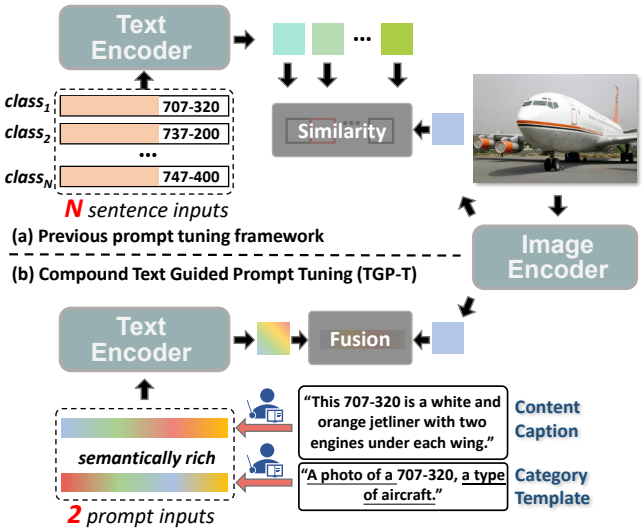

Figure 1: Paradigm Comparison. (a) Prior works parallelize N learnable sentence inputs to text encoder and concatenate category names to each input. (b) TGP-T introduces text supervision to the optimization of prompts, which releases the reliance on category names and reduces the number of prompt inputs to two. This decreases GPU memory consumption significantly. By employing two levels of text supervision, TGP-T performs strong adaptation ability.

Among them, CLIP utilized the contrastive paradigm to align two modalities with 400 million image-text pairs. One of its significant advancements is the ability to achieve open-vocabulary recognition by calculating the similarity between the query image feature and hand-craft text prompts (e.g., "a photo of a <class>.") without additional training.

Fine-tuning a foundation model can be computationally expensive in the era of foundation models. Consequently, there has been a shift in focus from adapting the model to specific tasks (i.e., fine-tuning) to fitting the downstream tasks with the foundation model (i.e., prompting) (Liu et al. 2023b). The milestone work CoOp (Zhou et al. 2022b) is the first time to apply prompt tuning to the VLMs, introducing a series of learnable prompts for textual input instead of taking hand-crafted templates. The follow-up Co-

*These authors contributed equally.
†Corresponding author

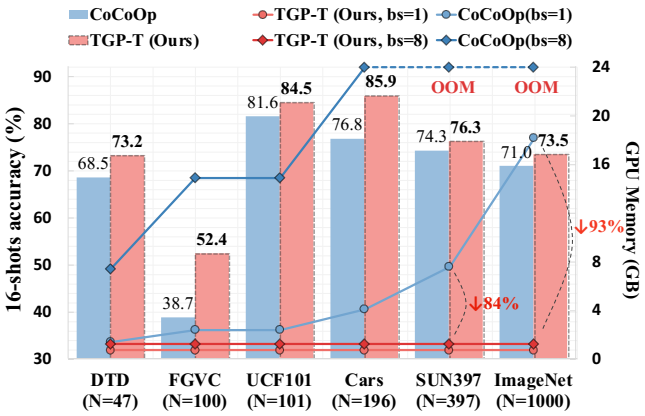

Figure 2: Comparison on Performance (%) and GPU Memory Consumption (GB). N is the number of categories and "bs" donates batch size. TGP-T reduces GPU memory usage across all datasets while achieving superior performance. Note that when using batch size of 8, CoCoOp runs into out-of-memory (OOM) problems on StanfordCars, SUN397, and ImageNet with Nvidia RTX 3090.

CoOp (Zhou et al. 2022a) uses visual features to construct adaptive prompts and further improve the generalization abilities. Regarding data-scarce scenarios, e.g., few-shot recognition (Wan, Guo, and Li 2015) and long-tailed learning (Li et al. 2022), prompt tuning (Zhou et al. 2022b; Dong et al. 2022) has shown considerable improvements compared with those manually curated text inputs.dawdadadada

Despite notable progress, existing methods still suffer from certain shortcomings. The series of methods (Zhou et al. 2022b,a; Zhu et al. 2022; Khattak et al. 2023; Yao, Zhang, and Xu 2023) based on CoOp simultaneously feed N (the number of categories) learnable sentence inputs to text encoder. This is equivalent to learning N category centers before the text encoder, which requires preserving all intermediate activations of the text encoder for gradient back-propagation. As a result, this approach leads to a rapid increase in GPU memory consumption as the N grows, as shown by the blue curve in Fig. 2. Such a training process deviates from the original intention of efficient prompt tuning. Based on the observation above, we propose to utilize a projector to *relocate the learning of N category centers after the text encoder.* Accordingly, we only require two instead of N prompts as the input to the text encoder. As shown in Fig. 2, these designs significantly reduce GPU memory consumption while achieving better performance.

Furthermore, prior works encounter another significant challenge. The prompt tokens are prepended to each <class> token. Consequently, the classification weights, i.e., the textual features, are dependent on a pre-defined category name set, leading to subpar performance when dealing with ambiguous category names, e.g., "707-320" in the FGVCAircraft dataset (Maji et al. 2013). In this case, CoCoOp only achieves 38.7% accuracy on 16-shot FGVCAircraft. In contrast, we *avoid including any category name within prompts*. As shown in Fig. 2, our method obtains a 13.7% improvement on FGVCAircraft, exhibiting superior potential when addressing ambiguous category names.

To sum up, we propose a novel approach called Compound **T**ext-**G**uided **P**rompt **T**uning (TGP-T) that releases the huge resource demand while achieving state-of-the-art performance. As illustrated in Fig. 1: 1) we avoid including category names within prompts during inference, which allows for a more flexible prompt generation. 2) Instead of parallelizing ***N* prompt inputs** for each image, we only require **two prompt inputs** to text encoder. This reduces GPU memory consumption by 93% while achieving a 2.5% increase in accuracy on the 16-shot ImageNet, as shown in Fig. 2. Moreover, the GPU memory consumption of our framework is almost unaffected by the number of categories, which makes it accessible to tune VLMs on datasets of any scale.

However, a crucial problem is to ensure the prompts carry sufficient information that is closely related to the current sample, without directly including category names. Specifically, we suppose that *the optimization of prompts should not be unconstrained*. Therefore, we introduce two "teachers" to guide the process as shown in Fig. 1. 1) *Category-wise Text Supervision*, which offers a high-level understanding of the target category. 2) *Content-wise Text Supervision*, which captures the intra-class variations. The two "teachers" provide general category cues and specific content details, helping the model adapt to varying degrees of sample diversity and complexity. Moreover, to incorporate fine-grained visual cues, we condition the prompt generation on the visual features (Zhou et al. 2022a; Rao et al. 2022) through a well-designed structure called Bonder. During inference, we simply input the image into the network, and the Bonder generates suitable prompts via image-adaptive cues. Then the prompts are fed into the text encoder to get enriched textual features. Finally, the visual features and textual features are concatenated together, and then a projector is applied to obtain the predicted probability of the input image. Our main contributions are summarized as follows:

- We propose a novel framework called Compound **T**ext-**G**uided **P**rompt **T**uning (TGP-T), which significantly reduces training costs while achieving state-of-the-art performance on 11 datasets for few-shot classification.
- We offer an alternative perspective for prompt tuning, i.e., using text supervision to guide the optimization of prompts. This enables two benefits: 1) releasing the model reliance on category names during inference, and thereby enabling more flexible prompt generation; 2) reducing the number of inputs to the text encoder, which decreases GPU memory consumption significantly.
- Through empirical study, we found that compound text supervisions, i.e., *category-wise* and *content-wise*, are highly effective. Since they provide inter-class separability and capture intra-class variations, respectively.
- We propose to use a lightweight structure called Bonder to bridge the visual and language modalities. By interacting prompt queries with image features, the Bonder facilitates the generated prompts to be closely aligned with the current image features, which allows a better harness of VLM.

Related Works

Vision-Language Models. There has been a growing interest in vision-language models since CLIP (Radford et al. 2021) was proposed. With a contrastive-based pretraining approach, CLIP has achieved impressive progress in visual representation learning by utilizing large-scale image-text pairs. Many fields have benefited from CLIP, such as object detection (Gu et al. 2021), image generation (Saharia et al. 2022), and visual question answering (Alayrac et al. 2022). With the rapid development of large language models (LLM) (Brown et al. 2020; Touvron et al. 2023) in recent times, it has become possible to directly apply separately pre-trained LLM and vision foundation models to build VLM, which is capable of understanding both images and text by adding a small number of connection parameters for training. For instance, BLIP-2 (Li et al. 2023) achieves this by training the additional Q-former and linear mapping. Similarly, MiniGPT-4 (Zhu et al. 2023) and LLaVA (Liu et al. 2023a) achieve impressive multimodal chat abilities with only additional training of linear projection for aligning the two modalities.

Benefiting from the advancements of these VLMs, we further explored how to utilize CLIP to improve visual recognition tasks. Inspired by (Li et al. 2023), we propose to use a module to bridge the vision and language modalities, which facilitates the learnable prompts to be closely aligned with the visual features. In addition, by leveraging these VLMs to generate text descriptions for images, we inject rich cross-modal knowledge into the vision tasks.

Prompt Tuning in Computer Vision. Prompt Learning was initially introduced in the field of Natural Language Processing (NLP). In GPT-2 (Radford et al. 2019), the pre-trained language model can complete specific downstream tasks without fine-tuning by adding some prefix descriptions, i.e., prompts, before the input sequence. Some works (Schick and Schütze 2020; Shin et al. 2020) also make prompts learnable to better adapt to downstream tasks. In general, the existing methods of prompt tuning in computer vision can be roughly categorized into two clusters: improving the discrimination abilities and enhancing the generalization abilities. For the discrimination abilities, CoOp (Zhou et al. 2022b) first introduces prompt tuning of the CLIP into few-shot prediction while further works improve the performance on various tasks, including fine-grained object retrieval (Wang et al. 2023), multi-label recognition (Guo et al. 2023) and long-tailed classification (Dong et al. 2022). For the generalization abilities, CoCoOp (Zhou et al. 2022a) adapts to new target domains by making the prompt relevant to the input image while ProGrad (Zhu et al. 2022) achieves it by gradient correction. More recently, KgCoOp (Yao, Zhang, and Xu 2023) propose to enhance generalization by constraining the learned prompt embeddings with general knowledge, and MaPLe (Khattak et al. 2023) achieve it by learning coupled prompts in both visual and text modalities. Aside from this, some works (Zhang et al. 2021, 2023) turn to retrieval in a knowledge base to achieve better performance, while CaFo (Zhang et al. 2023) actually adopt $(K + K')$-shot for a K-shot problem.

However, those prompt tuning based methods need to utilize all category names to learn category centers before the text encoder. When the number of categories is large, this results in a significantly larger text batch, leading to substantial resource consumption. In contrast, we explore a text-guided prompt tuning paradigm that relocates the learning of category centers after the text encoder, which only requires a small text batch.

Method

Our method is based on a pre-trained vision-language model, as shown in Fig. 3, and by adding a small number of learnable parameters, it allows for the cost-effective transfer of visual classification tasks. In this section, we first give a brief review of CLIP (Radford et al. 2021). Then, we give a detailed introduction to the proposed TGP-T.

Preliminary

CLIP is an effective method to learn visual representation from natural language supervision. Specifically, suppose the training set contains M samples and is denoted as $\mathcal{S} = \{I_i, T_i\}_{i=0}^{M-1}$, where $I_i \in \mathbb{R}^{H\times W\times 3}$ is the image and T_i is the textual description corresponding to the image I_i. During training, the visual encoder $\mathcal{V}(\cdot)$ encodes the I_i into visual feature: $\boldsymbol{v_i} = \mathcal{V}(I_i), \boldsymbol{v_i} \in \mathbb{R}^d$, where d is the hidden dimension of CLIP. The textual encoder $\mathcal{T}(\cdot)$ encodes the T_i into textual feature: $\boldsymbol{t_i} = \mathcal{T}(T_i), \boldsymbol{t_i} \in \mathbb{R}^d$. Matched image and text feature pairs are regarded as positive pairs, i.e., $\{\boldsymbol{v_i}, \boldsymbol{t_i}\}$. Correspondingly, unmatched pairs are regarded as negative pairs, i.e., $\{\boldsymbol{v_i}, \boldsymbol{t_j} | i \neq j\}$. Given a batch of image-text pairs, CLIP maximizes the cosine similarity of positive pairs and minimizes it among negative pairs. After such pre-training, CLIP can learn a good visual representation and be transferred to various downstream tasks.

Taking the classification task as an example, CLIP can accomplish zero-shot classification by reasonably constructing the text input. CLIP utilizes a hand-crafted prompt to form the text input $T_i' = \{$"A photo of a $[\text{CLASS}_i]$."$\}, i = 0, 1, ..., N-1$, where $[\text{CLASS}_i]$ is the specific category name, such as "dog", "cat", etc., and N is the number of categories. The image I that needs to be predicted is fed into the image encoder to obtain the corresponding image feature: $\boldsymbol{v} = \mathcal{V}(I)$. Then all T_i' is fed into the textual encoder in parallel to yield a set of textual features $\{\boldsymbol{t_i'} | \boldsymbol{t_i'} = \mathcal{T}(T_i')\}_{i=0}^{N-1}$. Based on the visual feature and textual feature, the probability is computed for CLIP's prediction:

$$p(y = i|I) = \frac{e^{sim(\boldsymbol{t_i'}, \boldsymbol{v})/\tau}}{\sum_{j=0}^{N-1} e^{sim(\boldsymbol{t_j'}, \boldsymbol{v})/\tau}}, \tag{1}$$

where $sim(\cdot)$ denotes the similarity calculation and τ is a temperature parameter. In our method, we directly load a pre-trained CLIP model.

TGP-T: Compound Text-Guided Prompt Tuning

Prompt Generation. We condition the prompt generation on visual features $\boldsymbol{v}$ through a structure called Bonder,

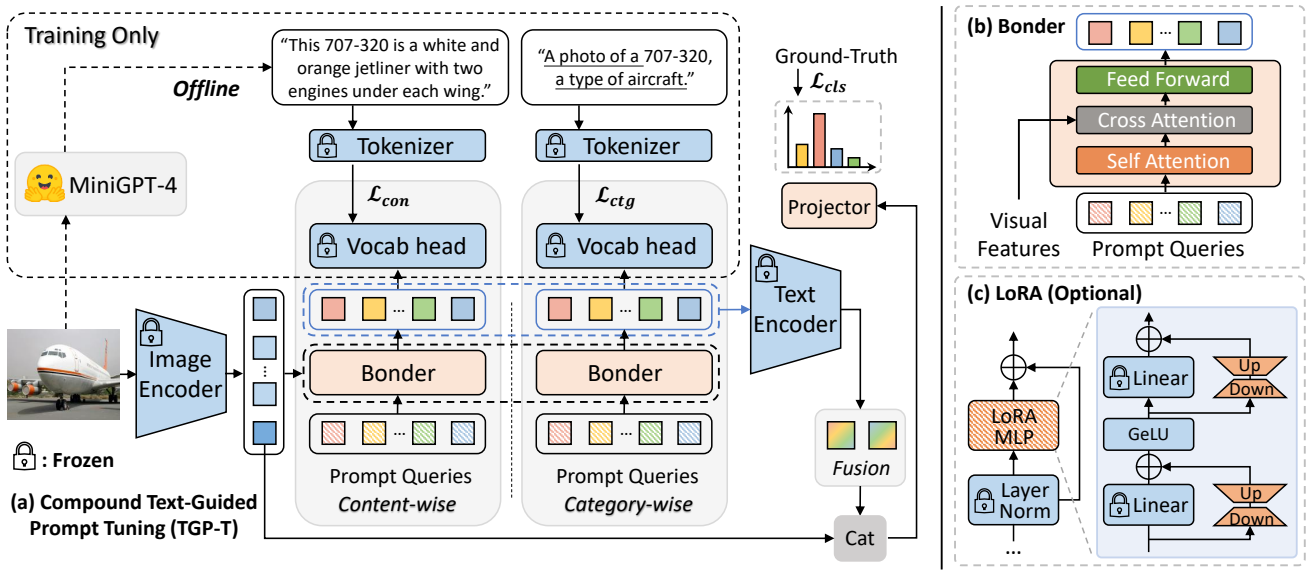

Figure 3: (a) Overview of the proposed TGP-T framework. 1) Prompt Generation: the learnable prompts are conditioned on visual features through a Bonder structure. 2) Text Supervision: we introduce content-wise and category-wise text supervision to guide the optimization of prompts during training. 3) Feature Fusion: two text features are fused to yield the final text feature. The visual and text features are then concatenated and projected to perform recognition. Dashed lines denote that text descriptions are generated offline. (b) The detailed structure of Bonder. (c) TGP-T can further benefit from LoRA (Hu et al. 2021), where we identified tuning linear layers in MLP as an effective pattern.

which is implemented with transformer layers, as shown in Fig. 3. Formally, We randomly initialize K learnable prompt queries $\boldsymbol{Q} = \{\boldsymbol{q_1}, \boldsymbol{q_2}, ..., \boldsymbol{q_K}\} \in \mathbb{R}^{K\times d}$. Then these queries are fed into Bonder, which can be formulated as follows:

$$\begin{aligned} \boldsymbol{Q_S} &= \boldsymbol{Q} + \text{Self-Attn}(\text{LN}(\boldsymbol{Q})), \\ \boldsymbol{Q_C} &= \boldsymbol{Q_S} + \text{Cross-Attn}(\text{LN}(\boldsymbol{Q_S}),\ \text{LN}(\boldsymbol{v})), \\ \boldsymbol{P} &= \boldsymbol{Q_C} + \text{FFN}(\text{LN}(\boldsymbol{Q_C})), \end{aligned} \tag{2}$$

where $\boldsymbol{P} = \{\boldsymbol{p_1}, \boldsymbol{p_2}, ..., \boldsymbol{p_K}\} \in \mathbb{R}^{K\times d}$ is the generated prompts, Self-Attn$(\cdot)$ and Cross-Attn$(\cdot)$ denote the self-attention and cross-attention operation, respectively. LN$(\cdot)$ is the Layer Normalization and FFN$(\cdot)$ is a two-layer fully connected network. Instead of concatenating category name tokens to the prompts, we directly feed $\boldsymbol{P}$ into the textual encoder. The learnable $\boldsymbol{Q}$ will be updated during training through gradient backpropagation. The interactions through Bonder insert plentiful visual cues and allow the prompts to suit the current image better.

Text Supervision. The vanilla prompts generated above are short of textual semantics and lack specific category cues due to the absence of category names. Therefore, we introduce two "teachers" to guide the optimization of prompts from two distinct levels, i.e., content-wise and category-wise. Accordingly, we construct two branches to learn two prompt inputs $\boldsymbol{P_{con}}$ and $\boldsymbol{P_{ctg}}$, respectively.

As for content-wise text supervision, we take advantage of the prevailing VLMs to generate descriptions based on the specific content of the image. Specifically, we employ MiniGPT-4 (Zhu et al. 2023), which is powered by Vicuna-7B (Chiang et al. 2023) and is resource-efficient to generate descriptions: D_{con} = MiniGPT-4(I). We adopt the question "*Describe the* {*class*} *in this image in one sentence*". D_{con} is generated offline before training, which ensures that no additional computational overhead is introduced during the training process.

As for the category-wise text supervision, we adopt the hand-engineered templates selected by Tip-Adapter (Zhang et al. 2021), which provides overall descriptions for the categories. Take the FGVCAircraft dataset (Maji et al. 2013) as an example, the text description is D_{ctg} = {"A photo of a [CLASS$_i$], a type of aircraft."}$_{i=0}^{N-1}$. For more details of $\{D_{con}, D_{ctg}\}$ please refer to Appendix.

During the training process, we employ D_{con} and D_{ctg} to guide the optimization of $\boldsymbol{P_{con}}$ and $\boldsymbol{P_{ctg}}$, respectively. Inspired by (Devlin et al. 2018), we calculate the loss in the vocabulary space $\mathbb{R}^{d_V}$, where d_V denotes the vocabulary size of the pre-trained model. Specifically, we use the tokenizer of the pre-trained text encoder to map $\{D_{con}, D_{ctg}\}$ into the vocabulary space. Then we project the generated prompts $\{\boldsymbol{P_{con}}, \boldsymbol{P_{ctg}}\}$ into the vocabulary space using the transposed weights of the pre-trained embedding layer, which is frozen and denoted as $\boldsymbol{W_E^T} \in \mathbb{R}^{d\times d_V}$. The text supervision loss is then measured as follows:

$$\begin{aligned} \{\mathcal{L}_{con}, \mathcal{L}_{ctg}\} = \{&\text{CE}(\boldsymbol{P_{con}}\boldsymbol{W_E^T}, \text{Tokenizer}(D_{con})), \\ &\text{CE}(\boldsymbol{P_{ctg}}\boldsymbol{W_E^T}, \text{Tokenizer}(D_{ctg}))\}, \end{aligned} \tag{3}$$

where $\mathcal{L}_{con}$ and $\mathcal{L}_{ctg}$ denote the loss for the content-wise and category-wise text supervision, respectively. CE$(\cdot)$ denotes the Cross-Entropy Loss.

Cross-modal Feature Fusion. The $\{\boldsymbol{P_{con}}, \boldsymbol{P_{ctg}}\}$ generated through Bonder is then fed into text encoder to produce enriched features, which can be formulated as follows:

$$\{\boldsymbol{t_{con}}, \boldsymbol{t_{ctg}}\} = \{\mathcal{T}(\boldsymbol{P_{con}}), \mathcal{T}(\boldsymbol{P_{ctg}})\}, \tag{4}$$

where $\boldsymbol{t_{con}} \in \mathbb{R}^d, \boldsymbol{t_{ctg}} \in \mathbb{R}^d$ denote the textual features for content-wise and category-wise branches, respectively.

Subsequently, the text modality features and visual modality features are concatenated together and passed through a projector $\mathcal{F}(\cdot)$ to get the final predictions. The classification loss is computed as follows:

$$\mathcal{L}_{cls} = \text{CE}(\mathcal{F}([\boldsymbol{v}, (\boldsymbol{t_{con}} + \boldsymbol{t_{ctg}})/2]), y), \tag{5}$$

where $[\cdot]$ donates the concatenated operation along the feature dimension, and $\text{CE}(\cdot)$ donates the Cross-Entropy loss. $\mathcal{F}(\cdot)$ consists of a single linear layer followed by a softmax function, and y is the corresponding label to the image I. The fusion between text features follows a general practice of additive averaging.

Model Training and Inference. During the training process, both the image encoder and text encoder are frozen, and only the Bonder and projector are learnable. The overall loss can be formulated as follows:

$$\mathcal{L} = \mathcal{L}_{cls} + \mathcal{L}_{con} + \mathcal{L}_{ctg}. \tag{6}$$

In addition to directly freezing the weights of the pre-trained model, we also explored efficient fine-tuning methods such as LoRA (Hu et al. 2021), which is currently widely used for fine-tuning large language models. Through our experiments, we have identified a reasonable way to apply it to visual tasks (i.e., tuning only linear layers in the MLP), ensuring that our method can also benefit from this fine-tuning approach. For specific details, please refer to the Appendix.

During the inference stage, our method does not require any additional textual information, which is different from previous manual template-based methods and prompt tuning based frameworks. We directly input the image into the network, and Bonder will adaptively generate suitable prompts to feed into the text encoder. Ultimately, the visual features and textual features are concatenated and projected to obtain the prediction results.

Experiment

Experimental Settings

Datasets. Following CLIP (Radford et al. 2021), we adopt 11 publicly available image classification datasets that cover diverse scenes and scales, including ImageNet (Deng et al. 2009), Caltech (Fei-Fei, Fergus, and Perona 2004), OxfordPets (Parkhi et al. 2012), Flowers (Nilsback and Zisserman 2008), Food101 (Bossard, Guillaumin, and Van Gool 2014), StanfordCars (Krause et al. 2013), FGVCAircraft (Maji et al. 2013), EuroSAT (Helber et al. 2019), UCF101 (Soomro, Zamir, and Shah 2012), DTD (Cimpoi et al. 2014), and SUN397 (Xiao et al. 2010). We follow the few-shot evaluation protocol in CoOp (Zhou et al. 2022b), i.e., we use 1, 2, 4, 8, and 16 shots for training, respectively, and report results on the full test sets.

Method	Ref.	Number of Shots				
		1	2	4	8	16
Linear Probe	–	43.87	56.84	67.12	73.77	78.16
CoOp	IJCV'22	68.39	71.50	74.45	77.03	79.86
CoCoOp	CVPR'22	69.10	70.38	72.32	76.20	78.43
Tip-Adapter	ECCV'22	69.81	71.56	74.18	75.17	77.39
Tip-Adapter-F	ECCV'22	<u>70.86</u>	73.10	76.04	78.81	81.27
MaPLe	CVPR'23	69.93	72.54	76.37	79.02	81.79
Cross-Modal	CVPR'23	70.75	73.29	76.79	79.05	80.75
TGP-T	–	70.51	<u>74.08</u>	<u>77.13</u>	<u>79.34</u>	<u>82.65</u>
TGP-T-F	–	**72.15**	**75.22**	**78.20**	**80.69**	**84.06**

Table 1: Comparison (%) to SOTA using the CoOp protocol, which reports averaged top-1 accuracy across 11 test sets with ViT-B/16. "Linear Probe" denotes the Linear-Probe CLIP. The best results are bolded, and the second best results are underlined.

Implementation Details. We set ViT-B/16 as the image encoder. The depth of the Bonder is set to 1. The number of category-wise and content-wise prompt queries is 32 and 64, respectively. We adopt the AdamW optimizer (Loshchilov and Hutter 2017) with a learning rate of 5e-5 and a weight decay of 1e-4. The model is trained for 12,800 iterations with a batch size of 8. We tune the hyperparameters on a few-shot validation set with $min(n, 4)$ shots (n is the number of training shots) rather than searching on the test set.

Performance

To evaluate the superiority of our novel framework, we compare with prior arts including CoOp (Zhou et al. 2022b), CoCoOp (Zhou et al. 2022a), Cross-Modal Adaptation (Lin et al. 2023), MaPLe (Khattak et al. 2023), Tip-Adapter (Zhang et al. 2021) with its fine-tuning version Tip-Adapter-F, and Linear-Probe CLIP (Radford et al. 2021). We reproduce all previous methods using the same randomly sampled few-shot images for a fair comparison.

Comparisons to Prior Arts. As reported in Tab. 1, TGP-T surpasses previous methods from 2 shots to 16 shots, demonstrating its superior performance. Remarkably, TGP-T with 2 shots outperforms the 4-shot CoCoOp and Linear-Probe CLIP. TGP-T with 8 shots performs better than 16-shot CoCoOp, Tip-Adapter, and Linear-Probe CLIP, which underscores its effectiveness in learning from limited data. Compared with recent work Cross-Modal, TGP-T achieves a 1.90% increase in 16-shot settings. Compared with CoCoOp, which conditions the prompt generation on visual features, TGP-T obtains an average gain from 78.43% to 82.65%. Compared with Tip-Adapter, which constructs a knowledge base for categorization, TGP-T attains an average gain from 77.39% to 82.65%. These results further demonstrate the superiority of our proposed framework. Moreover, with LoRA (Hu et al. 2021), which fine-tunes a small number of parameters (less than 0.1% of model parameters), TGP-T-F sets the new state-of-the-art performance across all shot settings, exceeding previous methods with a decent margin.

Method	Source	Target	
	ImageNet	-V2	-Sketch
Zero-Shot CLIP	66.7	60.8	46.2
Linear Probe	65.9	56.3	34.8
CoOp	71.7	64.6	47.9
CoCoOp	71.0	64.1	48.8
Cross-Modal	72.8	64.8	47.9
MaPLe	71.9	64.1	**49.2**
TGP-T	**73.5**	**65.1**	48.7

Table 2: Comparison (%) on Distribution Shift. We train the model on "Source" dataset and test on "Target" datasets with ViT-B/16.

Method	RN50	B/32	B/16	L/14	Mem.	Time
Zero-Shot CLIP	59.60	63.20	68.60	75.30	-	-
CoOp	62.95	66.85	71.60	*OOM*	18	12hr
CoCoOp	63.30	66.20	71.30	*OOM*	18	13hr
Tip-Adapter	62.03	65.48	70.19	77.06	5	-
MaPLe	-	66.80	71.90	*OOM*	20	2hr
TGP-T	**65.19**	**68.15**	**73.48**	**79.07**	1	12min

Table 3: Comparison (%) of Different Visual Backbones. For ViT models, we take suffixes such as "B/32" as their names for simplicity. We also report the GPU Memory consumption (GB) and the Training Time of the ViT-B/16. All results are conducted on the 16-shot ImageNet.

Distribution Shift. We further assess the robustness of TGP-T under out-of-distribution (OOD) conditions. Specifically, we train on the ImageNet (Deng et al. 2009) and test on ImageNet-V2 (Recht et al. 2019) and ImageNet-Sketch (Wang et al. 2019). As shown in Tab. 2, when achieving the best result on the source dataset, TGP-T consistently attains promising performance on both OOD datasets. It achieves the highest accuracy of 65.1% on ImageNet-V2 and a competitive 48.7% on the more challenging ImageNet-Sketch. The results demonstrate the superior OOD performance of TGP-T. It effectively generalizes from ImageNet to the out-of-distribution datasets, showcasing its potential in handling distribution shifts.

Different Visual Backbones. We implement TGP-T with various visual encoders. Moreover, we report the GPU memory usage and training time of the ViT-B/16 backbone. As shown in Tab. 3, TGP-T consistently achieves leading performance with different backbones, indicating our generalizability to network architectures. As for training costs, both CoOp, CoCoOp, and MaPLe are memory and time-intensive, consuming over 18GB of GPU memory and taking a long time to train. In contrast, TGP-T consumes only 1GB of GPU memory. Moreover, TGP-T enables the utilization of more powerful backbones such as ViT-L/14, while CoOp, CoCoOp, and MaPLe run into out-of-memory (OOM) problems on Nvidia RTX 3090. In addition, TGP-T trains in a short period, demonstrating better efficiency.

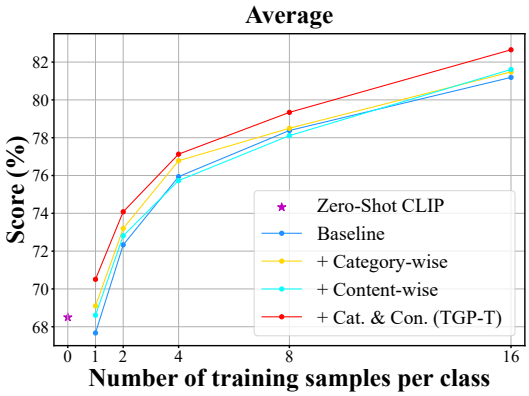

Figure 4: Ablation Study of the Granularity of Text Supervision. Results are averaged across 11 datasets. "Baseline" denotes the model trained without text supervision.

Cat.	Con.	4-Shot		16-Shot	
		ImageNet	Aircraft	ImageNet	Aircraft
		70.18	32.10	72.82	42.63
✓		70.06	34.53	73.02	47.49
	✓	70.41	33.60	73.09	47.40
✓	✓	**70.58** (↑0.40)	**36.60** (↑4.50)	**73.48** (↑0.66)	**52.39** (↑9.76)

Table 4: Ablation Study (%) of the Granularity of Text Supervision. "Cat." and "Con." denote Category-wise and Content-wise text supervision, respectively.

Ablation Studies

We conduct analysis across all 11 datasets for the ablations on text supervision. For other ablations, we evaluate on the most representative ImageNet.

Granularity of Text Supervision. We investigate the influence of different text supervision. As shown in Fig. 4, the model trained without text supervision is denoted as "Baseline". In general, concurrently employing both types of text supervision results in significant improvements. Digging deeper, when the number of shots is small, using category-wise supervision alone yields comparable performance. As the number of shots increases, only combining both types of text supervision achieves a prominent lead. This is because when there is only 1 shot, category-wise description is sufficient to provide overall information. As the number of shots increases, intra-class diversity starts to emerge. The standalone edition of category-wise and content-wise supervision can not significantly improve the performance, since each provides only a partial perspective. *Instead, combining both types of supervision helps the model understand the general category information while capturing intra-class variations, which leads to a noticeable improvement in categorization.* In Tab. 4, we further discuss results on the most representative ImageNet, and FGVCAircraft, where the pre-defined category names such

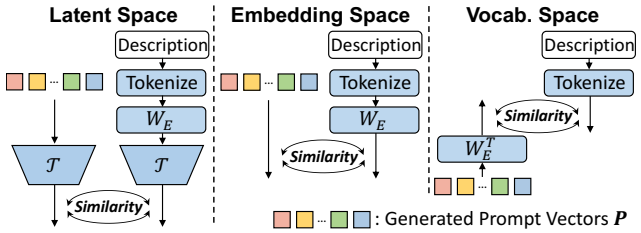

Figure 5: Illustration of Different Spaces of Text Supervision.

Supervision Space	Number of Shots				
	1	2	4	8	16
Embedding Space	68.59	72.50	76.07	77.66	81.46
Latent Space	67.80	72.39	76.23	78.36	81.58
Vocab. Space (TGP-T)	**70.51**	**74.08**	**77.13**	**79.34**	**82.65**

Table 5: Ablation Study (%) of the Supervision Space. Results are averaged across 11 datasets.

as "707-320" are ambiguous for the model. Combining both types of text supervision leads to 4.50% and 9.76% gains on FGVCAircraft with 4 and 16 shots, respectively, which verifies the effectiveness of compound text supervision.

Different Spaces of Text Supervision. As illustrated in Fig. 5, there are several spaces to employ text supervision. The Embedding Space is after the embedding layer, and the Latent Space is after the text encoder. The Vocabulary Space refers to the discrete representation space of words (Devlin et al. 2018). As shown in Tab. 5, applying supervision in the vocabulary space leads to better performance across all shot settings. In fact, discrete vocabulary space is inherently more structured than continuous feature space. *Guiding the optimization of prompts in discrete space reduces ambiguity and simplifies the learning process*, which further makes the task of distinguishing between different categories more straightforward. Therefore, we suggest that incorporating discrete vocabulary space to guide prompt learning is more effective.

Different Structures of Bonder. There are several choices for the Bonder structure. We evaluate the Self-Attention module, Cross-Attention module, and Meta-Net. The Meta-Net is adopted in CoCoOp (Zhou et al. 2022a), which denotes a two-layer bottleneck structure (Linear-ReLU-Linear). As shown in Tab. 6, the Cross-Attention consistently outperforms the others. This is attributed to its ability to aggregate information from different sources, enabling an effective incorporation of visual cues.

Further Analysis

Generalization to different Text Encoders. We assess the generalizability of TGP-T to different text encoders. As shown in Tab. 7, the CLIP's text encoder achieves the best performance with 1 shot. When the training shots increase, the FLAN-T5$_{\text{Base}}$ delivers superior performance. Above all,

Bonder Design	Number of Shots				
	1	2	4	8	16
Meta-Net	67.34	69.78	69.74	70.59	72.67
Self-Attention	68.82	69.60	69.72	71.28	72.48
Cross-Attention (TGP-T)	**69.32**	**70.12**	**70.58**	**72.07**	**73.48**

Table 6: Different Structures of Bonder. Results are reported on the most representative ImageNet dataset.

Method	Text Encoder	Number of Shots				
		1	2	4	8	16
TGP-T	BERT$_{\text{Base}}$	68.99	**70.13**	70.80	72.15	73.55
TGP-T	FLAN-T5$_{\text{Base}}$	68.45	69.64	**70.84**	**72.30**	**73.67**
TGP-T	CLIP-Text	**69.32**	70.12	70.58	72.07	73.48

Table 7: Generalization to different Text Encoders. Results are reported on the most representative ImageNet.

Bonder Depth	Number of Shots				
	1	2	4	8	16
$\times 1$	**69.32**	**70.12**	70.58	72.07	**73.48**
$\times 2$	69.30	69.76	70.66	**72.23**	73.29
$\times 4$	69.12	70.04	**70.73**	71.71	73.23
$\times 8$	69.13	69.61	69.96	71.46	72.91

Table 8: Effects of Bonder Depth. Results are reported on the most representative ImageNet dataset.

different text encoders achieve competitive results, while slightly scaling up to FLAN-T5$_{\text{Base}}$, our method achieves an improvement accordingly, demonstrating the adaptability of the TGP-T to different text encoders.

Effects of Bonder Depth. We evaluate the influence of the depth of Bonder. As shown in Tab. 8, a depth of 1 yields the most robust results across 1, 2, and 16 shots, while a deeper bonder brings improvements with 4 and 8 shots. Interestingly, a depth of 8 underperforms across all shot settings, suggesting that an overly deep bonder leads to overfitting or optimization difficulties. Therefore, we suggest that a depth of 1 is a better trade-off between performance and efficiency.

Conclusions

In this work, we propose TGP-T, an efficient prompt tuning framework for adapting VLMs with significantly lower resource demand. We introduce compound text supervision to guide the optimization of prompts. Through a Bonder structure, we align the generated prompts with visual features. As a result, we only need two prompt inputs to text encoder to produce state-of-the-art performance on 11 datasets for few-shot classification. Future works could explore more diverse forms and task-adaptive text supervision to further improve the effectiveness of text supervision in prompt tuning.

Acknowledgments

This work was supported by the National Key Research and Development Plan under Grant 2021YFE0205700, Beijing Natural Science Foundation JQ23016, the External cooperation key project of Chinese Academy Sciences 173211KYSB20200002, the Science and Technology Development Fund of Macau Project 0123/2022/A3, and 0070/2020/AMJ, Open Research Projects of Zhejiang Lab No. 2021KH0AB07, CCF-Zhipu AI Large Model OF 202219 and InnoHK program.

References

Alayrac, J.-B.; Donahue, J.; Luc, P.; Miech, A.; Barr, I.; Hasson, Y.; Lenc, K.; Mensch, A.; Millican, K.; Reynolds, M.; et al. 2022. Flamingo: a visual language model for few-shot learning. *Advances in Neural Information Processing Systems*, 35: 23716–23736.

Bao, H.; Wang, W.; Dong, L.; Liu, Q.; Mohammed, O. K.; Aggarwal, K.; Som, S.; Piao, S.; and Wei, F. 2022. Vlmo: Unified vision-language pre-training with mixture-of-modality-experts. *Advances in Neural Information Processing Systems*, 35: 32897–32912.

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

Brown, T.; Mann, B.; Ryder, N.; Subbiah, M.; Kaplan, J. D.; Dhariwal, P.; Neelakantan, A.; Shyam, P.; Sastry, G.; Askell, A.; et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33: 1877–1901.

Chiang, W.-L.; Li, Z.; Lin, Z.; Sheng, Y.; Wu, Z.; Zhang, H.; Zheng, L.; Zhuang, S.; Zhuang, Y.; Gonzalez, J. E.; Stoica, I.; and Xing, E. P. 2023. Vicuna: An Open-Source Chatbot Impressing GPT-4 with 90%* ChatGPT Quality.

Cimpoi, M.; Maji, S.; Kokkinos, I.; Mohamed, S.; and Vedaldi, A. 2014. Describing textures in the wild. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 3606–3613.

Deng, J.; Dong, W.; Socher, R.; Li, L.-J.; Li, K.; and Fei-Fei, L. 2009. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, 248–255. Ieee.

Devlin, J.; Chang, M.-W.; Lee, K.; and Toutanova, K. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Dong, B.; Zhou, P.; Yan, S.; and Zuo, W. 2022. Lpt: Long-tailed prompt tuning for image classification. *arXiv preprint arXiv:2210.01033*.

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

Gu, X.; Lin, T.-Y.; Kuo, W.; and Cui, Y. 2021. Open-vocabulary object detection via vision and language knowledge distillation. *arXiv preprint arXiv:2104.13921*.

Guo, Z.; Dong, B.; Ji, Z.; Bai, J.; Guo, Y.; and Zuo, W. 2023. Texts as images in prompt tuning for multi-label image recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2808–2817.

Helber, P.; Bischke, B.; Dengel, A.; and Borth, D. 2019. Eurosat: A novel dataset and deep learning benchmark for land use and land cover classification. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 12(7): 2217–2226.

Hu, E. J.; Shen, Y.; Wallis, P.; Allen-Zhu, Z.; Li, Y.; Wang, S.; Wang, L.; and Chen, W. 2021. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*.

Huang, S.; Gong, B.; Pan, Y.; Jiang, J.; Lv, Y.; Li, Y.; and Wang, D. 2023. VoP: Text-Video Co-operative Prompt Tuning for Cross-Modal Retrieval. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 6565–6574.

Jia, C.; Yang, Y.; Xia, Y.; Chen, Y.-T.; Parekh, Z.; Pham, H.; Le, Q.; Sung, Y.-H.; Li, Z.; and Duerig, T. 2021. Scaling up visual and vision-language representation learning with noisy text supervision. In *International conference on machine learning*, 4904–4916. PMLR.

Khattak, M. U.; Rasheed, H.; Maaz, M.; Khan, S.; and Khan, F. S. 2023. Maple: Multi-modal prompt learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 19113–19122.

Kim, W.; Son, B.; and Kim, I. 2021. Vilt: Vision-and-language transformer without convolution or region supervision. In *International Conference on Machine Learning*, 5583–5594. PMLR.

Krause, J.; Stark, M.; Deng, J.; and Fei-Fei, L. 2013. 3d object representations for fine-grained categorization. In *Proceedings of the IEEE international conference on computer vision workshops*, 554–561.

Li, J.; Li, D.; Savarese, S.; and Hoi, S. 2023. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. *arXiv preprint arXiv:2301.12597*.

Li, J.; Tan, Z.; Wan, J.; Lei, Z.; and Guo, G. 2022. Nested collaborative learning for long-tailed visual recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 6949–6958.

Lin, Z.; Yu, S.; Kuang, Z.; Pathak, D.; and Ramanan, D. 2023. Multimodality helps unimodality: Cross-modal few-shot learning with multimodal models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 19325–19337.

Liu, H.; Li, C.; Wu, Q.; and Lee, Y. J. 2023a. Visual instruction tuning. *arXiv preprint arXiv:2304.08485*.

Liu, P.; Yuan, W.; Fu, J.; Jiang, Z.; Hayashi, H.; and Neubig, G. 2023b. Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing. *ACM Computing Surveys*, 55(9): 1–35.

Loshchilov, I.; and Hutter, F. 2017. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*.

Maji, S.; Rahtu, E.; Kannala, J.; Blaschko, M.; and Vedaldi, A. 2013. Fine-grained visual classification of aircraft. *arXiv preprint arXiv:1306.5151*.

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

Parkhi, O. M.; Vedaldi, A.; Zisserman, A.; and Jawahar, C. 2012. Cats and dogs. In *2012 IEEE conference on computer vision and pattern recognition*, 3498–3505. IEEE.

Radford, A.; Kim, J. W.; Hallacy, C.; Ramesh, A.; Goh, G.; Agarwal, S.; Sastry, G.; Askell, A.; Mishkin, P.; Clark, J.; et al. 2021. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, 8748–8763. PMLR.

Radford, A.; Wu, J.; Child, R.; Luan, D.; Amodei, D.; Sutskever, I.; et al. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8): 9.

Rao, Y.; Zhao, W.; Chen, G.; Tang, Y.; Zhu, Z.; Huang, G.; Zhou, J.; and Lu, J. 2022. Denseclip: Language-guided dense prediction with context-aware prompting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 18082–18091.

Recht, B.; Roelofs, R.; Schmidt, L.; and Shankar, V. 2019. Do imagenet classifiers generalize to imagenet? In *International conference on machine learning*, 5389–5400. PMLR.

Saharia, C.; Chan, W.; Saxena, S.; Li, L.; Whang, J.; Denton, E. L.; Ghasemipour, K.; Gontijo Lopes, R.; Karagol Ayan, B.; Salimans, T.; et al. 2022. Photorealistic text-to-image diffusion models with deep language understanding. *Advances in Neural Information Processing Systems*, 35: 36479–36494.

Schick, T.; and Schütze, H. 2020. Exploiting cloze questions for few shot text classification and natural language inference. *arXiv preprint arXiv:2001.07676*.

Shin, T.; Razeghi, Y.; Logan IV, R. L.; Wallace, E.; and Singh, S. 2020. Autoprompt: Eliciting knowledge from language models with automatically generated prompts. *arXiv preprint arXiv:2010.15980*.

Smith, J. S.; Karlinsky, L.; Gutta, V.; Cascante-Bonilla, P.; Kim, D.; Arbelle, A.; Panda, R.; Feris, R.; and Kira, Z. 2023. CODA-Prompt: COntinual Decomposed Attention-based Prompting for Rehearsal-Free Continual Learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 11909–11919.

Soomro, K.; Zamir, A. R.; and Shah, M. 2012. UCF101: A dataset of 101 human actions classes from videos in the wild. *arXiv preprint arXiv:1212.0402*.

Touvron, H.; Lavril, T.; Izacard, G.; Martinet, X.; Lachaux, M.-A.; Lacroix, T.; Rozière, B.; Goyal, N.; Hambro, E.; Azhar, F.; et al. 2023. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*.

Wan, J.; Guo, G.; and Li, S. Z. 2015. Explore efficient local features from RGB-D data for one-shot learning gesture recognition. *IEEE transactions on pattern analysis and machine intelligence*, 38(8): 1626–1639.

Wang, H.; Ge, S.; Lipton, Z.; and Xing, E. P. 2019. Learning robust global representations by penalizing local predictive power. *Advances in Neural Information Processing Systems*, 32.

Wang, S.; Chang, J.; Wang, Z.; Li, H.; Ouyang, W.; and Tian, Q. 2023. Fine-grained retrieval prompt tuning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, 2644–2652.

Xiao, J.; Hays, J.; Ehinger, K. A.; Oliva, A.; and Torralba, A. 2010. Sun database: Large-scale scene recognition from abbey to zoo. In *2010 IEEE computer society conference on computer vision and pattern recognition*, 3485–3492. IEEE.

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

Yao, Y.; Zhang, A.; Zhang, Z.; Liu, Z.; Chua, T.-S.; and Sun, M. 2021. Cpt: Colorful prompt tuning for pre-trained vision-language models. *arXiv preprint arXiv:2109.11797*.

Zhang, R.; Fang, R.; Zhang, W.; Gao, P.; Li, K.; Dai, J.; Qiao, Y.; and Li, H. 2021. Tip-adapter: Training-free clip-adapter for better vision-language modeling. *arXiv preprint arXiv:2111.03930*.

Zhang, R.; Hu, X.; Li, B.; Huang, S.; Deng, H.; Qiao, Y.; Gao, P.; and Li, H. 2023. Prompt, generate, then cache: Cascade of foundation models makes strong few-shot learners. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 15211–15222.

Zhou, K.; Yang, J.; Loy, C. C.; and Liu, Z. 2022a. Conditional prompt learning for vision-language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 16816–16825.

Zhou, K.; Yang, J.; Loy, C. C.; and Liu, Z. 2022b. Learning to prompt for vision-language models. *International Journal of Computer Vision*, 130(9): 2337–2348.

Zhu, B.; Niu, Y.; Han, Y.; Wu, Y.; and Zhang, H. 2022. Prompt-aligned gradient for prompt tuning. *arXiv preprint arXiv:2205.14865*.

Zhu, D.; Chen, J.; Shen, X.; Li, X.; and Elhoseiny, M. 2023. Minigpt-4: Enhancing vision-language understanding with advanced large language models. *arXiv preprint arXiv:2304.10592*.