ADAPT: Adaptive Prompt Tuning for Vision-Language Models

Anonymous authors
Paper under double-blind review

Abstract

Prompt tuning has emerged as an effective way for parameter-efficient fine-tuning. Conventional deep prompt tuning inserts continuous prompts of a fixed context length into the input to each layer. When a pre-trained model is tailored to a specific downstream task, different layers initialized with pre-trained weights might have different levels of deviation from the optimal weights. Inserted prompts with a fixed context length might have redundant context tokens or insufficient context length. To address this issue, we propose a deep continuous prompting method dubbed Adapt that encourages heterogeneous context lengths. In this method, context lengths are automatically determined by iteratively pruning context tokens. We use the saliency criterion for neural network pruning to compute the importance scores of context tokens in order to determine which tokens to prune. To avoid the forgetting issue in the fine-tuning process, we apply the angular knowledge distillation, introduced to enforce angular separation of instances, to encourage the model to learn the angular separation between pairs of classes and that of instances from the pre-trained model. We examine the proposed method on the pre-trained vision-language model CLIP. 16-shot experiments on 11 downstream datasets reveal the advantage of Adapt: the average test accuracy achieves competitive performance, and the highest performance gain on individual datasets is 7.44%. We release the code in `https://anonymous.4open.science/r/Adapt-Prompt-Release`.

1 Introduction

Large-scale models have gained significant attention in language (Brown, 2020; Wang et al., 2021; Touvron et al., 2023), vision (He et al., 2022; Zou et al., 2024; Esser et al., 2024) and multimodality (Radford et al., 2021; Lu et al., 2022; Lai et al., 2024; Liu et al., 2024). When applying pre-trained large-scale models to various downstream tasks, the zero-shot performance can be sub-optimal. Although fine-tuning remarkably elicits the potential of pre-trained models, the fine-tuning process is computationally expensive. Parameter-efficient fine-tuning (PEFT) offers an efficient way to adapt the pre-trained model to various downstream tasks at a low cost. PEFT enhances the performance of pre-trained models by model weight reparameterization (Hu et al., 2021; Dettmers et al., 2024; Zhao et al., 2024), additive modules (Chen et al., 2022b; Zhang et al., 2023; Mou et al., 2024) and selective weight updates (Ding et al., 2023; Lawton et al., 2023; Fu et al., 2023). Among PEFT methods, prompting methods have the least effect on backbone models as they focus on the input of layers instead of model parameters. Prompt-based adaptation has been applied in various areas, including foundational computer vision (CV) tasks, constrained learning, trustworthy AI, foundational analysis, etc. (Xiao et al., 2025b).

Prompts can be categorized into discrete prompts and continuous prompts (Liu et al., 2023a). Discrete prompts use concrete word tokens to prompt pre-trained models. Compared to discrete prompts, continuous prompts (also called soft prompts) represent trainable tokens in a continuous embedding space. Hence, continuous prompts are differentiable and parameterized by their weights. They can be automatically tuned conditioning on downstream tasks.

Continuous prompts have shown competitive performance in language (Li & Liang, 2021; Gu et al., 2021; Liu et al., 2021; 2023b), vision (Jia et al., 2022; Bahng et al., 2022; Han et al., 2023) and multimodality (Zhou

et al., 2022b; Shu et al., 2022; Ju et al., 2022; Wang et al., 2022). Existing continuous prompting methods use the prompt depth and context length (*i.e.*, the number of context tokens) to design continuous prompts. The underlying constraint is that the context length remains constant at different depths. However, if different layers have different levels of deviation from the optimal weights for downstream tasks, the constraint might be detrimental to the performance.

Recent works (Lee et al., 2022; Chiatti et al., 2023; Panigrahi et al., 2023) have found that some layers of pre-trained models are closer to the optimal for downstream tasks. Fine-tuning layers that are far away from the optimal weights can achieve better performance than training all the layers uniformly. For prompting methods, we postulate that the layers far away from the optimal weights require longer context length while the layers close to the optimal weights demand shorter context length or even no context token. Hence, we seek to remove the constraint in the existing continuous prompting methods that require the same context length at different depths.

To this end, we propose a method dubbed **ada**ptive **p**rompt **t**uing (Adapt) that automatically determines context lengths at various depths. During the training process, Adapt uses time-varying binary masks to dynamically control context lengths. The variation of the binary mask depends on the importance of context tokens. The least important context token is constantly removed until the budget (a hyperparameter to control the total context length) is reached. In vision-language models, predictions are made based on cosine similarity between text embeddings and image embeddings. Traditional KL-divergence loss is sub-optimal as the performance of the foundational model is remarkably worse compared to using soft prompting methods, as shown in Figure 12. Learning angular separation becomes an effective way to distill prior knowledge from the teacher model. Hence, we use angular knowledge distillation to boost the performance. We test the performance of Adapt on various downstream tasks and report a new state-of-the-art average accuracy under the 16-shot setting. To our best knowledge, this is the first work to prune prompts for achieving heterogeneous context lengths in prompt learning.

The main contributions of Adapt are summarized below:

- We propose a method that removes the constraint in the existing continuous prompting methods that context lengths remain constant through the entire prompt depth. Adapt encourages a more flexible and optimal design for prompting methods.
- Context lengths are automatically determined in a non-parametric manner: prompts are initialized with the maximum context length and then iteratively pruned based on the importance score of context tokens. We use saliency criteria to characterize the importance of inserted context tokens. Pruning can effectively reduce the computational overhead with minimal performance drop.
- Context lengths can vary based on the downstream datasets. We use a hyperparameter of total context length to ensure the complexity of Adapt on various datasets is approximately the same while being heterogeneous.

2 Related Work

Prompt Tuning Prompt tuning (PT) uses continuous prompts to improve the performance of pre-trained models in diverse downstream tasks. CoOp (Zhou et al., 2022b) is the pioneering work to apply PT for vision-language models. PT has shown great potential in various areas including image classification (Zhou et al., 2022b;a; Hirohashi et al., 2024), out-of-distribution detection (Miyai et al., 2024; Li et al., 2024a), video understanding (Ju et al., 2022; Huang et al., 2023), object detection (Du et al., 2022; He et al., 2023), etc. Due to the good alignment of text and image representations of foundational vision-language models, there are emerging researches on applying those models such as CLIP (Radford et al., 2021) to vision-language tasks. VPT (Jia et al., 2022) proposes a paradigm of deep continuous prompting. PLOT (Chen et al., 2022a) applies the optimal transport theory to improve the alignment between visual features and prompts. ProGrad (Zhu et al., 2023) distills the prior knowledge from the pre-trained model to avoid forgetting issues (Li & Hoiem, 2017; Gou et al., 2021). MaPLe (Khattak et al., 2023a) uses linear transformation layers to enhance the coupling between the text and image branches while UPT (Zang et al., 2022) utilizes transformer

blocks. PromptSRC (Khattak et al., 2023b) applies the self-regulation strategy to avoid the forgetting issue. DAPT (Cho et al., 2023) proposes the distribution-aware prompts. LAMM (Gao et al., 2024) uses dynamic category embedding and hierarchical loss to achieve an appropriate label distribution. ViAPT proposes instance-aware prompts to balance dataset-level and instance-level knowledge (Xiao et al., 2025a).

Network Pruning Over-parametrization is a well-known property of deep neural networks. Network pruning removes unimportant model parameters to improve efficiency. It can be categorized into structured pruning and unstructured pruning. Unstructured pruning such as (Han et al., 2015) removes individual parameters while structured pruning such as (Liu et al., 2018) prunes models at a higher level (*e.g.* neurons, filters, and layers). A fundamental question in network pruning is to identify a saliency criterion to determine the importance of model parameters. Snip (Lee et al., 2018) is a classic way to characterize the importance and can lead to a very sparse network without sacrificing too much performance.

Knowledge Distillation Knowledge distillation between the foundational model and the fine-tuned model to avoid the catastrophic forgetting issue. It has been found that there can be a gradient conflict when optimizing the KL-divergence loss and the model evaluation loss (Zhu et al., 2023). In addition to traditional supervised knowledge distillation, PromptKD (Li et al., 2024b) applies knowledge distillation in an unsupervised fashion. In addition to matching the probability distribution prediction from the teacher model, the separation between instances is proposed to distill prior knowledge from the teacher model (Park et al., 2019; Tung & Mori, 2019).

3 Adaptive Prompt Tuning (Adapt)

We examine Adapt on the vision-language model CLIP (Radford et al., 2021). CLIP is pre-trained over 400 million image-text pairs. The pre-training process is in a contrastive learning fashion to promote the alignment between text and image representations. CLIP consists of an image encoder and a text encoder. The prediction is done by matching the text and image representations.

3.1 Revisiting CLIP

Given an input image $\mathbf{I} \in \mathbb{R}^{H \times W \times 3}$, the image encoder splits it into fixed-size patches that are projected into patch embeddings $\mathbf{x} \in \mathbb{R}^{(N_i - 1) \times d_i}$ (Dosovitskiy, 2020). d_i is the hidden dimension on the image branch and N_i is the input sequence length. A learnable classification token embedding $\mathbf{c}_i^{(0)}$ is prepended to the patch embeddings. The concatenated sequence of embeddings is passed to ℓ transformer blocks:

$$[\mathbf{c}_i^{(l)}, \mathbf{E}_i^{(l)}] = f^{(l)}([\mathbf{c}_i^{(l-1)}, \mathbf{E}_i^{(l-1)}]) \ , \tag{1}$$

where $l \in \mathbb{N}^+, 1 \leq l \leq \ell$, $f^{(l)}$ is the l-th transformer block of the image encoder. $\mathbf{E}_i^{(0)} = \mathbf{x}$. In the head of the image encoder, a linear transformation layer $\pi_i : \mathbb{R}^{d_i} \to \mathbb{R}^d$ transforms the classification token embedding in the image branch to the image representation $\mathbf{f}$.

A text prompt is fed to the text encoder to obtain the text embedding $\mathbf{E}_t = [\mathbf{w}^1, \mathbf{w}^2, \ldots, \mathbf{w}^{N_t}] \in \mathbb{R}^{N_t \times d_t}$. d_t is the hidden dimension on the text branch. The text embedding contains the classification token embedding as the EOT token embedding. The text embedding is passed to ℓ transformer blocks:

$$\mathbf{E}_t^{(l)} = g^{(l)}(\mathbf{E}_t^{(l-1)}) \ , \tag{2}$$

where $g^{(l)}$ is the l-th transformer block of the text encoder. In the head of the text encoder, a linear transformation layer $\pi_t : \mathbb{R}^{d_t} \to \mathbb{R}^d$ transforms the classification token embedding in the text branch to the text representation $\mathbf{g}$.

The prediction for the input image $\mathbf{I}$ is computed by the cosine similarity between the text embedding and the image embedding:

$$p(y = k|\mathbf{x}) = \frac{\exp(\cos(\mathbf{f}, \mathbf{g}_k)/\tau)}{\sum_{m=1}^{K} \exp(\cos(\mathbf{f}, \mathbf{g}_m)/\tau)} \ . \tag{3}$$

Here τ is the temperature parameter, K is the total number of classes.

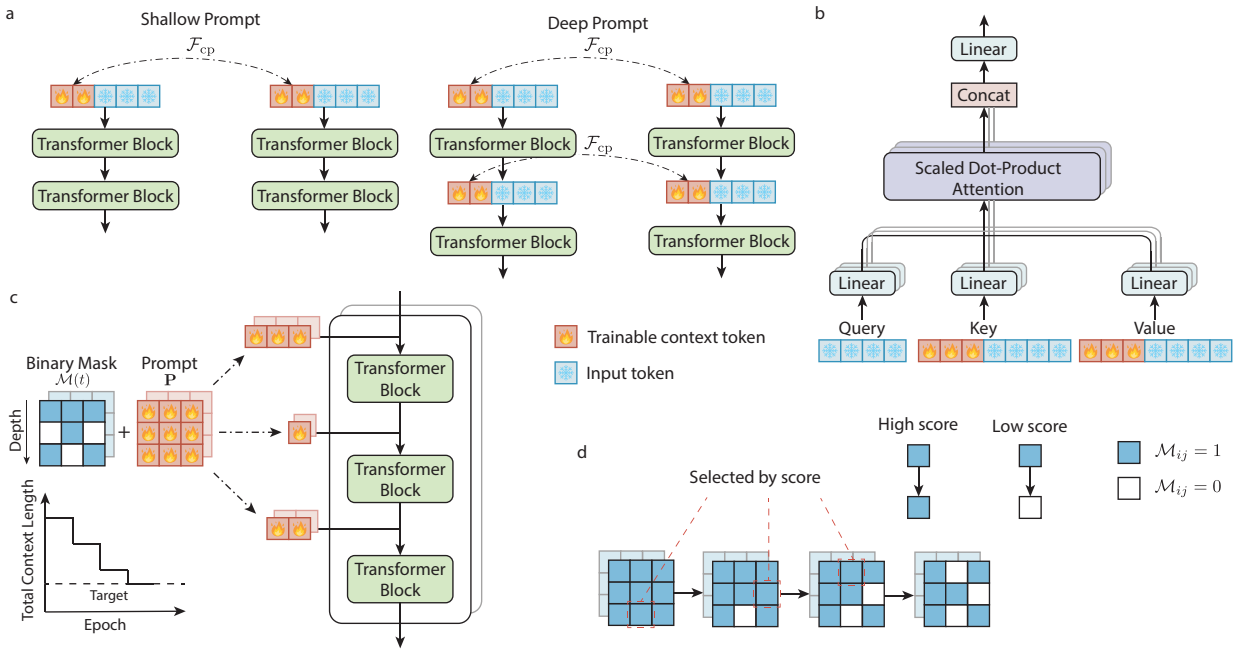

Figure 1: **Overview**. (a) The prompting paradigm of existing shallow and deep prompting methods. A coupling function $\mathcal{F}_{cp}$ can be applied to bridge the text branch to the image branch. (b) In the Adapt framework, we insert continuous prompts for key and value computation. The backbone model is frozen during the fine-tuning process. Only continuous prompts are trainable (trainable continuous token). Frozen discrete token essentially corresponds to input to the transformer block. (c) The proposed Adapt method encourages the pre-trained model to insert prompts with different context lengths. We use two binary masks $\mathcal{M}_f(t)$ and $\mathcal{M}_g(t)$ to adaptively control context lengths. Context lengths constantly change until the target $\mathcal{T}_{\text{target}}$ is reached. (d) The selection of context tokens to be pruned is based on the saliency scores.

3.2 Dynamic and Heterogeneous Soft Prompts

Figure 1 (a) showcases the traditional shallow and deep prompts for vision-language models. Figure 1 (b)-(d) show the proposed Adapt method. Different from the traditional deep prompting method, the Adapt method enables heterogeneous prompts inserted without changing the context lengths of hidden representations.

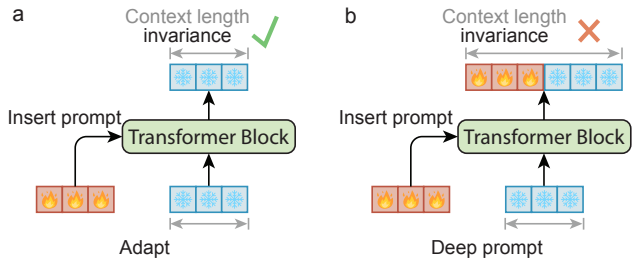

Figure 2: Comparison between (a) the Adapt method and (b) deep prompting paradigm for vision-language models. Adapt does not change the sequence length and hence enables heterogeneous context lengths at various depths, *i.e.*, context lengths are different at different depths.

In the fine-tuning process of the pre-trained model, Adapt maximizes the likelihood of the correct label y:

$$\max_{\mathbf{P}\odot\mathcal{M}(t)} \mathbb{P}_{\mathbf{P}\odot\mathcal{M}(t),\boldsymbol{\theta}}(y|\mathbf{x}, \mathbf{P}\odot\mathcal{M}(t), \boldsymbol{\theta}) \ , \tag{4}$$

where $\boldsymbol{\theta}$ is the weight of the pre-trained model that is frozen during the fine-tuning process. $\mathbf{P} \in \mathbb{R}^{\ell\times\xi\times d}$ is the inserted continuous prompt. ξ is the maximum context length at various depths. $\mathcal{M}(t) \in \{0,1\}^{\ell\times\xi}$ is a

time-dependent (dynamic) binary mask. We use $\odot$ to denote a modified Hadamard operation $\mathbf{M} = \mathbf{P} \odot \mathcal{M}(t)$, where $M_{ijk} = P_{ijk}\mathcal{M}(t)_{ij}$, $1 \leq i \leq \ell, 1 \leq j \leq \xi, 1 \leq k \leq d$. For the vision-language model, there are two sets of independent prompts and binary masks. The optimization objective is over $\mathbf{P}_f \odot \mathcal{M}_f(t)$ and $\mathbf{P}_g \odot \mathcal{M}_g(t)$. $\mathbf{P}_f$ and $\mathbf{P}_g$ are prompts for image and text branches. $\mathcal{M}_f(t)$ and $\mathcal{M}_g(t)$ are binary masks for image and text branches.

We describe the optimization process of Adapt for the vision-language model as:

$$\begin{aligned} \underset{\substack{\mathbf{P}_f, \mathcal{M}_f(t), \\ \mathbf{P}_g, \mathcal{M}_g(t)}}{\operatorname{argmin}} \quad & \frac{1}{|\mathcal{D}|} \sum_{\mathbf{x}, y \in \mathcal{D}} \mathcal{L}(\mathbf{x}, y | \mathbf{P}_f, \mathcal{M}_f(t), \mathbf{P}_g, \mathbf{M}_g(t), \boldsymbol{\theta}) \ , \\ \text{s.t.} \quad & \sum_{i=1}^{\ell_f} \sum_{j=1}^{\xi_f} \mathcal{M}_f(t)_{ij} + \sum_{i=1}^{\ell_g} \sum_{i=1}^{\xi_g} \mathcal{M}_g(t)_{ij} \leq \mathcal{T}_{\text{target}} \ , \end{aligned} \tag{5}$$

where the hyperparameter $\mathcal{T}_{\text{target}}$ is the target total context length. It determines the complexity of the Adapt method. For brevity, we do not explicitly mention $\mathcal{M}_f(t)$ for the image branch and $\mathcal{M}_g(t)$ for the text branch. Instead we use $\mathcal{M}(t)$ as it can be applied to both the image and text branch. $\mathcal{M}(t)$ is initialized to be $\mathcal{M}(0) = \mathbf{1}_{\ell \times \xi}$ (an all-one matrix). At each iteration, we identify which token to prune and set the corresponding binary mask to be 0, *i.e.* $\mathcal{M}(t)_{ij} = 0$. The total context length continuously decreases until $\mathcal{T}_{\text{target}}$ is reached. We use $\mathcal{T}_{\text{target}} \ll \ell \times \xi$ to ensure the efficiency of the Adapt method.

In the pruning process, which context token to prune, *i.e.* finding i, j and set $\mathcal{M}(t)_{ij} = 0$, is determined by the importance of corresponding context tokens as shown in Figure 1 (d). We borrow the saliency criterion widely used in the unstructured network pruning literature to measure the importance of context tokens. Specifically, we use Snip (Lee et al., 2018), gradient norm and l_2-norm to compute the importance scores S_c (also called saliency scores) for characterizing the importance. For the t-th context token ($t \in \mathbb{N}^+, 1 \leq t \leq \xi$) at the depth l ($l \in \mathbb{N}^+, 1 \leq l \leq \ell$), the importance score computed by these three metrics is:

$$\begin{aligned} &\text{Snip: } S_c = \left| \frac{\partial \mathcal{L}}{\partial \mathbf{P}_{lt}} \odot \mathbf{P}_{lt} \right| , \\ &\text{gradient norm: } S_c = \left| \frac{\partial \mathcal{L}}{\partial \mathbf{P}_{lt}} \right| , \\ &l_2\text{-norm: } S_c = |\mathbf{P}_{lt}| \ . \end{aligned} \tag{6}$$

$\mathcal{M}(t)$ controls the context length for each transformer block as shown in Figure 1 (b). $\mathcal{M}(t) \odot \mathbf{P}$ is the continuous prompt inserted to the pre-trained model. There is no constraint for context lengths at various depths. Hence, the added prompt $\mathbf{P} \odot \mathcal{M}(t)$ can be heterogeneous.

3.3 Prompt Tuning

Owing to $\mathcal{M}(t)$, the context length ξ_l varies during the fine-tuning process. Unlike the existing deep prompting methods for the vision-language models that insert continuous prompts in the computation of key, value and query, Adapt inserts continuous prompts only for key and value in the self-attention (Vaswani et al., 2017) as shown in Figure 1 (c). Different from the deep prompting paradigm, the context length does not change as indicated in Figure 2. Given an input $\mathbf{x}$ for a transformer block, the self-attention with inserted prompts in Adapt is computed by:

$$\begin{aligned} \mathbf{Q} &= f_q(\mathbf{x}) \in \mathbb{R}^{\xi_{\text{org}} \times d}, \\ \mathbf{K} &= f_k([\mathbf{P}, \mathbf{x}]) \in \mathbb{R}^{(\xi_{\text{org}} + \xi) \times d}, \\ \mathbf{V} &= f_v([\mathbf{P}, \mathbf{x}]) \in \mathbb{R}^{(\xi_{\text{org}} + \xi) \times d} \ . \end{aligned} \tag{7}$$

$$\text{Self-Attention} = \text{Softmax}(\frac{\mathbf{Q}\mathbf{K}^T}{\sqrt{d}})\mathbf{V} \ . \tag{8}$$

Here f_q, f_k and f_v are linear transformation functions for the query, key and value. ξ_{org} is the sequence length of the input without the inserted prompt. After self-attention, the resulting sequence has a sequence

length of ξ_{org}, which is independent of ξ. During the fine-tuning process, the pre-trained model is frozen. Only inserted continuous prompts are optimized.

The proposed Adapt method for vision-language models is summarized in Algorithm 1. In the pruning step, the ranking is done based on scores in the image and text branches. The context tokens with the lowest score will be removed. The total context length in the text branch can be different from that in the image branch. For the same branch, context lengths might vary at different depths. Hence, compared to the manually designed continuous prompt, Adapt can have highly heterogeneous context lengths. Besides, using the saliency criterion enables varying context lengths without additional trainable parameters.

Till the total context lengths of text and image branches reach $\mathcal{T}_{\text{target}}$, Adapt continues to remove context tokens. The accumulation period n_k determines the number of accumulated steps to compute the score. The pruning rate r_p dictates the number of removed tokens per pruning step.

Algorithm 1 Adapt for vision-language models.

Input: A pre-trained vision-language model, prompt depth ℓ_f for the image encoder and ℓ_g for the text encoder, maximum context length ξ_f for the image encoder and ξ_g for the text encoder, target $\mathcal{T}_{\text{target}}$, accumulation period n_k and pruning rate r_p.
Create a randomly initialized prompt $\mathbf{P}_f \in \mathbb{R}^{\ell_f \times \xi_f \times d}$ for the image branch and $\mathbf{P}_g \in \mathbb{R}^{\ell_g \times \xi_g \times d}$ for the text branch, and a binary mask $\mathcal{M}_f(0) = \mathbf{1}_{\ell_f \times \xi_f}$ for the image branch and $\mathcal{M}_g(0) = \mathbf{1}_{\ell_g \times \xi_g}$ for the text branch.
Initialize accumulated score $\mathbf{S}_f = \mathbf{0}_{\ell_f \times \xi_f}$ and $\mathbf{S}_g = \mathbf{0}_{\ell_g \times \xi_g}$.
for $t = \{1, \ldots, n_t\}$ **do**
    Insert the prompt $\mathbf{P}_f \odot \mathcal{M}_f(t)$ for the image branch and $\mathbf{P}_g \odot \mathcal{M}_g(t)$ for the text branch of the pre-trained model as shown in Equation 7 and 8.
    Perform forward and backward propagation using loss function in Equation 10 to update $\mathbf{P}_f$ and $\mathbf{P}_g$.
    if $\sum_{i=1}^{\ell_f} \sum_{j=1}^{\xi_f} \mathcal{M}_f(t)_{ij} + \sum_{i=1}^{\ell_g} \sum_{j=1}^{\xi_g} \mathcal{M}_g(t)_{ij} > \mathcal{T}_{\text{target}}$ **then**
        $\Delta\mathbf{S}_f \in \mathbb{R}^{\ell_f \times \xi_f}$ and $\Delta\mathbf{S}_g \in \mathbb{R}^{\ell_g \times \xi_g}$ according to Equation 6.
        Update $\mathbf{S}_f$ by $\mathbf{S}_f = \mathbf{S}_f + \Delta\mathbf{S}_f$ and $\mathbf{S}_g$ by $\mathbf{S}_g = \mathbf{S}_g + \Delta\mathbf{S}_g$.
        if $t == an_k, a \in \mathbb{N}^+$ **then**
            for Prune step $= \{1, \ldots, r_p\}$ **do**
                Find location $(k_{\min}, i_{\min}, j_{\min}) = \text{argmin}_{k,i,j}\{[\mathbf{S}_k]_{ij} | \mathcal{M}_k(t)_{ij} == 1\}$.
                $\mathcal{M}_{k_{\min}}(t)_{i_{\min} j_{\min}} = 0$.
            end for
            Reset accumulated score $\mathbf{S}_f = \mathbf{0}_{\ell_f \times \xi_f}$ and $\mathbf{S}_g = \mathbf{0}_{\ell_g \times \xi_g}$.
        end if
    end if
end for

3.4 Knowledge Distillation

Fine-tuning pre-trained models in various downstream tasks is subject to the forgetting issue (Niu et al., 2022; Smith et al., 2023; Wang et al., 2022). Knowledge distillation (KD) (Hinton et al., 2015) is an effective way to distill knowledge from the teacher model to the student model. A direct idea is to use the pre-trained model as the teacher model to distill the prior knowledge and to avoid the forgetting issue. However, there is a large performance gap between the pre-trained model and prompting methods as shown in Table 1. Hence, the logits predicted by the teacher model are sub-optimal. PromptSRC (Khattak et al., 2023b) proposes a self-regulation loss on image and text embeddings for the prompt tuning. They test the performance of using different types of self-regulating losses and conclude that l_1 distance between the embeddings of the student model and the teacher model performs best in their setting. Following their setting, we test both cosine similarity and l_1 distance. We find that cosine similarity generates better performance in our setting. We denote this self-regulation loss as angular KD loss. The angular KD aligns the cosine similarity between embeddings of the student model and the teacher model. In vision-language models, the prediction is based on cosine similarity between text and image embeddings. Angular KD considers cosine similarity between

embeddings of the student model and the teacher model, which preserves the prior knowledge in the teacher model. The student model is forced to learn the angular separation between pairs of text embeddings for different classes and that between pairs of image embeddings for different images. Specifically, we apply a loss term associated with the angular separation:

$$\mathcal{L}_{\text{ang}} = \mathbb{E}_{\mathbf{x}^s \sim \mathcal{X}^s, \mathbf{x}^t \sim \mathcal{X}^t} \left| \langle \mathbf{x}_i^t, \mathbf{x}_j^t \rangle - \langle \mathbf{x}_i^s, \mathbf{x}_j^s \rangle \right| , \tag{9}$$

where $\langle \cdot \rangle$ computes the cosine similarity between two embeddings. $\mathbf{x}^s$ is the embedding generated by the student model while $\mathbf{x}^t$ is the embedding generated by the teacher model. $\mathbf{x}_i, \mathbf{x}_j \in \mathcal{X}$ are any pair of embeddings.

It has been found that a slight variation in the embedding can lead to classification error for vision-language models (Cho et al., 2023; Hirohashi et al., 2024). $\mathcal{L}_{\text{ang}}$ helps the student model learn the prior knowledge by learning the angular separation between pairs of text embeddings of different classes and that of pairs of different image embeddings. The total training loss is computed by:

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{ce}} + \alpha \cdot \mathcal{L}_{\text{ang}}^{\text{text}} + \beta \cdot \mathcal{L}_{\text{ang}}^{\text{image}} . \tag{10}$$

Here α and β are hyperparameters to control the contribution of two loss terms related to angular KD. $\mathcal{L}_{\text{ce}}$ is the cross-entropy loss.

4 Experiments and Results

4.1 Experiments

Datasets We examine the proposed Adapt method over 11 datasets: Caltech101 (Fei-Fei et al., 2004) and ImageNet (Deng et al., 2009) for the generic object recognition, Describable Textures (Cimpoi et al., 2014) for the texture recognition, EuroSAT (Helber et al., 2019) for the satellite image recognition, FGVCAircraft (Maji et al., 2013), Food101 (Bossard et al., 2014), OxfordFlowers (Nilsback & Zisserman, 2008), OxfordPets (Parkhi et al., 2012), and StanfordCars (Krause et al., 2013) for the fine-grained image recognition, UCF101 (Soomro et al., 2012) for the action recognition, and SUN397 (Xiao et al., 2010) for the scene recognition. We follow the few-shot learning setting in CoOp (Zhou et al., 2022b). The number of shots is 16. For each dataset, the result is averaged over 3 runs. A detailed description of 11 datasets can be found in Appendix Section A.3.

Baselines We compare the proposed method with CoOp (Zhou et al., 2022b), VPT (Jia et al., 2022), CoCoOp (Zhou et al., 2022a), PLOT (Chen et al., 2022a), UPT (Zang et al., 2022), ProGrad (Zhu et al., 2023), DAPT (Cho et al., 2023), MaPLe (Khattak et al., 2023a), PromptSRC (Khattak et al., 2023b) and LAMM (Gao et al., 2024). CoOp, CoCoOp, PLOT, ProGrad, LAMM and DAPT use shallow prompts while UPT, VPT-Deep, MaPLe and PromptSRC use deep prompts. CoOp (Zhou et al., 2022b) inserts continuous prompts into the text prompt for vision-language models. CoCoOp (Zhou et al., 2022a) adds continuous prompts conditioned on the input Image. VPT (Jia et al., 2022) proposes a paradigm of the deep prompting method. ProGrad (Zhu et al., 2023) proposes the gradient-aligned knowledge distillation. PLOT (Chen et al., 2022a) utilizes the optimal transport theory (Cuturi, 2013) to align text and image embeddings. DAPT (Cho et al., 2023) proposes distribution-aware prompting methods. UPT (Zang et al., 2022) uses the transformer layer to contextualize prompt tokens in both text and image branches. MaPLe (Khattak et al., 2023a) uses a linear projection in lieu of transformer layers to bridge the image branch and text branch. PromptSRC (Khattak et al., 2023b) proposes self-ensembling for prompt templates, self-regulation for embeddings and prediction logits, and Gaussian weighted sampling for model weights. LAMM (Gao et al., 2024) replaces the category tokens with trainable vectors and utilizes the hierarchical loss to preserve the generalization ability of the pre-trained model. Tip-Adapter-F (Zhang et al., 2021) utilizes a key-value cache model to determine weights of the adapter.

Implementation Details We use the pre-trained ViT-B/16 CLIP model (Radford et al., 2021) in this work. The text template used for prompting the vision-language model is summarized in Appendix Section

Table 1: Test accuracy comparison on various downstream tasks in the few-shot learning setting. The number of shots is 16. The reported performance is averaged over 3 runs. Adapt uses Snip to compute scores of context tokens.

Method	Caltech101	DTD	EuroSAT	Aircraft	Food101	Flowers	Pets	Cars	Sun	UCF	ImageNet	Average
ZS CLIP	87.20	42.34	37.57	17.29	77.30	66.18	85.79	55.63	58.55	61.45	58.20	58.86
Linear Probe	95.50	69.69	87.21	45.39	83.11	97.47	86.40	80.69	82.32	72.95	67.46	78.93
Tip-Adapter-F	93.42	66.73	84.63	39.13	79.15	94.34	89.21	75.43	70.47	77.5	80.23	77.29
VPT-Shallow	94.66	52.27	84.93	30.9	86.91	81.42	92.61	69.05	75.18	68.07	68.98	73.18
CoCoOp	95.18	62.98	73.28	31.25	87.17	87.83	93.18	71.55	78.08	72.12	70.81	74.86
PLOT	93.70	70.90	84.03	34.93	78.13	97.27	88.20	68.10	72.23	69.40	72.17	75.37
ProGrad	95.63	66.27	82.03	41.30	86.70	95.33	93.10	81.23	81.60	75.13	72.27	79.14
VPT-Deep	95.83	69.77	91.60	40.88	86.17	94.97	92.91	76.09	82.74	71.62	70.57	79.38
CoOp	95.58	69.81	84.96	43.43	84.14	97.05	91.87	83.05	82.16	74.64	71.81	79.86
UPT	95.95	70.61	90.77	46.57	86.82	97.40	92.77	84.17	83.87	75.91	72.65	81.59
LAMM	**96.90**	72.21	82.71	45.42	86.43	97.36	93.11	85.10	83.78	76.09	72.71	81.07
DAPT	95.80	71.55	92.58	46.21	86.59	97.04	92.28	82.99	84.48	75.96	72.26	81.61
MaPLe	95.97	71.34	92.35	48.31	85.28	96.97	92.97	83.41	84.99	75.61	72.27	81.77
PromptSRC	96.04	72.75	92.48	50.86	**87.44**	97.62	**93.81**	83.79	**86.43**	**77.24**	73.12	82.87
Adapt	96.57	**74.07**	**93.53**	**58.30**	86.67	**98.43**	92.67	**85.97**	**86.43**	77.17	**73.20**	**83.91**

A.3. The number of mini-batches used for computing the score is $n_k = 80$. We use $\ell_f = \ell_g = 12$, $\xi_f = \xi_g = 18$ and $\mathcal{T}_{\text{target}} = 256$. In each pruning step, the number of pruned context tokens is $r_p = 2$. The batch size is 4. The learning rate is 2.5×10^{-3}. The total number of training epochs is 100. The test accuracy is obtained using the model weights at the epoch of 100. We use the stochastic gradient descent (SGD) to optimize the inserted prompts. Experiments are conducted using a single NVIDIA A40 GPU. Reported results on 11 datasets are averaged over 3 runs.

4.2 Results

The effectiveness of the Adapt method is examined using the few-shot learning setting. We summarize the experimental results in Table 1. We use Snip to compute scores. The performance of gradient norm and l_2-norm is reported in Appendix Figure 13 (c). Overall, the Adapt method exhibits superior performance compared to baseline methods. The Adapt method improves the best-known performance (82.87% by PromptSRC) to 83.91% on the average of 11 datasets. The largest performance gain is 7.44% on the FGVC Aircraft dataset. A Wilcoxon signed-rank test is conducted to compare the accuracy of our method and the best baseline method PromptSRC. We get a p-value of $0.003 < 0.05$, indicating a statistically significant performance gain. The performance of using a smaller number of shots is reported in Appendix Figure 8. We notice that when the number of shots is less than 8, there is a degradation of the advantage of the Adapt method compared to baseline methods. This might be caused by poor convergence of bi-level optimization when the data are very limited.

Notably, Adapt inserts continuous prompts only for the key and value computations while the prevalent deep prompting methods for vision-language models insert continuous prompts for query, key and value computations. Using the same context length, this approach can effectively decrease the model complexity. Besides, the continuous prompts are not added to the query, which does not change the context length after the attention computation. Therefore, Adapt enables the addition of prompts with heterogeneous context lengths to the pre-trained model.

Adapt remarkably improves the performance of zero-shot CLIP model as shown in Appendix Figure 12. On each of 11 datasets, the Adapt method pronouncedly improves the performance. The highest test accuracy gain is 55.96% on the EuroSAT dataset. Compared to other prompting methods, there is also a large performance gap for the zero-shot CLIP model as shown in Table 1. This seems to indicate that the pre-trained CLIP model is not an ideal candidate as the teacher's model for Kullback–Leibler (KL) divergence on the prediction of the probability distribution of various classes. The angular distillation distills the prior knowledge by learning the angular separation, which avoids the distillation by the predicted probability distribution from the pre-trained CLIP model.

We examine the prompts after pruning. An example result is shown in Figure 3. The full results are shown in Appendix Figure 15. Adapt enables automatically determining highly heterogeneous prompts. The context lengths at different depths and different branches can be different.

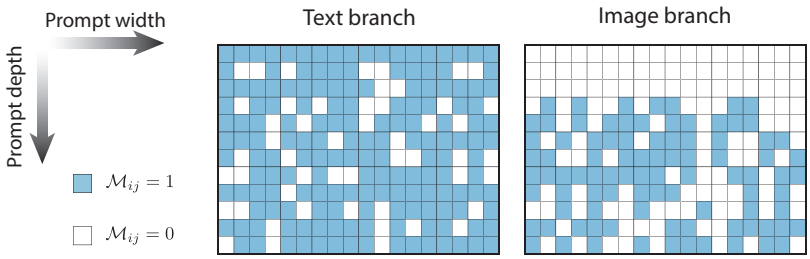

Figure 3: Example of prompts after pruning. The dataset is Stanford Cars. $\mathcal{M}_{ij} = 1$ indicates that the context token is used while $\mathcal{M}_{ij} = 0$ indicates that the context token is removed. Prompts after pruning are highly heterogeneous.

4.3 Ablation Study

Angular knowledge distillation Angular knowledge distillation enables the distillation of prior knowledge learned from the pre-training process. The traditional way of knowledge distillation (KD) is KL divergence (Hinton et al., 2015) of the predicted probability distribution between the student model and the teacher model. However, the pre-trained CLIP model has inferior performance as shown in Table 1 and Appendix Figure 12. The angular distillation avoids learning potentially sub-optimal prediction of the probability distribution.

Figure 4 (a) and (c) shows the distribution of image embeddings using angular KD and without using angular KD. Angular KD enables a better separation of embeddings of different classes and clustering of embeddings of the same classes. Figure 4 (b) shows the comparison of the average test accuracy over 11 datasets between these two approaches. The angular KD effectively improves the performance.

Maximum context length Considering the pruning process as a process of searching the optimal context length for each transformer block, the maximum context lengths ξ_f for the image branch and ξ_g for the text branch determine the size of the search space. When $\xi_f = \xi_g < 18$, the performance degrades even though $\mathcal{T}_{\text{target}}$ stays the same as shown in Figure 5. In other words, although the total context length of the final prompt stays the same, the maximum context length (essentially the size of the search space) has an effect on the performance of the Adapt method. We use $\mathcal{T}_{\text{effective}}$ to indicate the total number of context tokens after pruning.

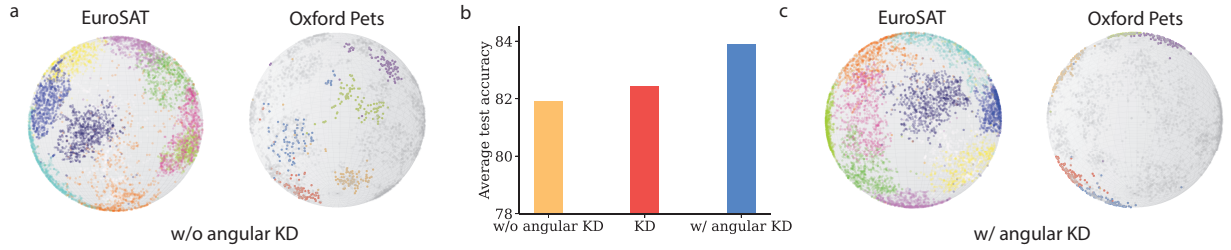

Figure 4: Effect of the angular knowledge distillation on model performance. (a) t-SNE (Van der Maaten & Hinton, 2008) image embeddings without applying angular knowledge distillation on downstream datasets. On the Oxford Pets dataset, we selectively highlight some classes for better visualization; (b) Applying angular knowledge distillation improves the average test accuracy compared to both without angular knowledge distillation and traditional knowledge distillation; (c) t-SNE image embedding using angular knowledge distillation.

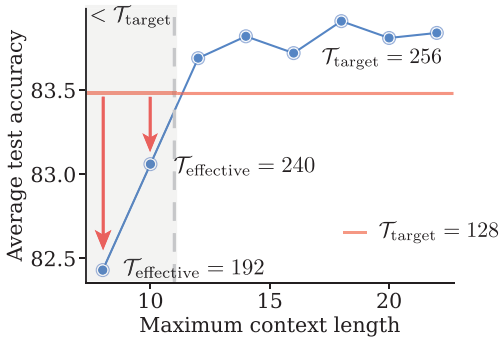

Figure 5: The effect of maximum context length $\xi_f = \xi_g$ on the model performance. In the shaded area, the total number of context tokens is smaller than $\mathcal{T}_{\text{target}}$ before pruning. Red line marks the performance of using $\mathcal{T}_{\text{target}} = 128$ in lieu of $\mathcal{T}_{\text{target}} = 256$. Even though the total context length (heterogeneous prompt) is smaller than that (homogeneous prompt without any pruning) in the shaded area, it has pronouncedly better performance.

We notice that when ξ_f and ξ_g are small enough to enter the regime where the total context length before pruning is smaller than $\mathcal{T}_{\text{target}}$ (the shaded area in Figure 5), there is a remarkable performance degradation. The performance is even worse than using $\mathcal{T}_{\text{target}} = 128$. We use $\mathcal{T}_{\text{effective}}$ to represent the total context length after pruning. The red horizontal line in Figure 5 uses the $\xi_f = \xi_g = 18$ and $\mathcal{T}_{\text{target}} = 128$. The total context length after pruning is $\mathcal{T}_{\text{effective}} = \mathcal{T}_{\text{target}} = 128$. The performance is pronouncedly better than the two configurations in the shaded area. This indicates the pruning process not only decreases the computational cost but also improves the performance.

5 Discussion

When tailoring a pre-trained model to various downstream tasks, the model can underperform (Taori et al., 2020; Fang et al., 2020; Wiles et al., 2021; Xiao et al., 2024). When examining the model on a more granular level, a question arises "*is the inferior performance caused by the deviation from the optimal for all layers or a subset of layers*". Surgical fine-tuning (Lee et al., 2022) finds that fine-tuning the selective part of the pre-trained model achieves a performance comparable to or better than training all layers. This result indicates that not all layers are at the same level of deviation from the optimal. Selectively training layers deviating from the optimal while keeping the remaining layers frozen achieves favorable performance.

In prompt tuning, the entire pre-trained model is frozen. Given the fact that some layers might already be near optimal, there is no need to insert continuous prompts for those layers. Prompts can be inserted into layers that are deviating from the optimal. If we consider this strategy in a more granular way, context lengths for different layers can vary depending on the level of deviation from the optimal. This leads to heterogeneous context lengths which are challenging for the manually designed prompting methods.

The proposed Adapt method achieves the automatic design of heterogeneous prompts. There is no constraint for context lengths at various depths to be the same, nor for context lengths to be the same for different branches. The results on 11 datasets indicate that context lengths can be highly heterogeneous as shown in Appendix Figure 15. The automation is achieved by iteratively pruning unimportant context tokens. By setting $\mathcal{T}_{\text{target}} \ll \ell_f \times \xi_f + \ell_g \times \xi_g$, the pruning greatly reduces the computational overheads. The comparison is based on Adapt before pruning and Adapt after pruning. The total number of trainable parameters is decreased by 41%. In the network pruning, pruning concentrated on one layer can cause the layer collapse issue (Lee et al., 2019; Hayou et al., 2020). Pruning prompts, however, can have a minimal context length in one layer without affecting the functionality of prompts for this layer.

By using $\mathcal{M}(t)$ conditioning on downstream datasets, Adapt adaptively changes for different datasets. Compared to manually designed prompts, Adapt has a more flexible structure. It achieves a pronounced performance gain compared to baseline methods. We use $\mathcal{T}_{\text{target}}$ to control the complexity of Adapt.

6 Conclusion

We propose a continuous prompting method that adaptively changes during the fine-tuning process. Different from existing prompting methods that require homogeneous context lengths for various depths, our proposed method Adapt encourages heterogeneous context lengths, unlocking the space of prompt learning that was previously underexplored. Adapt uses iterative pruning to remove unimportant context tokens, which greatly reduces the computational costs. Compared to homogeneous prompting that is commonly used in manually designed prompts, the heterogeneous prompting enabled by Adapt has better performance. Extensive experiments over 11 datasets exhibit the strength of the Adapt method, achieving competitive performance in prompt learning for pre-trained vision-language models.

7 Limitation

The Adapt method uses the two-level optimization with a higher level optimizing prompt configuration and a lower level optimizing weights of soft prompts. When the dataset size is limited (e.g. very limited number of shots in the few-shot learning), there can be a performance degradation.

8 Broader Impact

We propose an automatic way to determine a heterogeneous prompt, which can be applied to fine-tune foundational models in various downstream tasks efficiently. Different from existing manually designed prompting approaches that use a homogeneous prompt, heterogeneous prompts improve efficiency by pruning unimportant context tokens. While this work focuses on VLMs, our method can, in principle, be applied to pure language and vision transformer models.

References

Hyojin Bahng, Ali Jahanian, Swami Sankaranarayanan, and Phillip Isola. Exploring visual prompts for adapting large-scale models. *arXiv preprint arXiv:2203.17274*, 2022.

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*, pp. 446–461. Springer, 2014.

Lorenzo Brigato and Luca Iocchi. A close look at deep learning with small data. In *2020 25th international conference on pattern recognition (ICPR)*, pp. 2490–2497. IEEE, 2021.

Tom B Brown. Language models are few-shot learners. *arXiv preprint arXiv:2005.14165*, 2020.

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

Shoufa Chen, Chongjian Ge, Zhan Tong, Jiangliu Wang, Yibing Song, Jue Wang, and Ping Luo. Adaptformer: Adapting vision transformers for scalable visual recognition. *Advances in Neural Information Processing Systems*, 35:16664–16678, 2022b.

Agnese Chiatti, Riccardo Bertoglio, Nico Catalano, Matteo Gatti, and Matteo Matteucci. Surgical fine-tuning for grape bunch segmentation under visual domain shifts. In *2023 European Conference on Mobile Robots (ECMR)*, pp. 1–7. IEEE, 2023.

Eulrang Cho, Jooyeon Kim, and Hyunwoo J Kim. Distribution-aware prompt tuning for vision-language models. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 22004–22013, 2023.

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*, pp. 3606–3613, 2014.

Marco Cuturi. Sinkhorn distances: Lightspeed computation of optimal transport. *Advances in neural information processing systems*, 26, 2013.

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*, pp. 248–255. Ieee, 2009.

Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. Qlora: Efficient finetuning of quantized llms. *Advances in Neural Information Processing Systems*, 36, 2024.

Ning Ding, Yujia Qin, Guang Yang, Fuchao Wei, Zonghan Yang, Yusheng Su, Shengding Hu, Yulin Chen, Chi-Min Chan, Weize Chen, et al. Parameter-efficient fine-tuning of large-scale pre-trained language models. *Nature Machine Intelligence*, 5(3):220–235, 2023.

Alexey Dosovitskiy. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

Yu Du, Fangyun Wei, Zihe Zhang, Miaojing Shi, Yue Gao, and Guoqi Li. Learning to prompt for open-vocabulary object detection with vision-language model. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 14084–14093, 2022.

Patrick Esser, Sumith Kulal, Andreas Blattmann, Rahim Entezari, Jonas Müller, Harry Saini, Yam Levi, Dominik Lorenz, Axel Sauer, Frederic Boesel, et al. Scaling rectified flow transformers for high-resolution image synthesis. In *Forty-first International Conference on Machine Learning*, 2024.

Tongtong Fang, Nan Lu, Gang Niu, and Masashi Sugiyama. Rethinking importance weighting for deep learning under distribution shift. *Advances in neural information processing systems*, 33:11996–12007, 2020.

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*, pp. 178–178. IEEE, 2004.

Zihao Fu, Haoran Yang, Anthony Man-Cho So, Wai Lam, Lidong Bing, and Nigel Collier. On the effectiveness of parameter-efficient fine-tuning. In *Proceedings of the AAAI conference on artificial intelligence*, volume 37, pp. 12799–12807, 2023.

Jingsheng Gao, Jiacheng Ruan, Suncheng Xiang, Zefang Yu, Ke Ji, Mingye Xie, Ting Liu, and Yuzhuo Fu. Lamm: Label alignment for multi-modal prompt learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pp. 1815–1823, 2024.

Jianping Gou, Baosheng Yu, Stephen J Maybank, and Dacheng Tao. Knowledge distillation: A survey. *International Journal of Computer Vision*, 129(6):1789–1819, 2021.

Yuxian Gu, Xu Han, Zhiyuan Liu, and Minlie Huang. Ppt: Pre-trained prompt tuning for few-shot learning. *arXiv preprint arXiv:2109.04332*, 2021.

Cheng Han, Qifan Wang, Yiming Cui, Zhiwen Cao, Wenguan Wang, Siyuan Qi, and Dongfang Liu. Eˆ 2vpt: An effective and efficient approach for visual prompt tuning. *arXiv preprint arXiv:2307.13770*, 2023.

Song Han, Jeff Pool, John Tran, and William Dally. Learning both weights and connections for efficient neural network. *Advances in neural information processing systems*, 28, 2015.

Soufiane Hayou, Jean-Francois Ton, Arnaud Doucet, and Yee Whye Teh. Robust pruning at initialization. *arXiv preprint arXiv:2002.08797*, 2020.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 16000–16009, 2022.

Weizhen He, Weijie Chen, Binbin Chen, Shicai Yang, Di Xie, Luojun Lin, Donglian Qi, and Yueting Zhuang. Unsupervised prompt tuning for text-driven object detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 2651–2661, 2023.

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.

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

Yuki Hirohashi, Tsubasa Hirakawa, Takayoshi Yamashita, and Hironobu Fujiyoshi. Prompt learning with one-shot setting based feature space analysis in vision-and-language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 7761–7770, 2024.

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

Siteng Huang, Biao Gong, Yulin Pan, Jianwen Jiang, Yiliang Lv, Yuyuan Li, and Donglin Wang. Vop: Text-video co-operative prompt tuning for cross-modal retrieval. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 6565–6574, 2023.

Menglin Jia, Luming Tang, Bor-Chun Chen, Claire Cardie, Serge Belongie, Bharath Hariharan, and Ser-Nam Lim. Visual prompt tuning. In *European Conference on Computer Vision*, pp. 709–727. Springer, 2022.

Chen Ju, Tengda Han, Kunhao Zheng, Ya Zhang, and Weidi Xie. Prompting visual-language models for efficient video understanding. In *European Conference on Computer Vision*, pp. 105–124. Springer, 2022.

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

Muhammad Uzair Khattak, Syed Talal Wasim, Muzammal Naseer, Salman Khan, Ming-Hsuan Yang, and Fahad Shahbaz Khan. Self-regulating prompts: Foundational model adaptation without forgetting. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 15190–15200, 2023b.

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*, pp. 554–561, 2013.

Xin Lai, Zhuotao Tian, Yukang Chen, Yanwei Li, Yuhui Yuan, Shu Liu, and Jiaya Jia. Lisa: Reasoning segmentation via large language model. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 9579–9589, 2024.

Neal Lawton, Anoop Kumar, Govind Thattai, Aram Galstyan, and Greg Ver Steeg. Neural architecture search for parameter-efficient fine-tuning of large pre-trained language models. *arXiv preprint arXiv:2305.16597*, 2023.

Namhoon Lee, Thalaiyasingam Ajanthan, and Philip HS Torr. Snip: Single-shot network pruning based on connection sensitivity. *arXiv preprint arXiv:1810.02340*, 2018.

Namhoon Lee, Thalaiyasingam Ajanthan, Stephen Gould, and Philip HS Torr. A signal propagation perspective for pruning neural networks at initialization. *arXiv preprint arXiv:1906.06307*, 2019.

Yoonho Lee, Annie S Chen, Fahim Tajwar, Ananya Kumar, Huaxiu Yao, Percy Liang, and Chelsea Finn. Surgical fine-tuning improves adaptation to distribution shifts. *arXiv preprint arXiv:2210.11466*, 2022.

Tianqi Li, Guansong Pang, Xiao Bai, Wenjun Miao, and Jin Zheng. Learning transferable negative prompts for out-of-distribution detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 17584–17594, 2024a.

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

Zheng Li, Xiang Li, Xinyi Fu, Xin Zhang, Weiqiang Wang, Shuo Chen, and Jian Yang. Promptkd: Unsupervised prompt distillation for vision-language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 26617–26626, 2024b.

Zhizhong Li and Derek Hoiem. Learning without forgetting. *IEEE transactions on pattern analysis and machine intelligence*, 40(12):2935–2947, 2017.

Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. *Advances in neural information processing systems*, 36, 2024.

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

Xiao Liu, Kaixuan Ji, Yicheng Fu, Weng Lam Tam, Zhengxiao Du, Zhilin Yang, and Jie Tang. P-tuning v2: Prompt tuning can be comparable to fine-tuning universally across scales and tasks. *arXiv preprint arXiv:2110.07602*, 2021.

Xiao Liu, Yanan Zheng, Zhengxiao Du, Ming Ding, Yujie Qian, Zhilin Yang, and Jie Tang. Gpt understands, too. *AI Open*, 2023b.

Zhuang Liu, Mingjie Sun, Tinghui Zhou, Gao Huang, and Trevor Darrell. Rethinking the value of network pruning. *arXiv preprint arXiv:1810.05270*, 2018.

Pan Lu, Swaroop Mishra, Tanglin Xia, Liang Qiu, Kai-Wei Chang, Song-Chun Zhu, Oyvind Tafjord, Peter Clark, and Ashwin Kalyan. Learn to explain: Multimodal reasoning via thought chains for science question answering. *Advances in Neural Information Processing Systems*, 35:2507–2521, 2022.

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

Atsuyuki Miyai, Qing Yu, Go Irie, and Kiyoharu Aizawa. Locoop: Few-shot out-of-distribution detection via prompt learning. *Advances in Neural Information Processing Systems*, 36, 2024.

Chong Mou, Xintao Wang, Liangbin Xie, Yanze Wu, Jian Zhang, Zhongang Qi, and Ying Shan. T2i-adapter: Learning adapters to dig out more controllable ability for text-to-image diffusion models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pp. 4296–4304, 2024.

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*, pp. 722–729. IEEE, 2008.

Shuaicheng Niu, Jiaxiang Wu, Yifan Zhang, Yaofo Chen, Shijian Zheng, Peilin Zhao, and Mingkui Tan. Efficient test-time model adaptation without forgetting. In *International conference on machine learning*, pp. 16888–16905. PMLR, 2022.

Abhishek Panigrahi, Nikunj Saunshi, Haoyu Zhao, and Sanjeev Arora. Task-specific skill localization in fine-tuned language models. In *International Conference on Machine Learning*, pp. 27011–27033. PMLR, 2023.

Wonpyo Park, Dongju Kim, Yan Lu, and Minsu Cho. Relational knowledge distillation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 3967–3976, 2019.

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

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*, pp. 8748–8763. PMLR, 2021.

Ramprasaath R Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *Proceedings of the IEEE international conference on computer vision*, pp. 618–626, 2017.

Manli Shu, Weili Nie, De-An Huang, Zhiding Yu, Tom Goldstein, Anima Anandkumar, and Chaowei Xiao. Test-time prompt tuning for zero-shot generalization in vision-language models. *Advances in Neural Information Processing Systems*, 35:14274–14289, 2022.

James Seale Smith, Leonid Karlinsky, Vyshnavi Gutta, Paola Cascante-Bonilla, Donghyun Kim, Assaf Arbelle, Rameswar Panda, Rogerio Feris, and Zsolt Kira. 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*, pp. 11909–11919, 2023.

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.

Rohan Taori, Achal Dave, Vaishaal Shankar, Nicholas Carlini, Benjamin Recht, and Ludwig Schmidt. Measuring robustness to natural distribution shifts in image classification. *Advances in Neural Information Processing Systems*, 33:18583–18599, 2020.

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

Hsi-Ai Tsao, Lei Hsiung, Pin-Yu Chen, Sijia Liu, and Tsung-Yi Ho. Autovp: An automated visual prompting framework and benchmark. *arXiv preprint arXiv:2310.08381*, 2023.

Frederick Tung and Greg Mori. Similarity-preserving knowledge distillation. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 1365–1374, 2019.

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

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Geomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in Neural Information Processing Systems*, 2017.

Sinong Wang, Han Fang, Madian Khabsa, Hanzi Mao, and Hao Ma. Entailment as few-shot learner. *arXiv preprint arXiv:2104.14690*, 2021.

Yabin Wang, Zhiwu Huang, and Xiaopeng Hong. S-prompts learning with pre-trained transformers: An occam's razor for domain incremental learning. *Advances in Neural Information Processing Systems*, 35: 5682–5695, 2022.

Olivia Wiles, Sven Gowal, Florian Stimberg, Sylvestre Alvise-Rebuffi, Ira Ktena, Krishnamurthy Dvijotham, and Taylan Cemgil. A fine-grained analysis on distribution shift. *arXiv preprint arXiv:2110.11328*, 2021.

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*, pp. 3485–3492. IEEE, 2010.

Xi Xiao, Yunbei Zhang, Xingjian Li, Tianyang Wang, Xiao Wang, Yuxiang Wei, Jihun Hamm, and Min Xu. Visual instance-aware prompt tuning. In *Proceedings of the 33rd ACM International Conference on Multimedia*, pp. 2880–2889, 2025a.

Xi Xiao, Yunbei Zhang, Lin Zhao, Yiyang Liu, Xiaoying Liao, Zheda Mai, Xingjian Li, Xiao Wang, Hao Xu, Jihun Hamm, et al. Prompt-based adaptation in large-scale vision models: A survey. *arXiv preprint arXiv:2510.13219*, 2025b.

Zehao Xiao, Jiayi Shen, Mohammad Mahdi Derakhshani, Shengcai Liao, and Cees GM Snoek. Any-shift prompting for generalization over distributions. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 13849–13860, 2024.

Yuhang Zang, Wei Li, Kaiyang Zhou, Chen Huang, and Chen Change Loy. Unified vision and language prompt learning. *arXiv preprint arXiv:2210.07225*, 2022.

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

Renrui Zhang, Jiaming Han, Chris Liu, Peng Gao, Aojun Zhou, Xiangfei Hu, Shilin Yan, Pan Lu, Hongsheng Li, and Yu Qiao. Llama-adapter: Efficient fine-tuning of language models with zero-init attention. *arXiv preprint arXiv:2303.16199*, 2023.

Bowen Zhao, Hannaneh Hajishirzi, and Qingqing Cao. Apt: Adaptive pruning and tuning pretrained language models for efficient training and inference. *arXiv preprint arXiv:2401.12200*, 2024.

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

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, 2022b.

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

Xueyan Zou, Jianwei Yang, Hao Zhang, Feng Li, Linjie Li, Jianfeng Wang, Lijuan Wang, Jianfeng Gao, and Yong Jae Lee. Segment everything everywhere all at once. *Advances in Neural Information Processing Systems*, 36, 2024.

A Appendix

A.1 Time-Dependent Context Length

Figure 6 shows the variation of context lengths during the fine-tuning process. When $\mathcal{T}_{\text{target}}$ is reached, the total context length for the image and text branches stays constant. We use the warm-up strategy and hence the total context length does not change for the first few epochs. In the pruning process, we prune the context tokens in the text and image branches with the lowest score. The number of tokens removed per epoch can vary from the image branch to the text branch. Within the same branch, the context length can vary at various depths. Overall, Adapt encourages highly heterogeneous context lengths which can be difficult for the manually designed prompts. Adapt is able to automatically determine context lengths. The pruning processes on all 11 datasets are visualized in Figure 15.

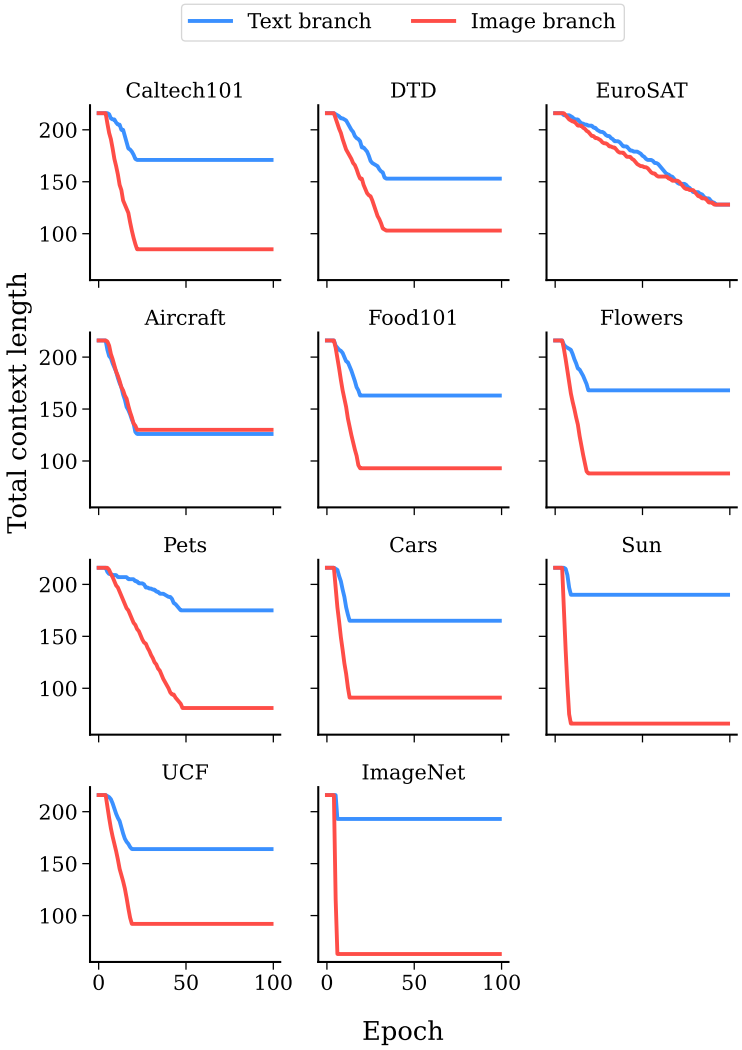

Figure 6: Variation of the total context length during the fine-tuning process on 11 datasets. The plateau in the initial stage is caused by the warm-up strategy. The second one is the regime in which the pruning is finished, *i.e.* $\mathcal{T}_{\text{effective}} = \mathcal{T}_{\text{target}}$.

A.2 Model Attention Using Prompts

We use Grad-CAM (Selvaraju et al., 2017) to examine the localized attention of the CLIP model using zero-shot CLIP, PromptSRC (state-of-the-art baseline) and Adapt. Figure 7 shows the comparison of the model attention. We find the difference in the localized attention between Adapt and zero-shot CLIP is remarkably larger than that between PromptSRC and zero-shot CLIP. Besides, Adapt pronouncedly improves the model's attention on salient features of image objects and avoids the distraction from the background.

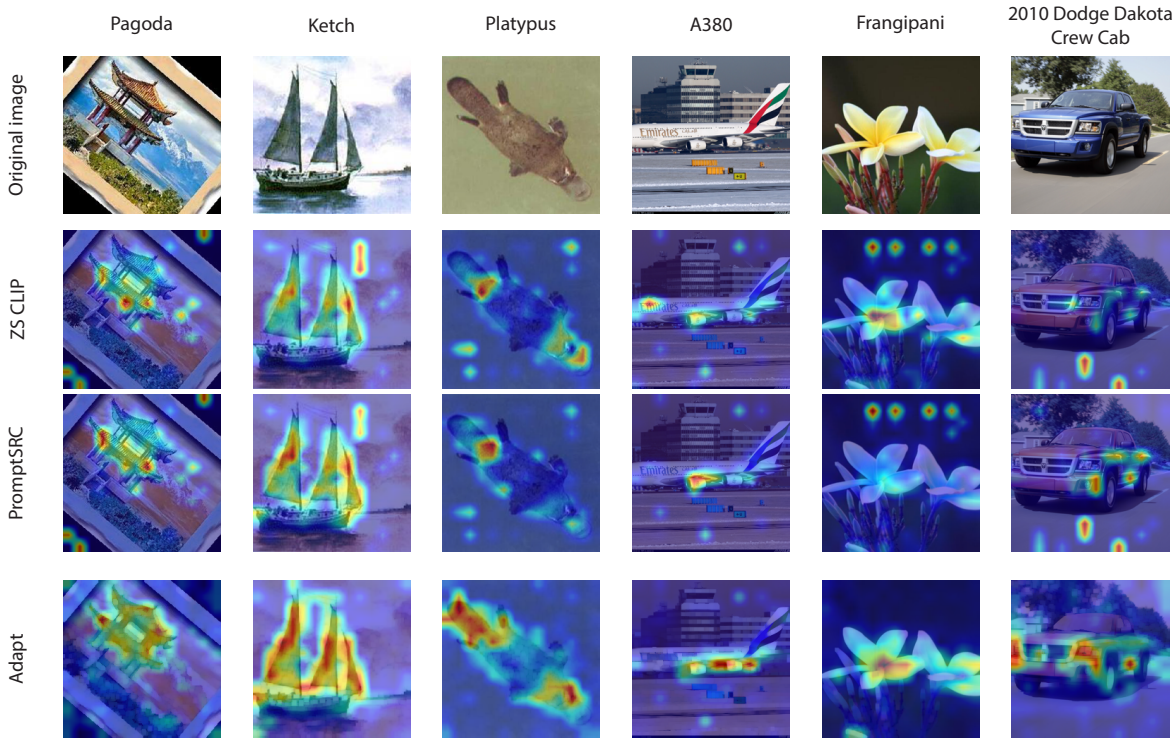

Figure 7: Grad-CAM (Selvaraju et al., 2017) visualization of the model's localized attention. The Adapt method helps the backbone model focus on the salient features of input images and avoid distractions from the background.

A.3 Datasets for Downstream Tasks

Table 2 shows statistics of 11 datasets used for the fine-tuning of the pre-trained CLIP model and text prompt templates used for prompting the vision language model. These datasets covering a wide range of tasks are commonly used as benchmarks for vision-language models. Of 11 datasets, only Caltech101 (Fei-Fei et al., 2004) and ImageNet (Deng et al., 2009) are generic object classification datasets. EuroSAT (Helber et al., 2019), FGVCAircraft (Maji et al., 2013), Food101 (Bossard et al., 2014), OxfordFlowers (Nilsback & Zisserman, 2008) are OxfordPets (Parkhi et al., 2012) are fine-grained classification datasets including images of sub-types under a common type. The text templates we use for 11 datasets are the same as ProGrad (Zhu et al., 2023).

A.4 Few-shot Learning

Figure 8 shows the few-shot learning results using $\{16, 8, 4, 2, 1\}$ shots. The pruning process of the Adapt method depends on the pruning rate r_p and the number of mini-batches n_k. When the number of shots is very small, $\mathcal{T}_{\text{target}}$ might not be reached due to the limited number of mini-batches. To ensure $\mathcal{T}_{\text{target}}$ is

Table 2: Details regarding 11 datasets and corresponding text prompts used for prompting vision-language models. We replace `[class]` with real class names and feed text prompts to text encoders for obtaining corresponding text embeddings.

Dataset	# Classes	Training size	Test size	Task	Text prompt template
Caltech101 (Fei-Fei et al., 2004)	100	4,128	2,465	Generic object classification	a photo of a [class]
ImageNet (Deng et al., 2009)	1,000	1.28M	50,000	Generic object classification	a photo of a [class]
Describable Textures (Cimpoi et al., 2014)	47	2,820	1692	Texture classification	a texture of [class]
EuroSAT (Helber et al., 2019)	10	13,500	8,100	Satellite image classification	a centered satellite photo of [class]
FGVCAircraft (Maji et al., 2013)	100	3,334	3,333	Fine-grained aircraft classification	a type of aircraft, a photo of a [class]
Food101 (Bossard et al., 2014)	101	50,500	30,300	Fine-grained food classification	a type of food, a photo of a [class]
OxfordFlowers (Nilsback & Zisserman, 2008)	102	4,093	2,463	Fine-grained flower classification	a type of flower, a photo of a [class]
OxfordPets (Parkhi et al., 2012)	37	2,944	3,669	Fine-grained pet classification	a type of pet, a photo of a [class]
StanfordCars (Krause et al., 2013)	196	6,509	8,041	Fine-grained car classification	a photo of a [class]
UCF101 (Soomro et al., 2012)	101	7,639	3,783	Action classification	a photo of a [class]
SUN397 (Xiao et al., 2010)	397	15,880	19,850	Scene classification	a photo of a [class]

reached, we decrease ξ_f and ξ_g as the number of shots decreases. We notice that as the number of shots decreases, the Adapt method does not maintain the best performance over all baselines. The cause might be related to the increasingly challenging optimization when the training data have a very limited size. The Adapt framework requires not only optimizing prompt weights but also the prompt configuration, *i.e.* the context length for each transformer block. It has been found that when a limited number of samples per class is available, the model complexity is a critical factor (Brigato & Iocchi, 2021). The performance can be improved by decreasing the model complexity. Besides, prompting methods generally perform poorly when only very limited data are available. New architecture design can boost the performance in the setting of very limited data (Hirohashi et al., 2024).

In addition to comparing test accuracies as shown in Table 1. We calculate the 95% confidence interval (CI) to statistically compare our method and the best baseline method PromptSRC. On the datasets where the Adapt method shows tangible performance boost compared to the PromptSRC baseline (*e.g.* Aircraft dataset), we do not observe an overlapping in CI. The result shows that our approach shows the performance gain statistically.

Table 3: Performance comparison including CI. We use results over 3 runs to compute CI.

Method	Caltech101	DTD	EuroSAT	Aircraft	Food101	Flowers
PromptSRC	96.04 ± 0.17	72.75 ± 0.96	92.48 ± 0.17	50.86 ± 0.70	87.44 ± 0.05	97.62 ± 0.41
Adapt	96.57± 0.05	74.07 ± 0.79	93.53 ± 0.17	58.30 ± 0.73	86.67 ± 0.00	98.43 ± 0.14
Method	**Pets**	**Cars**	**SUN**	**UCF**	**ImageNet**	
PromptSRC	93.81 ± 0.40	83.79 ± 0.68	86.43 ± 0.12	77.24 ± 0.61	73.12 ± 0.22	
Adapt	92.67 ± 0.12	85.97 ± 0.12	86.43 ± 0.09	77.17 ± 0.38	73.20 ± 0.12	

A.5 Baseline Performance

We obtain the performance of PromptSRC (Khattak et al., 2023b), MaPLe (Khattak et al., 2023a), CoOp (Zhou et al., 2022b) and CoCoOp (Zhou et al., 2022a) from (Khattak et al., 2023b), that of VPT-Deep, VPT-Shallow, and UPT (Zang et al., 2022) from (Zang et al., 2022), that of LAMM from (Gao et al., 2024), that of linear probe approach and DAPT from (Cho et al., 2023). PLOT (Chen et al., 2022a) and ProGrad (Zhu et al., 2023) use ResNet50 (He et al., 2016) as the backbone of the image encoder. We replace the ResNet50 model with ViT-B/16 model and test the performance for a fair comparison.

A.6 Effect of Pruning Rate

The pruning rate r_p determines the rate of binary mask reaching $\mathcal{T}_{\text{target}}$. To study the effect of r_p on the final binary mask, we use the agreement ratio to characterize the agreement of the binary mask between two

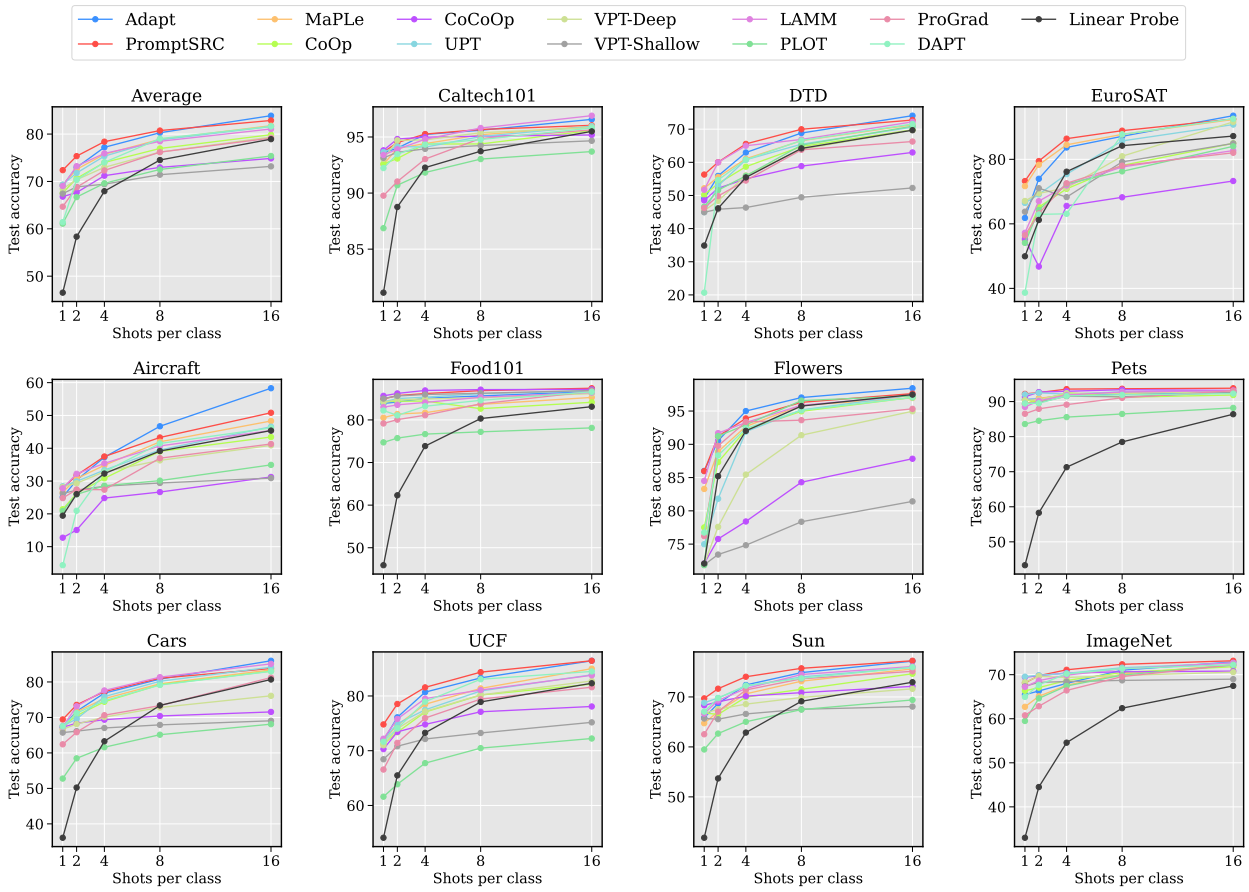

Figure 8: Few-shot learning in the image classification task on 11 datasets. The number of shots is $\{16, 8, 4, 2, 1\}$.

different pruning rates:

$$\text{Agreement ratio} \coloneqq \frac{1}{\ell \times \xi} \sum_{i=1}^{\ell} \sum_{j=1}^{\xi} \mathbb{I}\{\mathcal{M}_1(t)_{ij} = \mathcal{M}_2(t)_{ij}\} \ , \tag{11}$$

where $\mathbb{I}$ is the indicator function and $\mathbb{I}\{E\} = 1$ if the event E happens, otherwise $\mathbb{I}\{E\} = 0$. $\mathcal{M}_1(t)$ and $\mathcal{M}_2(t)$ are binary matrices using two different pruning rates. The agreement ratio is in the range $[0, 1]$. A higher agreement ratio indicates two binary masks are closer.

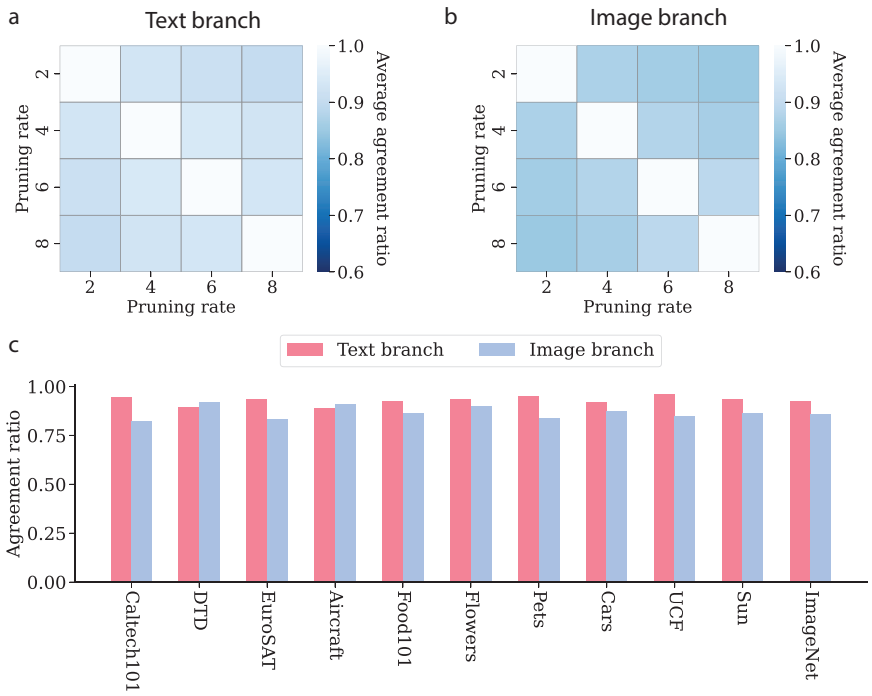

Figure 9: Agreement ratio of binary masks at the last epoch in the text and image branches. We visualize the average pair-wise agreement ratio using different pruning rates $r_p \in \{2, 4, 6, 8\}$ for (a) text branch (b) image branch. (c) The agreement ratio on each of 11 datasets between $r_p = 2$ and $r_p = 4$.

Figure 9 (a) and (b) show the average agreement ratio between pairs of pruning rates $r_p \in \{2, 4, 6, 8\}$. A fixed $\mathcal{T}_{\text{target}} = 256$ is used. The agreement ratio in the text branch is slightly higher than that in the image branch. Overall, the Adapt method is not sensitive to r_p. Using different r_p leads to similar binary masks. Figure 9 (c) shows the agreement ratio on each dataset using $r_p = 2$ and $r_p = 4$. On all 11 datasets, the agreement ratio remains on a high level. The result using different r_p indicates pruning one context token has a nearly negligible effect on the importance of remaining context tokens, which justifies the pruning strategy in the Adapt method. The Adapt method prunes context tokens based on saliency scores. Considering a scenario where r_p context tokens are pruned at the iteration t. If one removed context token can lead to an increase in the saliency score of another pruned context token, pruning r_p tokens at the same time might lead to the removal of important tokens. Results shown in Figure 9 exclude this possibility.

A.7 Pruning Process of Binary Masks

Figure 15 shows the pruning process of binary masks in the text and image branches for 11 datasets. The binary mask is initialized to be $\mathcal{M}(0) = \mathbf{1}_{\ell \times \xi}$. Given a binary mask $\mathcal{M}(t)$, the row dimension is related to the prompt depth ℓ while the column dimension is associated with the prompt width (also called context length). $\mathcal{M}(t)$ stays constant when $\mathcal{T}_{\text{target}}$ is reached. We use Snip to compute the score for each context tokens. The binary masks are highly heterogeneous: context lengths over prompt depth for different branches have

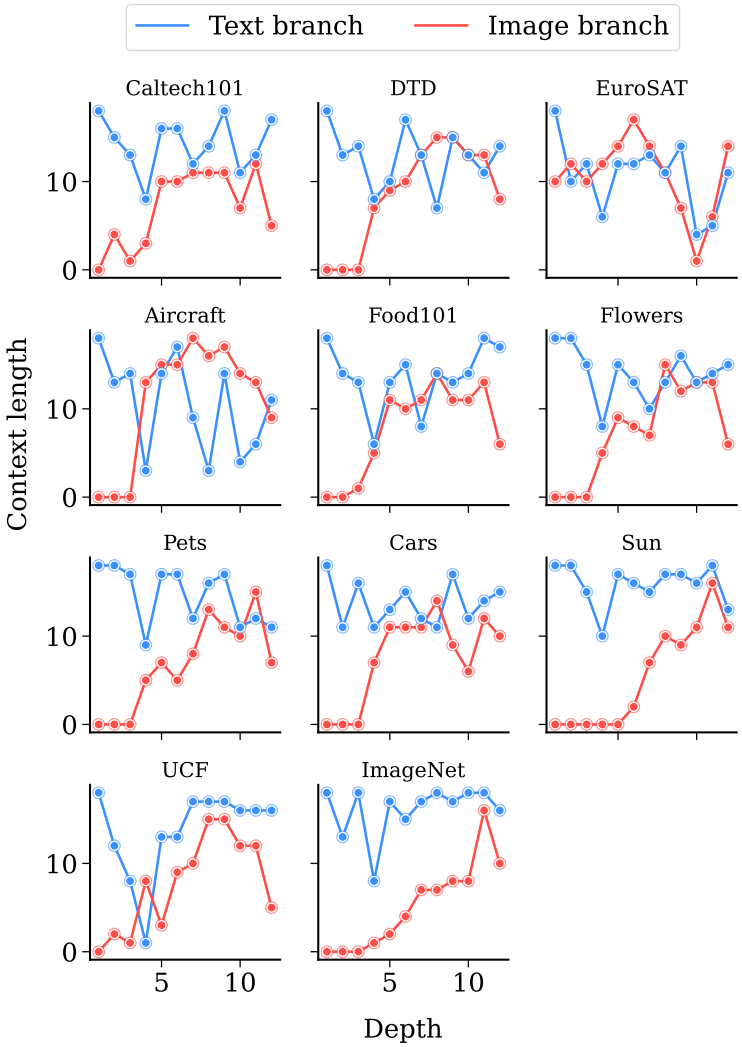

Figure 10: Context lengths at various depths after pruning, *i.e.* $\mathcal{T}_{\text{effective}} = \mathcal{T}_{\text{target}}$. The Adapt method generally prefers to prune context tokens on the image branch.

a large variation. The heterogeneity feature exhibits the strength of the automatic design of context lengths compared to manually designed prompts which tend to be homogeneous.

When $\mathcal{T}_{\text{target}}$ is reached, $\mathbf{P} \odot \mathcal{M}(t)$ on some datasets have a context length of 1 at a certain depth. Even though the pruning concentrated in the prompts for this layer, the performance is not negatively affected. In the network pruning, however, the concentrated pruning can lead to the layer collapse issue (Lee et al., 2019; Hayou et al., 2020). Considering an extreme case, if the weight of an entire layer is pruned, the model will have a disconnection issue. When pruning the prompt, there is no such issue. This indicates that pruning prompts can lead to highly heterogeneous prompt lengths.

After pruning, *i.e.* $\mathcal{T}_{\text{effective}} = \mathcal{T}_{\text{target}}$, we examine the context lengths at various depths. Figure 10 shows the context lengths and Figure 15 is the corresponding pruning process. Generally, the total context length on the text branch is higher than that in the image branch. Besides, the first layer on the text branch has the highest context length. Shallow prompting paradigms for vision-language models such as CoOp (Zhou et al., 2022b) insert prompt only to the text input for the text encoder of the vision-language model. This is consistent to the result shown in Figure 10: inserting prompts to the first layer on the text branch is important to boost the performance of pre-trained models.

Figure 11 shows context lengths at different depths after pruning. We do not observe a uniform context length pattern across datasets, which justifies the automatic pruning strategy.

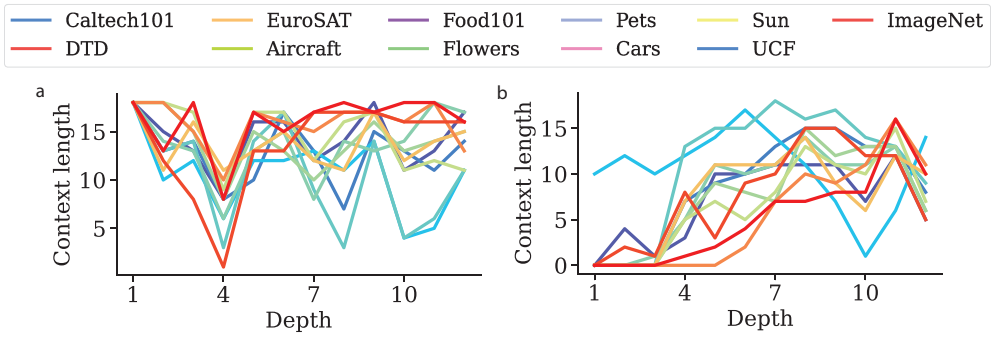

Figure 11: Context lengths across model depth after pruning for 11 datasets. Context lengths exhibit both dataset-dependent and depth-dependent behaviors. (a) Text branch; (b) Image branch.

A.8 Statistics of Reported Performance

Table 4 shows the standard deviation of the performance of the Adapt method over 3 runs. The number of shots is 16. We use $\mathcal{T}_{\text{target}} = 256$, $\xi_f = \xi_g = 18$, $l_f = l_g = 12$. We do not find a significant performance variation on any of 11 datasets.

Table 4: Standard deviation of the reported performance of the Adapt method over 3 runs.

Caltech101	DTD	EuroSAT	Aircraft	Food101	Flowers
96.57 ± 0.25	74.07 ± 0.39	93.53 ± 1.16	58.30 ± 0.87	86.67 ± 0.05	98.43 ± 0.28
Pets	Cars	Sun	UCF	ImageNet	
92.67 ± 0.08	85.97 ± 0.40	86.43 ± 0.17	77.17 ± 0.29	73.20 ± 0.26	

A.9 Performance Boost Compared to Zero-Shot CLIP

Figure 12 shows the performance boost of the Adapt method in comparison with zero-shot CLIP model. The performance boost correlates with the distribution shift of the downstream dataset compared to the pretraining dataset. EuroSAT is the out-of-distribution (OOD) dataset (Tsao et al., 2023) and has the largest performance boost of 55.96%.

A.10 Ablation Study

Target total context length $\mathcal{T}_{\text{target}}$ $\mathcal{T}_{\text{target}}$ is associated with the complexity of inserted prompts. Figure 13 (a) shows the performance variation when changing $\mathcal{T}_{\text{target}}$. When decreasing $\mathcal{T}_{\text{target}}$ from 256 to 128, the average test accuracy drops by 0.43% while the total context length after pruning becomes half of the original length. Continuing decreasing $\mathcal{T}_{\text{target}}$ leads to a pronounced degradation of the model performance. Increasing $\mathcal{T}_{\text{target}}$ from 256 to 512, however, results in the inferior performance.

Figure 14 shows the effect of $\mathcal{T}_{\text{target}}$ on the performance. Different from the results shown in Figure 13 (a), pruning rate is adjusted to ensure $\mathcal{T}_{\text{target}} = \mathcal{T}_{\text{effective}}$ when pruning is finished.

Heterogeneous prompting We compare the performance using homogeneous prompting to Adapt using heterogeneous prompting. For the homogeneous prompting, we use $\xi_f = \xi_g = 12$, and no pruning is applied. The Adapt method uses $\mathcal{T}_{\text{target}} = 240$ to ensure the total context length $\mathcal{T}_{\text{effective}}$ is the same as that of

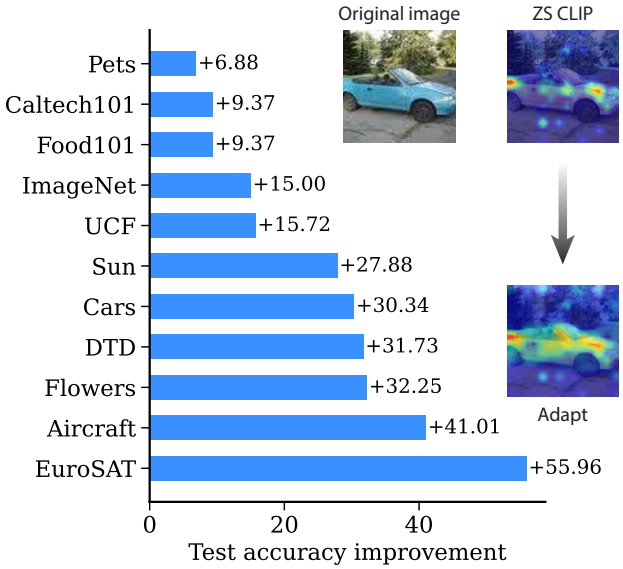

Figure 12: Performance improvement on 11 datasets compared to zero-shot CLIP model. The performance of the zero-shot CLIP model is greatly boosted by the Adapt method. The method helps the backbone model focus on salient features of input images. More examples of Grad-CAM (Selvaraju et al., 2017) visualization can be found in Appendix Figure 7. The significant performance enhancement indicates that knowledge distillation using KL divergence of the predicted probability distribution might be sub-optimal.

homogeneous prompting. Figure 13 (b) shows the performance comparison. Using heterogeneous prompts boosts performance.

Score computation We examine the effect of three different scoring functions: Snip (Lee et al., 2018), gradient norm, and l_2-norm. Snip considers both the gradient and magnitude of the prompt parameters. The comparison is shown in Figure 13 (c). Snip has the best performance. Overall, there is no remarkable difference among scoring functions.

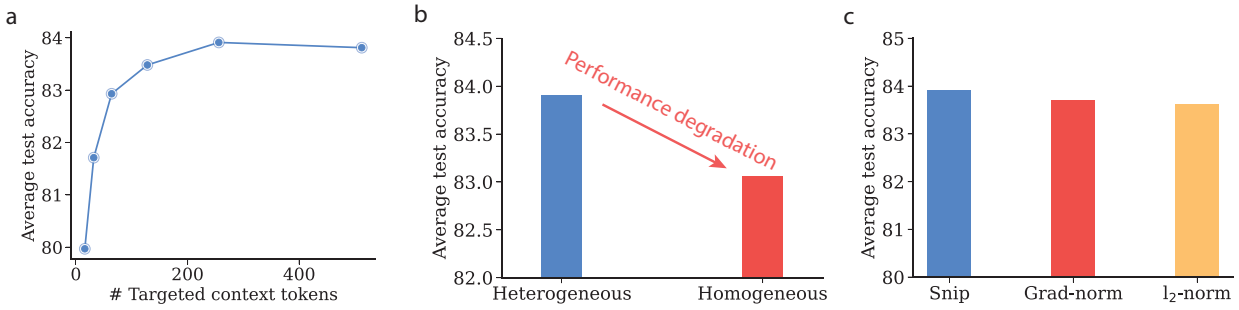

Figure 13: Ablation study on the effect of (a) target total context length $\mathcal{T}_{\text{target}}$. When $\mathcal{T}_{\text{target}} < 128$, $\mathcal{T}_{\text{target}}$ is not reached after pruning, *i.e.* $\mathcal{T}_{\text{effective}} < \mathcal{T}_{\text{target}}$. (b) heterogeneous prompting, and (c) scoring function.

A.11 Terminology

We use different terminologies in the prompt tuning. To ensure consistency in the terminology and improve the readability, we list a detailed explanation for all related terminologies:

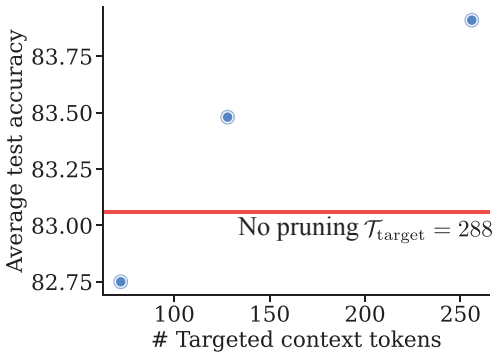

Figure 14: Effect of $\mathcal{T}_{\text{target}}$ on the performance. $\mathcal{T}_{\text{target}} \in \{72, 128, 256\}$ is examined. To ensure $\mathcal{T}_{\text{effective}} = \mathcal{T}_{\text{target}}$, we adjust the pruning rate accordingly. Horizontal line indicates the performance of Adapt without applying pruning and $\mathcal{T}_{\text{target}} = 288$.

- **Discrete prompting**: Discrete prompting is a common strategy in prompt engineering. It aims to search for an optimal prompt to elicit the potential of foundational models. The search space focuses on pure natural language words.
- **Continuous prompting**: continuous prompting, also called soft prompting, inserts differentiable context tokens in the prompt. Compared to discrete prompting that has a discrete search space, continuous prompting has a continuous search space. Hence, the optimized continuous context tokens might not correspond to any word tokens.
- **Shallow prompt tuning**: Shallow prompt tuning inserts context tokens only in the input to foundational models. Given a text prompt $[[w_1], \ldots, [w_k]]$, where $[w_i]$ is the word embedding for token w_i. One way of inserting shallow prompt is $[[e_1], \ldots [e_m], [w_1] \ldots, [w_k]]$, where $[e_i]$ is the inserted continuous prompting that is optimized during the training process.
- **Deep prompt tuning**: Deep prompt tuning inserts context tokens into not only the input to foundational models but also that to deeper model blocks. The deep prompting can be expressed by $\mathbf{x}^{(l)} = \mathcal{F}^{(l)}([\mathbf{x}^{(l-1)}, \mathbf{P}^{(l)}])$

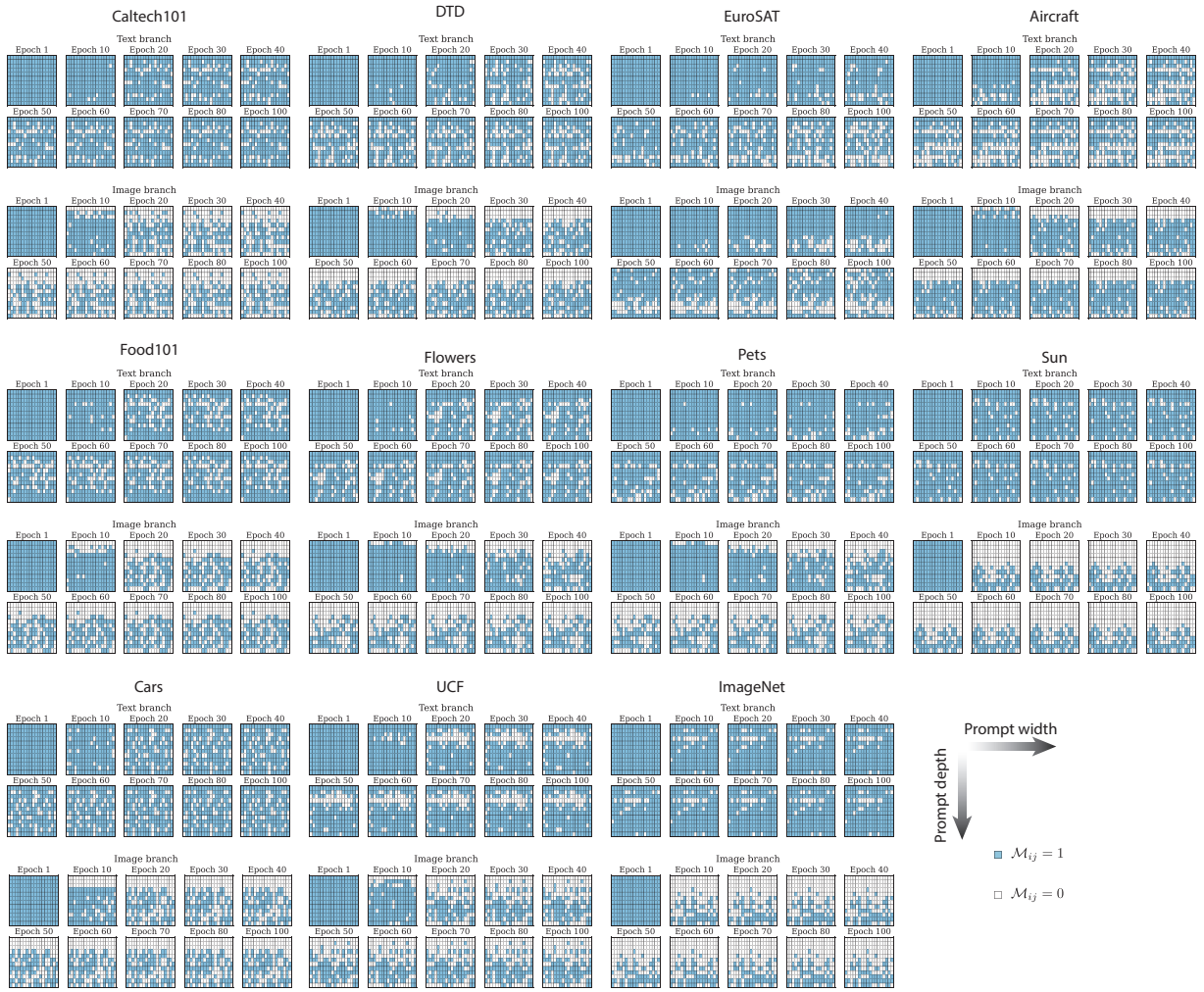

Figure 15: Pruning process of binary masks for the text and image branches for 11 datasets. $\mathcal{T}_{\text{target}} = 256$ and the warm-up epoch is 5. Owing to the warm-up process, $\mathcal{M}(t)$ at the epoch of 1 is same as the $\mathcal{M}(0)$, *i.e.* there is no pruning in inserted prompts.