Revisiting the Power of Prompt for Visual Tuning

Yuzhu Wang [1] **Lechao Cheng** [2 †] **Chaowei Fang** [3] **Dingwen Zhang** [4] **Manni Duan** [1] **Meng Wang** [2]

Abstract

Visual prompt tuning (VPT) is a promising solution incorporating learnable prompt tokens to customize pre-trained models for downstream tasks. However, VPT and its variants often encounter challenges like prompt initialization, prompt length, and subpar performance in self-supervised pretraining, hindering successful contextual adaptation. This study commences by exploring the correlation evolvement between prompts and patch tokens during proficient training. Inspired by the observation that the prompt tokens tend to share high mutual information with patch tokens, we propose initializing prompts with downstream token prototypes. The strategic initialization, a stand-in for the previous initialization, substantially improves performance. To refine further, we optimize token construction with a streamlined pipeline that maintains excellent performance with almost no increase in computational expenses compared to VPT. Exhaustive experiments show our proposed approach outperforms existing methods by a remarkable margin. For instance, after MAE pre-training, our method improves accuracy by up to 10%∼30% compared to VPT, and outperforms Full fine-tuning 19 out of 24 cases while using less than 0.4% of learnable parameters. Besides, the experimental results demonstrate the proposed SPT is robust to prompt lengths and scales well with model capacity and training data size. We finally provide an insightful exploration into the amount of target data facilitating the adaptation of pre-trained models to downstream tasks. The code is available at https://github.com/WangYZ1608/Self-Prompt-Tuning.

[1]Zhejiang Lab [2]School of Computer Science and Information Engineering, Hefei University of Technology [3]School of Artificial Intelligence, Xidian University [4]School of Automation, Northwestern Polytechnical University. Correspondence to: Lechao Cheng <chenglc@hfut.edu.cn>.

Proceedings of the 41st International Conference on Machine Learning, Vienna, Austria. PMLR 235, 2024. Copyright 2024 by the author(s).

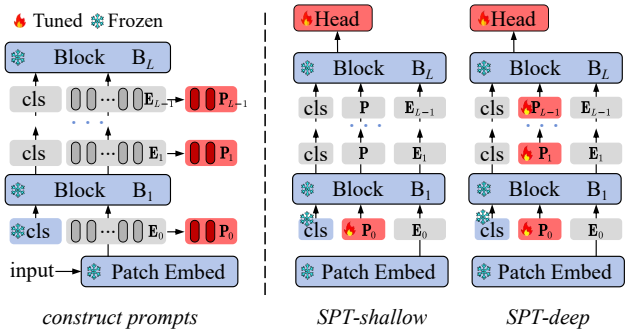


Figure 1. **Self-Prompt Tuning. Left**: We input a batch of the training data from the downstream task into the pre-trained model to get the forward patch embeddings. **Right**: We initialize prompts with sampled patch embeddings. Similar to VPT, we proposed **SPT-Shallow** and **SPT-Deep** depending on the layers involved. Only the prompts and task head parameters are learnable during adaptation on downstream tasks while the transformer encoder is frozen.

1. Introduction

The prevailing practice targeting downstream-focused computer vision applications adheres to the paradigm of pre-training followed by fine-tuning. This involves initially obtaining/training a versatile, task-agnostic backbone using supervised or self-supervised instructions. The structure of this foundation model is subsequently updated and tailored to the downstream tasks. While full parameter fine-tuning can yield impressive performance (always marked as the performance ceiling for downstream task tuning), it suffers from heavy computational burden that necessitates updating all model parameters and maintaining a distinct copy for each task when serving online. These issues become exacerbated particularly for modern model architectures with ever-expanding capacity, such as ViT-H (632M paramters (Dosovitskiy et al., 2020)) and ViT-22B (Dehghani et al., 2023), rendering it often impractical in real scenarios. As an alternative to trade off computational cost and performance, parameter-efficient fine-tuning typically brings part of tunable modules (Zhai et al., 2019; Houlsby et al., 2019; Lian et al., 2022; Jia et al., 2022) which usually have fewer parameters for task adaptation. Among these approaches, visual prompt tuning (VPT (Jia et al., 2022)) is gaining momentum as a promising solution that learns task-specific tokens while keeping the pretrained foundation model frozen during the tuning phase. Recent studies investigating VPT (Jia et al.,

2022) and its variants (Yoo et al., 2023; Cheng et al., 2023; Tu et al., 2023) unveil consistent challenges posed by the construction of prompt tokens[1], shedding light on the intricacies involved. We summarize the main challenges as follows.

Prompt initialization: Existing prompt-based methods, such as VPT (Jia et al., 2022), GateVPT (Yoo et al., 2023), employ the strategy of initializing prompts randomly (e.g., uniform or normal) and then update prompts during tuning, akin to optimizing the parameters of conventional neural networks. Nevertheless, the distinct initialization techniques for inserted prompt tokens significantly impact accuracy, as shown in the original paper of VPT and its variants.

Prompt length: The only extra hyperparameter that requires tweaking, in comparison to full fine-tuning, is the number of inserted prompt tokens. While ablation studies of existing approaches showcase that VPT and its variants are usually sensitive to the number of inserted prompts.

Subpar performance with self-supervised pretraining: Recent works (Yoo et al., 2023) have proven that the performance of VPT and its variants under self-supervised pretraining is significantly worse than its performance under supervised pretraining, hindering its application in various scenarios with massive unlabeled data.

To investigate these challenges, we employ a straightforward approach by conducting simple exploratory experiments to observe the distributional relationship, Masked Autoencoder (MAE (He et al., 2022)) pre-trained ViT-B as the backbone. As illustrated in Fig. 2, we compute the normalized mutual information between prompt tokens and patch tokens in each layer. Clearly, as the training proficiency progresses, the prompt tokens for downstream contextualization gradually converge towards the distribution of patch tokens, manifesting specifically as an increase in mutual information.

In light of the observation, we hypothesize that if the prompt tokens share sufficient information with patch tokens during the initial stage, the visual prompt tuning converges more rapidly and achieves superior results on the target task. To this end, we leverage the downstream inferred token prototypes as the initial prompts. Specifically, we first fed the training images into the pre-trained backbone for the target task to generate a collection of inferred tokens and cluster them together based on the corresponding prompt length. Subsequently, we initialize the task-specific learnable prompt tokens with those prototypes (Fig. 1). The results on the benchmark datasets demonstrate dramatic improvements with our proposed approach (Tab. 2). However, the time cost associated with the clustering of tokens is extraordinarily substantial, bordering on intolerable. To further optimize the procedure, we propose to sample the inferred tokens as the initial ones based on several sampling strategies, e.g. mean pooling, max pooling, and random sampling, achieving compelling performance with nearly no extra cost (Tab. 1).

The proposed SPT establishes a solid training start and is robust to prompt length, striking a trade-off between performance and computational cost. It positions itself as a robust complement to VPT for the adaptation of downstream tasks. We summarize the contributions as follows:

- We propose a simple yet effective approach termed **S**elf-**P**rompt **T**uning (SPT) that leverages the downstream inferred token prototypes as the initial prompts inspired by high mutual information. Additionally, we accelerate the clustering procedure by randomly sampling inferred tokens, which results in significant improvements with minimal additional computational cost compared to naive VPT. The proposed SPT is robust to prompt length and scales well with model capacity and training data size.
- For the first time, we showcase that the proposed SPT can achieve highly accurate results and can even compete with full fine-tuning. For instance, after MAE pre-training, our method improves average accuracy by up to 10% $\sim$ 30% relative to VPT, and outperforms full fine-tuning in 19 out of 24 cases while using less than 0.4% of learnable parameters.

2. Related Work

Pre-training and Full Fine-tuning. Pioneered by the seminal work of ImageNet (Deng et al., 2009), the methodology for image classification and various other computer vision tasks typically involves a pre-training followed by fine-tuning approach. In this paradigm, a general purpose, task-agnostic backbone undergoes pre-training (Krizhevsky et al., 2012; Dosovitskiy et al., 2020) through supervised or self-supervised (He et al., 2020) methods. Subsequently, the structure of this pre-trained backbone is modified and adapted to specific downstream tasks. Recent studies show that self-supervised pre-training (He et al., 2020; Grill et al., 2020; Chen et al., 2021; 2020; He et al., 2022) in downstream tasks outperforms supervised (Goyal et al., 2017; He et al., 2019) or weakly supervised (Kolesnikov et al., 2020) pre-training in full fine-tuning and shows promising scaling behavior. However, the results are always suboptimal (Jia et al., 2022; Yoo et al., 2023) when performing visual prompt tuning on self-supervised pre-trained models. The proposed method in this work improves the accuracy of VPT, especially for self-supervised pre-trained models, thereby effectively mitigating the limitation of VPT.

[1] In this paper, we refer to the inserted learnable parameters as *prompt tokens*, and the input images and the intermediate features as *patch tokens*.

Parameter-Efficient Fine-Tuning. *Full fine-tuning* is widely used to tailor the pre-trained foundation model to the downstream task. It is a straightforward approach that can often deliver impressive results, but it comes at the expense of updating all parameters and necessitates storing a distinct fine-tuned model for each task. As an alternative, *linear probing* focuses solely on training and storing new task heads, keeping the backbone frozen. However, its performance tends to lag behind that of full parameter fine-tuning. To strike a better balance between computational cost (i.e., training time and memory usage) and performance, researchers are exploring parameter-efficient fine-tuning (PEFT) algorithms. For example, SpotTune (Guo et al., 2019) investigates which layers require fine-tuning, while Bitfit (Zaken et al., 2021) updates only the bias term. SSF (Lian et al., 2022) proposes scaling and shifting deep features, and some works advocate inserting adapters (Houlsby et al., 2019; Chen et al., 2022) into the network. Unlike these backbone tuning methods, VPT (Jia et al., 2022) draws inspiration from prompt tuning (Li & Liang, 2021; Lester et al., 2021; Liu et al., 2022) in NLP to introduce a set of learnable prompt tokens into the input space and optimize them while keeping the backbone frozen, leading to significant computational savings. Surprisingly, VPT achieves comparable results even on par with full fine-tuning. However, existing studies (Yoo et al., 2023) also demonstrate that VPT and its variants suffer from some challenges, as discussed above. This work attempts to initialize the inserted prompt tokens with high mutual information, and the results demonstrate that the proposed approach can alleviate those issues. We believe that the proposed technique can be viewed as a substantial complement to the original VPT framework.

3. Methods

Within this part, we first revisit the standard visual prompt tuning (VPT (Jia et al., 2022)). Next, we conduct a simple exploratory experiment to investigate the connections between prompt tokens and patch tokens as the training proficiency progresses. Upon careful inspection, we finally detail the proposed approach and optimize the process.

3.1. Visual Prompt Tuning

Over the past few years, Vision Transformers (ViT) (Dosovitskiy et al., 2020) have been established as a powerful backbone for visual recognition. ViT usually consists of a patch embedding layer, a stack of L transformer blocks, and a classification head. The patch embedding layer divides the input image $\mathbf{x}$ into a set of non-overlapping patches. Each patch is then embedded into D-dimensional latent space coupled with the position information, as follows:

$$\mathbf{E}_0 = \mathrm{PatchEmbed}(\mathbf{x}) + \mathbf{E}_{pos} \tag{1}$$

The image patch embeddings, $\mathbf{E}_i = \{\mathbf{e}_i^j \in \mathbb{R}^D | j \in \mathbb{N}, 1 \leq j \leq N_e\}$, together with an extra learnable class token ([CLS]), as inputs to the (i+1)-th Transformer block (B_{i+1}), where each block consists of a multi-head self-attention (Vaswani et al., 2017) followed by a feed-forward layer with layer normalization (Ba et al., 2016) and residual connection (He et al., 2016). The whole ViT is formulated as:

$$[\mathbf{x}_i, \mathbf{E}_i] = \mathrm{B}_i([\mathbf{x}_{i-1}, \mathbf{E}_{i-1}]), i = 1, 2, \cdots, L \tag{2}$$

$$\mathbf{y} = \mathrm{Head}(\mathbf{x}_L) \tag{3}$$

where $\mathbf{x}_i \in \mathbb{R}^D$ denotes [CLS]'s embedding at (i+1)-th block input. $[\cdot, \cdot]$ indicates stacking and concatenation on the sequence length dimension. The task head is used to map the last block's [CLS] embedding, $\mathbf{x}_L$, into a predicted class probability distribution $\mathbf{y}$.[2]

VPT-Shallow In VPT framework proposed by Jia et.al (Jia et al., 2022), only a set of N_p learnable prompt tokens, denoted as $\mathbf{P}_0 = \{\mathbf{p}_0^k \in \mathbb{R}^D | k \in \mathbb{N}, 1 \leq k \leq N_p\}$, is introduced in the first transformer block. The VPT-Shallow is formulated as:

$$[\mathbf{x}_1, \mathbf{Z}_1, \mathbf{E}_1] = \mathrm{B}_1([\mathbf{x}_0, \mathbf{P}_0, \mathbf{E}_0]) \tag{4}$$

$$[\mathbf{x}_i, \mathbf{Z}_i, \mathbf{E}_i] = \mathrm{B}_i([\mathbf{x}_{i-1}, \mathbf{Z}_{i-1}, \mathbf{E}_{i-1}]), i = 2, \cdots, L \tag{5}$$

VPT-Deep For a more complicated setting, we insert N_p trainable prompt tokens $\mathbf{P}_{i-1} = \{\mathbf{p}_{i-1}^k \in \mathbb{R}^D | k \in \mathbb{N}, 1 \leq k \leq N_p\}$ into i-th transformer block[3]. The VPT-Deep is defined as:

$$[\mathbf{x}_i, _, \mathbf{E}_i] = \mathrm{B}_i([\mathbf{x}_{i-1}, \mathbf{P}_{i-1}, \mathbf{E}_{i-1}]), i = 1, \cdots, L \tag{6}$$

Discussion. VPT showcases considerable potential in downstream contextualization, leveraging large-scale pre-trained models. Deviating from conventional full parameter fine-tuning, VPT efficiently addresses the significant computational cost (updating less than 1% of parameters) and storage overhead (storing a single large foundation model and a small set of tokens coupled with a task-specific head for each task). It notably achieves competitive performance, retains the generalization capabilities of the foundation model, and alleviates overfitting within a certain amount of data. However, current advancements highlight substantial fragility when applying VPT to downstream tasks with different settings for prompt initialization and length. Also, the performance of the more frequently used self-supervised pre-trained models appears not to be as good as we expected.

[2]Masked Autoencoder (MAE) pretraining does not use [CLS] token. We follow the original designs and treat global average pooled image embedding as input for the task head.

[3]For convenience, we set the inserted prompt length to N_p in each layer for better concatenation. If we were to devise tokens with well-crafted dynamic prompt lengths, it is worth noting that this is often non-trivial and falls beyond the scope of this study.

3.2. High Normalized Mutual Information Matters

The central concern has pivoted towards optimizing the training of inserted prompt tokens. We start by conducting a simple experiment to observe how the distribution of prompt tokens changes as VPT converges. An intuitive strategy involves utilizing patch tokens as a reference. Consequently, we introduce **N**ormalized **M**utual **I**nformation (NMI (Estévez et al., 2009)) as a metric to quantify the relationship. Let $\pi(\mathbf{p}_{i-1}, \mathbf{e}_{i-1})$ be the joint distribution of prompt and patch tokens for the i-th transformer block. We approximate π with sigmoid-normalized cross-attention over $\mathbf{P}_{i-1}$ and $\mathbf{E}_{i-1}$ as:

$$\pi(\mathbf{p}_{i-1}^k, \mathbf{e}_{i-1}^j) := \frac{\sigma(\mathbf{p}_{i-1}^k \cdot \mathbf{e}_{i-1}^j)}{\sum_{j,k} \sigma(\mathbf{p}_{i-1}^k \cdot \mathbf{e}_{i-1}^j)} \tag{7}$$

where σ is the sigmoid operation. We further collapse along j to derive the marginal distribution of $\mathbf{p}_{i-1}$ as $\pi(\mathbf{p}_{i-1}) = \sum_{j=1}^{N_e} \pi(\mathbf{p}_{i-1}^k, \mathbf{e}_{i-1}^j)$. The marginal distribution of $\mathbf{e}_{i-1}$ follows the same principle as $\pi(\mathbf{e}_{i-1}) = \sum_{k=1}^{N_p} \pi(\mathbf{p}_{i-1}^k, \mathbf{e}_{i-1}^j)$. Thus, the normalized mutual information is defined as:

$$NMI(\mathbf{P}_{i-1}; \mathbf{E}_{i-1}) = \frac{2 * I(\mathbf{P}_{i-1}; \mathbf{E}_{i-1})}{H(\mathbf{P}_{i-1}) + H(\mathbf{E}_{i-1})} \tag{8}$$

where $I(;)$ indicates the standard mutual information, and $H(.)$ is the entropy. We can easily compute them with the joint distribution $\pi(\mathbf{p}_{i-1}^k, \mathbf{e}_{i-1}^j)$ and marginal distributions.

To explore the degree of attention collapse for prompt tokens, we measure the normalized mutual information for four canonical datasets (CUB-200-2011 (Wah et al., 2011), Caltech-101 (Wah et al., 2011), Patch Camelyon (Veeling et al., 2018), Clevrcount (Johnson et al., 2017)). As can be found in Fig. 2, the training procedure of visual prompt tuning (e.g., blue curves in four subfigures) gradually leads to high normalized mutual information values. Considering the definition of NMI itself, we conclude our discovery as follows:

Discovery. *The distribution of prompts for downstream contextualization gradually converges towards the distribution of patch tokens as the training proficiency progresses, manifesting specifically as increasing values in normalized mutual information.*

Recall that we discussed the benefits and challenges of applying visual prompt tuning in Sec. 3.1. Given the discovery above, we conjecture that initial prompt tokens with highly shared information (high NMI values at the training start point) may benefit the tuning procedure (improved convergence speed and accuracy), fostering more stable training. The problem is then shifted to constructing prompt initialization with high normalized mutual information.

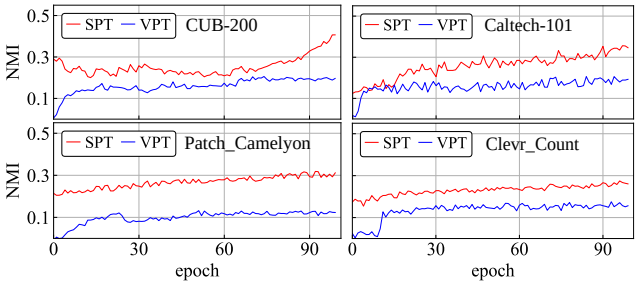


Figure 2. VPT presents the behavior of the **N**ormalized **M**utual **I**nformation (NMI (Estévez et al., 2009)) between prompts and patch tokens gradually increases during fine-tuning time. SPT has large NMI at the beginning, which will facilitate rapid convergence and achieve more advanced results.

3.3. Self-Prompt Tuning based on Token Prototypes

The intuitive approach to building prompts with highly shared information is to initialize prompt tokens with inferred token prototypes on the target dataset. We denote the forward results of patch embeddings in layer i as $\mathbf{E}^*{}_i = \{\mathbf{e}_i^j \in \mathbb{R}^D | j \in \mathbb{N}, 1 \leq j \leq N_e * N\}$ for the target dataset. N is the total number of the entire target dataset. Supposing $\mathbf{C}_i = \{\mathbf{c}_i^j \in \mathbb{R}^D | j \in \mathbb{N}, 1 \leq j \leq N_p\}$ are prototypes for layer i, we cluster $\mathbf{E}^*{}_i$ into N_p prototypes with K-means by minimizing the inertia as:

$$\arg\min_{\mathbb{S}_i} \sum_{j=1}^{N_p} \sum_{\mathbf{x}^j \in \mathbb{S}_i^j} ||\mathbf{x}^j - \mathbf{c}_i^j||^2 \tag{9}$$

$$\mathbf{c}_i^j = \frac{1}{\|\mathbb{S}_i^j\|} \sum_{\mathbf{x}^j \in \mathbb{S}_i^j} \mathbf{x}^j \tag{10}$$

where $\mathbb{S}_i$ are cluster sets and are initially partitioned uniformly with $\mathbf{E}^*{}_i$. $\mathbf{x}^j$ means the patch embedding issued into the j-th cluster set $\mathbb{S}_i^j$.

Similar to visual prompt tuning, we construct the prompt tokens with the prototypes $\mathbf{C}_i$. We term our approach **S**elf-**P**rompt **Tuning** (**SPT**). Following the example of VPT, we define **SPT-Shallow** and **SPT-Deep** depending on the number of transformer blocks involved.

SPT-Shallow We only initialize N_p learnable prompt tokens with $\mathbf{C}_0 = \{\mathbf{c}_0^k \in \mathbb{R}^D | k \in \mathbb{N}, 1 \leq k \leq N_p\}$ in the first transformer block. The SPT-Shallow is formulated as:

$$[\mathbf{x}_1, \mathbf{Z}_1, \mathbf{E}_1] = \mathrm{B}_1([\mathbf{x}_0, \mathbf{C}_0, \mathbf{E}_0]) \tag{11}$$

$$[\mathbf{x}_i, \mathbf{Z}_i, \mathbf{E}_i] = \mathrm{B}_i([\mathbf{x}_{i-1}, \mathbf{Z}_{i-1}, \mathbf{E}_{i-1}]), i = 2, \cdots, L \tag{12}$$

SPT-Deep For the input of i-th transformer block, we initialize N_p inserted prompt tokens with corresponding prototypes $\mathbf{C}_{i-1} = \{\mathbf{c}_{i-1}^k \in \mathbb{R}^D | k \in \mathbb{N}, 1 \leq k \leq N_p\}$. The

SPT-Deep[4] is defined as:

$$[\mathbf{x}_i, _, \mathbf{E}_i] = \mathrm{B}_i([\mathbf{x}_{i-1}, \mathbf{C}_{i-1}, \mathbf{E}_{i-1}]), i = 1, \cdots, L \quad (13)$$

We further compute the NMI based on eq. 7 and eq. 8. The red curves in Fig. 2 illustrate the trend of NMI over epoch for SPT. SPT starts with large NMI values as expected, facilitating rapid convergence and convincing results. During adaptation training on downstream tasks (use the whole training data), only the parameters of prompts and task head are learnable, while the Transformer encoder is frozen. In addition, we found that after self-prompt initialization, even if the parameters of the prompts are frozen (which means that only the task head is learnable), SPT can still achieve satisfactory results (Tab. 4 (a)). That indicates we indeed obtain a very good starting point with SPT. More experiments can be found in the experimental section.

3.4. Optimization for Token Construction

The previously introduced SPT treats token prototypes as initial prompts, which could approximately represent the overall distribution of the training data and/or intermediate tokens. However, the number of candidate tokens is usually far more than the length of the prompts. The clustering process is extremely slow (see Tab. 1 and Fig. 3 (a)), and even the time cost significantly exceeds the time of model fine-tuning. To alleviate this issue, we propose two more practical strategies.

Mean (max) pooling We perform mean or max pooling on one randomly selected batch tokens $\mathbf{E'}^*{}_i = \{\mathbf{e}_i^j \in \mathbb{R}^D | j \in \mathbb{N}, 1 \leq j \leq N_e * B\}$ to output N_p tokens. For non-overlapping sampling, the window kernel size is $\lfloor \frac{N_e * B}{N_p} \rfloor$. $\lfloor . \rfloor$ means floor operation. B is the batch size.

Random sample For a streamlined alternative, we even randomly sample N_p tokens within a random batch. For example, we randomly select one forward batch result of patch embeddings in layer i as $\mathbf{E'}^*{}_i$. The final N_p initial tokens are uniformly sampled from $N_e * B$ tokens.

Remark The proposed refinement procedure is more practical for downstream tasks. We undertake a brief assessment of computational cost in two aspects.

(I) Parameters: Prompts are the only additional hyper-parameters that need to be tuned and stored for SPT compared to full parameter fine-tuning. We only store the learned prompts and task head for each downstream task and re-use the copy of the pretrained backbone, significantly reducing the storage cost. For instance, given a ViT-B with 86M parameters and embedding dimension $D = 768$, we set the length of learnable prompts to 100 and 20 for shallow and deep variants respectively, which yield additional $N_p \times D = 100 \times 768 = 0.0768$M, and $L \times N_p \times D = 12 \times 20 \times 768 = 0.18$M, amounting to only 0.089% and 0.21% of ViT-B parameters, respectively.

[4]For vision transformer, the positional order of tokens has no effect after positional encoding, i.e., $[\mathbf{x}_i, \mathbf{C}_i, \mathbf{E}_i]$ and $[\mathbf{C}_i, \mathbf{x}_i, \mathbf{E}_i]$ are mathematically equivalent.

Table 1. **The time cost of the process of constructing prompts**. We employ SPT-Deep over four strategies (K-Means, Max pooling, Mean pooling, Random sample) on CUB-200-2011 with ViT-B as the backbone. (Note: 'd' means days, while 's' means seconds.)

Strategies	Proportion of the training data to construct prompts				
	10%	30%	50%	70%	100%
K-means	0.10d	0.30d	1.70d	6.20d	27.30d
Mean pooling	4.83s	13.20s	21.80s	29.19s	42.63s
Max pooling	7.20s	19.93s	32.80s	45.11s	64.86s
Random sample	5.30s	14.65s	24.13s	32.50s	43.56s

(II) Time Cost: Although the token prototypes based approach achieves impressive performance, the time cost of clustering tokens is extremely high. For example, we even spend 27.3 days executing K-means when applying SPT-Deep with token prototypes on CUB-200-2011 (see Tab. 1). Nevertheless, the proposed mean pooling, max pooling, and random sample strategies only need about 42.63 seconds, 64.86 seconds, and 43.56 seconds, respectively. Surprisingly, those two sampling strategies can still achieve compelling results (see Tab. 4 for more details).

4. Experiments

4.1. Experiment Setup

Datasets and Backbones. Our experiments are carried out on two image classification benchmarks. All the hyperparameters are determined with cross-validation on `val` set.

FGVC contains 5 benchmarked Fine-Grained Visual Classification, including CUB-200-2011 (Wah et al., 2011), NABirds (Van Horn et al., 2015), Oxford Flowers(Nilsback & Zisserman, 2008), Stanford Dogs(Khosla et al., 2011) and Stanford Cars (Gebru et al., 2017). Following VPT (Jia et al., 2022), we randomly split the training set into `train` (90%) and `val` (10%).

VTAB-1K collects 19 benchmarked Visual Task Adaptation (Zhai et al., 2019), categorized into three groups: i) *Natural* contains natural images captured by standard cameras; ii) *Specialized* images that are curated by specialized equipment; iii) *Structured* which requires structural understanding such as 3D depth prediction. Each task of VTAB-1k contains 1000 training examples. Following (Zhai et al., 2019; Jia et al., 2022), we apply the 800-200 split of the train/val set.

Pretrained Backbones. We use the plain vision transformer (ViT-Base/Large/Huge (Dosovitskiy et al., 2020))

Table 2. **Comparison under self-supervised pre-trained backbones.** We compare SPT and previous methods on 5 FGVC and 19 VTAB-1K benchmarks over two types of self-supervised pre-training methods: MAE (He et al., 2022) and MoCo-v3 (Chen et al., 2021). In short, our method achieves gains of 10% to 30% in average accuracy compared to VPT, and even outperforms full fine-tuning in 19 out of 24 cases while using less than 0.4% of learnable parameters under MAE pre-training. Per-task results for VTAB-1K are presented in Appendix Table. S3.

Methods	FGVC						VTAB-1K		
	Mean Acc	CUB-200-2011	NABirds	Oxford Flowers	Stanford Dogs	Stanford Cars	*Natural* (7)	*Specialized* (4)	*Structured* (8)
ViT-B with MAE pretrained on ImageNet-1K									
Full	82.80	**80.55**	**77.87**	91.71	80.38	83.51	59.31	79.68	53.82
VPT-Shallow	57.84	42.15	57.43	69.15	77.07	43.38	39.96	69.65	27.50
VPT-Deep	72.02	68.33	65.22	80.05	78.83	67.67	36.02	60.61	26.57
GateVPT	73.39	70.56	67.26	78.55	78.90	71.70	47.61	76.86	36.80
SPT-Shallow (ours)	73.95	71.15	61.87	89.47	80.01	67.23	62.53	80.90	53.46
SPT-Deep (ours)	**83.26**	80.13	76.28	**93.07**	**82.23**	**84.61**	**67.19**	**83.15**	**59.23**
ViT-B with MoCo-V3 pretrained on ImageNet-1K									
Full	84.25	81.75	**78.14**	94.52	81.19	85.67	71.95	84.72	51.98
VPT-Shallow	79.26	79.05	72.92	90.47	81.97	71.91	67.34	82.26	37.55
VPT-Deep	83.12	82.67	75.99	94.41	83.33	79.18	70.27	83.04	42.38
GateVPT	83.00	82.86	76.02	93.71	83.37	79.02	74.84	83.38	49.10
SPT-Shallow (ours)	84.08	83.50	75.79	95.03	84.17	81.93	74.47	83.93	55.16
SPT-Deep (ours)	**86.00**	**84.47**	77.63	**96.10**	**85.84**	**85.98**	**76.20**	**84.95**	**58.36**

Table 3. **Comparisons with PEFT approaches** with ViT-B pretrained on supervised ImageNet-21K as backbone. We report the average performance (Mean Acc and Params) and per-task results. "Input" and "Backbone" indicate the tuning parameter scope of each method. SPT achieves competitive results with far fewer trainable parameters. More results are presented in the Appendix (Table. S3 and S4).

Methods	Input	Backbone	Params (M)	Mean Acc	CUB-200	NABirds	Flowers	Dogs	Cars
Full		✓	85.98	88.54	87.3	82.7	98.8	89.4	84.5
Linear			0.18	79.32	85.3	75.9	97.9	86.2	51.3
Bias (Zhai et al., 2019)		✓	0.28	88.41	88.4	84.2	98.8	**91.2**	79.4
Adapter(Houlsby et al., 2019)		✓	0.41	85.66	87.1	84.3	98.5	89.8	68.6
MP (Gao et al., 2023)		✓	1.20	89.38	89.3	84.9	99.6	89.5	83.6
SSF (Lian et al., 2022)		✓	0.39	90.72	89.5	85.7	99.6	89.6	89.2
SNF (Wang et al., 2023)		✓	0.25	90.74	90.2	87.4	99.7	89.5	86.9
VPT-shallow (Jia et al., 2022)	✓		0.25	84.62	86.7	78.8	98.4	90.7	68.7
VPT-Deep	✓		0.85	89.11	88.5	84.2	99.0	90.2	83.6
E^2VPT (Cheng et al., 2023)	✓		0.56	89.22	89.1	84.6	99.1	90.5	82.8
SPT-Shallow (ours)	✓		0.25	90.10	90.2	85.1	99.5	89.3	86.4
SPT-Deep (ours)	✓		0.36	**91.40**	**90.6**	**87.6**	**99.8**	89.8	**89.2**

as the pretraining backbones that involve fewer inductive biases, since the backbone may be trained effectively using large-scale data and/or self-supervision. Specifically, we initialize the backbone with supervised pre-trained on ImageNet-21K (Deng et al., 2009) or self-supervised pre-trained on ImageNet-1K without labels (Chen et al., 2021; He et al., 2022), to ensure the flexibility of our method.

Implementation Details. For the FGVC datasets, we process images with a randomly resize crop operation to 224 × 224 resolution and a random horizontal flip for data augmentation. For VTAB-1k, we directly resize images to 224 × 224 and don't adopt any other augmentations (Zhai et al., 2019; Jia et al., 2022). We employ the AdamW optimizer with a mini-batch size of 32 for a total of 100 epochs (with a linear warm up for the first 10 epochs), and cosine learning rate (Loshchilov & Hutter, 2016) schedule, which gradually decays the learning rate from its initial value to 1e-8. We report the average accuracy score on the test set within three runs.

4.2. Compared to Stat-of-the-Art Results

SPT achieves excellent results under SSL backbones. Here we compare SPT with full fine-tuning, VPT (Jia et al., 2022), and GateVPT (Yoo et al., 2023), over two different self-supervised pre-training approaches: MAE (He et al., 2022) and MoCo-v3 (Chen et al., 2021). GateVPT (Yoo et al., 2023) is a recent adaptation method specifically designed for self-supervised pre-training models. Tab. 2 reports the comparisons. For the first time, we showcase that a *prompt tuning* method can achieve such highly accurate

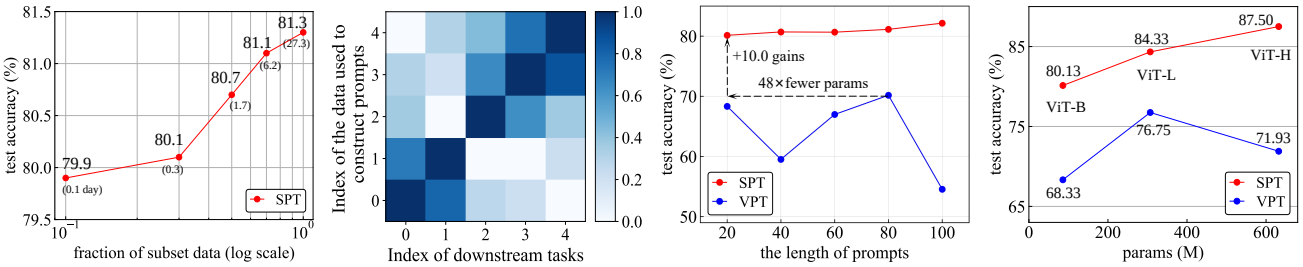
Table 4. **Ablation on prompt sampling strategies** with ViT-B as the backbone. We perform evaluation under two different pre-training strategies and two different types of downstream tasks: fine-grained (CUB-200, one of the FGVC benchmarks) and general natural images (Caltech-101, one of the VTAB-1K benchmarks), to explore the impact of sampling strategies.Experimental results (%) show that all prompt sampling strategies are significantly superior to the VPT baseline with random initialization (Jia et al., 2022).

(a) **The parameters of prompts are frozen** after initialization. Only the parameters of the task head are learnable during adaptation on downstream tasks.

prompt strategy	IN-21K, sup		IN-1K, MAE	
	CUB-200	Caltech-101	CUB-200	Caltech-101
Random init (VPT)	81.6	81.9	25.3	73.5
K-means	85.1 (+3.5)	87.8 (+5.9)	39.2 (+13.8)	84.1 (+10.6)
Mean pooling	83.5 (+1.9)	86.3 (+4.4)	37.7 (+12.4)	81.5 (+8.0)
Max pooling	84.0 (+2.4)	87.1 (+5.2)	37.0 (+11.7)	81.6 (+8.1)
Random sample	84.2 (+2.6)	86.9 (+5.0)	37.8 (+12.5)	82.2 (+8.7)

(b) During adaptation on downstream tasks, the parameters of prompts and task heads are learnable. This is a commonly used manner for visual prompt tuning.

prompt strategy	IN-21K, sup		IN-1K, MAE	
	CUB-200	Caltech-101	CUB-200	Caltech-101
Random init (VPT)	88.5	90.8	68.3	80.2
K-means	91.2 (+2.7)	92.7 (+1.9)	79.9 (+11.6)	89.1 (+8.9)
Mean pooling	90.4 (+1.9)	92.5 (+1.7)	79.8 (+11.5)	88.6 (+8.4)
Max pooling	90.6 (+2.1)	92.1 (+1.3)	79.6 (+11.3)	88.7 (+8.5)
Random sample	90.6 (+2.1)	92.6 (+1.8)	80.1 (+11.8)	88.9 (+8.7)



(a) k-means, acc *vs.* time (b) construct prompts across task (c) prompt length of each block (d) scale up model size

Figure 3. **Ablation studies on several basic components** using Masked Autoencoder (MAE) pre-trained backbone and SPT-deep evaluated on CUB-200-2011. **(a)** The k-means cluster strategy achieves further improvement with more data used to construct prompts. However, it introduces significant time costs during construct prompts. The wall-clock time is displayed in (·). **(b)** Prompts should be constructed using data from downstream tasks. **(c)** SPT is robust to prompt length changes and achieves slight gains with increasing prompt length. **(d)** SPT presents better scaling behavior than VPT with scaling up model size. These observations under supervised pre-trained backbone are similar (see the Appendix).

results and can even compete with full fine-tuning. Under MAE pre-training, our method improves average accuracy by up to $10\% \sim 30\%$ relative to VPT, and outperforms full fine-tuning in 19 out of 24 cases while using less than 0.4% of learnable parameters. Moreover, on those tasks with relatively poor performance, SPT is the only one that is even on par with full parameter tuning (e.g., 80.13% *vs.* 80.55% on CUB-200-2011 and 76.28% *vs.* 77.87% on NABirds). SPT has also achieved large gains against VPT under MoCo-v3 pre-training, e.g., +2.88% on FGVC *Mean Acc* and +15.98% on VTAB *Structured* group.

In addition, we note that MoCo-v3 pre-training presents better adaptability to downstream tasks than MAE pre-training. Improving the adaptability of MAE pre-training is an interesting future topic.

SPT is Comparable to PEFT Methods. Contrary to the scenario in self-supervised pre-training, these parameter-efficient fine-tuning techniques exhibit superior results compared to full fine-tuning in supervised pre-training. The current predominant approaches include optimizing the foundational backbones, such as SSF (Lian et al., 2022) and SNF (Wang et al., 2023). We compare our SPT with previous methods in Tab. 3 with backbone ViT-B, which is supervised pre-trained on ImageNet-21K. In short, our method, SPT-deep, achieves the best mean accuracy on FGVC benchmarks (91.40%). For more related experiments, please refer to Appendix Table. S4.

4.3. Ablation Study and Analysis

We perform ablation experiments on the FGVC and/or VTAB-1K dataset and compare them to the baseline that VPT (Jia et al., 2022) with random initialization. We set the prompt length to 100 and 20 for SPT-Shallow and SPT-Deep variants, respectively, with pre-trained ViT-B as the backbone.

The Impact of Different Sampling Strategies. We analyze the prompt sampling procedures when constructing prompts, as shown in Tab. 4. We assess the performance of SPT-deep using two distinct pre-training methods and two separate recognition tasks to explore the impact of the

prompt construction. In summary, our proposed approaches significantly outperform the existing strategy (random initialization in VPT).

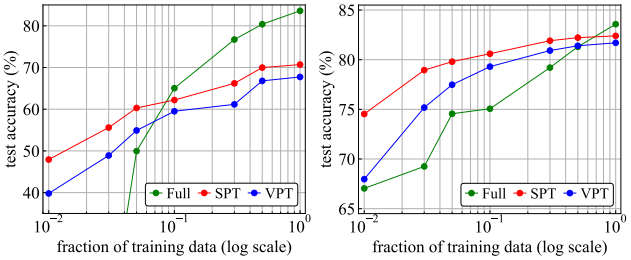
Tab. 4b provides a comparison between the k-means/mean pooling/max pooling/random sample strategies against the random initialization baseline. The setting is commonly used in prompt tuning that the prompts are optimized throughout the fine-tuning process. The values in the table demonstrate that not only the token prototypes can achieve good results, but also the proposed streamlined token construction approaches (Mean/Max pooling and Random Sample, more time cost can be found in Tab. 1) work well.

We also study an interesting setting in Tab. 4a, where the prompts are frozen after initialization. This means that only the task head is learnable during fine-tuning time (similar to linear probing). We found that our SPT-Deep can still achieve significant gains.

SPT is Robust to Prompt Length. Fig. 3 (c) investigates the influence of prompt length on accuracy. In this analysis, we vary the number of prompts inserted into each block for both SPT-Deep and VPT-Deep. Two key observations emerge from our study. Firstly, as the length of prompts increases, VPT exhibits fluctuating behavior. In contrast, our proposed method, SPT, demonstrates robustness to changes in prompt length and even yields slight performance gains as the prompt length increases. A notable gap of approximately 15% is observed between the optimal prompt length (i.e., $N_p = 80$) and the worst case scenario ($N_p = 100$) for VPT. Secondly, under the optimal prompt length, SPT achieves a significant improvement of +10% with $48\times$ fewer prompt parameters (ViT-B has 12 blocks) compared to VPT. We deduce that the superior performance of SPT can be attributed to the utilization of more appropriate prompts.

"Right Tool for the Right Job". In Fig.3(b), we investigate the utilization of cross-task data for prompt construction. Specifically, we designate one of the five FGVC benchmarks as the downstream task, while employing data from the other tasks to form prompts. The results within the same column in Fig.3 (b) are normalized for clarity. Ablation analysis reveals that prompts are constructed more effectively using data from the downstream task itself. Constructing prompts with data from other tasks results in suboptimal performance. We hypothesize that prompts and image embedding features may require an approximate distribution to adapt the features of downstream task data. This observation is intriguing and lays the groundwork for future research that aims to quantify the transferability of models across diverse datasets.

SPT Scales Well with Large Capacity. In Fig.3(d), we conduct a comparison between SPT and VPT by scaling up the pre-trained backbone size. In general, our method, SPT, exhibits superior scaling behavior. In contrast, VPT encounters degradation issues, as evident in the comparison between VPT with ViT-H and ViT-L. Notably, the significant advantages of SPT over VPT persist as the model scale increases. For example, SPT with ViT-H outperforms VPT with ViT-H by up to 15.5%.



(a) IN-1K, MAE (b) IN-21K, sup

Figure 4. **The impact of varying tuning data sizes with MAE and supervised pretraining.**

Insights for Downstream Tuning. In Tab. 2, our method, along with other parameter-efficient fine-tuning methods, demonstrates superior results compared to full fine-tuning. However, we emphasize that the task data used are relatively small, with only 1,000 training images in VTAB-1K, making full fine-tuning more susceptible to heavy overfitting. Here, we employ ImageNet-1K (approximately 1.28 million training images) as the downstream task and vary the number of available training images to investigate the choice of fine-tuning methods. The results are presented in Fig.4. Similar observations were made in iNaturalist, a long-tail recognition task, as illustrated in Appendix (Fig.S3). We can easily obtain some insights: **(I):** With MAE pre-training, SPT and VPT outperform full fine-tuning significantly in scenarios with less task data (e.g., less than 10%). However, the situation reverses with more task data. We infer that with sufficient task data, SPT, VPT, and other parameter-efficient fine-tuning methods cannot adequately capture data information with only a few learnable parameters (usually less than 1% of Full). A similar observation holds for supervised pre-training. **(II):** Supervised pre-training is a preferable choice when the available task data is small, despite the difference in pre-training data scale between MAE and supervised. Supervised pre-training can achieve satisfactory results with only minimal task data, e.g., 1%. We infer that the supervised pre-trained backbone already possesses recognition capabilities (knowledge), thereby reducing the reliance on downstream task data.

5. Discussion and Conclusion

Limitations. Our work has some visible limitations, e.g., we do not provide rigorous theoretical guarantee for the proposed initialization strategy and look forward to future research endeavors for this work.

Conclusion. Our work proposes a simple and novel method (termed **SPT**) to improve visual prompt tuning. Specifically, we observe that the mutual information between prompt tokens and image patch tokens gradually increases during the prompt tuning process of VPT, and propose the idea of constructing prompts based on inferred token prototypes. This idea can make the prompt tokens and patch tokens share larger mutual information at the beginning of training, which benefits the rapid convergence of the model and achieve higher accuracy. Our method delivers substantial performance improvements, especially for self-supervised pre-trained models. For instance, after MAE pretraining, our method improves accuracy by up to 10%$\sim$30% compared to VPT, and outperforms full fine-tuning 19 out of 24 cases while using less than 0.4% of learnable parameters. Significant performance improvements are also achieved for MoCo-V3 and supervised pre-training. We conduct sufficient ablation experiments and analyses. The results show that our method, SPT, exhibits better than VPT (a strong baseline), e.g., SPT is robust to prompt length and achieves slight gains with increasing prompt length; Also, SPT scales well with model capacity. These characteristics still work well with supervised pre-trained backbone.

Acknowledgements

The authors would like to thank the anonymous reviewers for their valuable comments and suggestions. This work was supported by the National Key R&D Program of China under Grant No. (2022YFF0608000, 2022ZD0119004), by the Fundamental Research Funds for the Central Universities, in part by the National Natural Science Foundation of China under Grant No. (72188101, 62106235, 62376206, U21B2048, 62322605, 62293543).

Impact Statement

This paper presents work whose goal is to advance the field of Machine Learning. There are many potential societal consequences of our work, none of which we feel must be specifically highlighted here.

References

Ba, J. L., Kiros, J. R., and Hinton, G. E. Layer normalization. *arXiv preprint arXiv:1607.06450*, 2016.

Beattie, C., Leibo, J. Z., Teplyashin, D., Ward, T., Wainwright, M., Küttler, H., Lefrancq, A., Green, S., Valdés, V., Sadik, A., et al. Deepmind lab. *arXiv preprint arXiv:1612.03801*, 2016.

Chen, S., Ge, C., Tong, Z., Wang, J., Song, Y., Wang, J., and Luo, P. Adaptformer: Adapting vision transformers for scalable visual recognition. *Advances in Neural Information Processing Systems*, 35:16664–16678, 2022.

Chen, T., Kornblith, S., Norouzi, M., and Hinton, G. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, pp. 1597–1607. PMLR, 2020.

Chen, X., Xie, S., and He, K. An empirical study of training self-supervised vision transformers. *arXiv preprint arXiv:2104.02057*, 2021.

Cheng, G., Han, J., and Lu, X. Remote sensing image scene classification: Benchmark and state of the art. *Proceedings of the IEEE*, 105(10):1865–1883, 2017.

Cheng, H., Qifan, W., Yiming, C., Zhiwen, C., Wenguan, W., Siyuan, Q., and Dongfang, L. E2vpt: An effective and efficient approach for visual prompt tuning. In *International Conference on Computer Vision (ICCV)*, 2023.

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

Dehghani, M., Djolonga, J., Mustafa, B., and et al. Scaling vision transformers to 22 billion parameters. *arXiv preprint arXiv:2302.05442*, 2023.

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

Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

Estévez, P. A., Tesmer, M., Perez, C. A., and Zurada, J. M. Normalized mutual information feature selection. *IEEE Transactions on neural networks*, 20(2):189–201, 2009.

Fei-Fei, L., Fergus, R., and Perona, P. One-shot learning of object categories. *IEEE transactions on pattern analysis and machine intelligence*, 28(4):594–611, 2006.

Gao, M., Wang, Q., Lin, Z., Zhu, P., Hu, Q., and Zhou, J. Tuning pre-trained model via moment probing. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 11803–11813, 2023.

Gebru, T., Krause, J., Wang, Y., Chen, D., Deng, J., and Fei-Fei, L. Fine-grained car detection for visual census estimation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 31, 2017.

Geiger, A., Lenz, P., Stiller, C., and Urtasun, R. Vision meets robotics: The kitti dataset. *The International Journal of Robotics Research*, 32(11):1231–1237, 2013.

Goyal, P., Dollár, P., Girshick, R., Noordhuis, P., Wesolowski, L., Kyrola, A., Tulloch, A., Jia, Y., and He, K. Accurate, large minibatch sgd: Training imagenet in 1 hour. *arXiv preprint arXiv:1706.02677*, 2017.

Graham, B. Kaggle diabetic retinopathy detection competition report. *University of Warwick*, 22, 2015.

Grill, J.-B., Strub, F., Altché, F., Tallec, C., Richemond, P., Buchatskaya, E., Doersch, C., Avila Pires, B., Guo, Z., Gheshlaghi Azar, M., et al. Bootstrap your own latent-a new approach to self-supervised learning. *Advances in neural information processing systems*, 33:21271–21284, 2020.

Guo, Y., Shi, H., Kumar, A., Grauman, K., Rosing, T., and Feris, R. Spottune: transfer learning through adaptive fine-tuning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 4805–4814, 2019.

He, K., Zhang, X., Ren, S., and Sun, J. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

He, K., Girshick, R., and Dollár, P. Rethinking imagenet pre-training. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 4918–4927, 2019.

He, K., Fan, H., Wu, Y., Xie, S., and Girshick, R. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 9729–9738, 2020.

He, K., Chen, X., Xie, S., Li, Y., Dollár, P., and Girshick, R. Masked autoencoders are scalable vision learners. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 16000–16009, 2022.

Helber, P., Bischke, B., Dengel, A., and Borth, D. 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.

Houlsby, N., Giurgiu, A., Jastrzebski, S., Morrone, B., De Laroussilhe, Q., Gesmundo, A., Attariyan, M., and Gelly, S. Parameter-efficient transfer learning for nlp. In *International Conference on Machine Learning*, pp. 2790–2799. PMLR, 2019.

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

Jia, M., Tang, L., Chen, B.-C., Cardie, C., Belongie, S., Hariharan, B., and Lim, S.-N. Visual prompt tuning. In *European Conference on Computer Vision*, pp. 709–727. Springer, 2022.

Johnson, J., Hariharan, B., Van Der Maaten, L., Fei-Fei, L., Lawrence Zitnick, C., and Girshick, R. Clevr: A diagnostic dataset for compositional language and elementary visual reasoning. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2901–2910, 2017.

Khosla, A., Jayadevaprakash, N., Yao, B., and Li, F.-F. Novel dataset for fine-grained image categorization: Stanford dogs. In *Proc. CVPR workshop on fine-grained visual categorization (FGVC)*, volume 2. Citeseer, 2011.

Kolesnikov, A., Beyer, L., Zhai, X., Puigcerver, J., Yung, J., Gelly, S., and Houlsby, N. Big transfer (bit): General visual representation learning. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part V 16*, pp. 491–507. Springer, 2020.

Kornblith, S., Norouzi, M., Lee, H., and Hinton, G. Similarity of neural network representations revisited. In *International conference on machine learning*, pp. 3519–3529. PMLR, 2019.

Krizhevsky, A., Hinton, G., et al. Learning multiple layers of features from tiny images. 2009.

Krizhevsky, A., Sutskever, I., and Hinton, G. E. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25, 2012.

LeCun, Y., Huang, F. J., and Bottou, L. Learning methods for generic object recognition with invariance to pose and lighting. In *Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2004. CVPR 2004.*, volume 2, pp. II–104. IEEE, 2004.

Lester, B., Al-Rfou, R., and Constant, N. The power of scale for parameter-efficient prompt tuning. *arXiv preprint arXiv:2104.08691*, 2021.

Li, X. L. and Liang, P. Prefix-tuning: Optimizing continuous prompts for generation. *arXiv preprint arXiv:2101.00190*, 2021.

Lian, D., Zhou, D., Feng, J., and Wang, X. Scaling & shifting your features: A new baseline for efficient model tuning. *Advances in Neural Information Processing Systems*, 35:109–123, 2022.

Liu, X., Ji, K., Fu, Y., Tam, W. L., Du, Z., Yang, Z., and Tang, J. P-tuning v2: Prompt tuning can be comparable to fine-tuning universally across scales and tasks. *arXiv preprint arXiv:2110.07602*, 2022.

Loshchilov, I. and Hutter, F. Sgdr: Stochastic gradient descent with warm restarts. *arXiv preprint arXiv:1608.03983*, 2016.

Matthey, L., Higgins, I., Hassabis, D., and Lerchner, A. dsprites: Disentanglement testing sprites dataset, 2017.

Netzer, Y., Wang, T., Coates, A., Bissacco, A., Wu, B., and Ng, A. Y. Reading digits in natural images with unsupervised feature learning. 2011.

Nilsback, M.-E. and Zisserman, A. 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.

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

Tu, C.-H., Mai, Z., and Chao, W.-L. Visual query tuning: Towards effective usage of intermediate representations for parameter and memory efficient transfer learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 7725–7735, 2023.

Van Horn, G., Branson, S., Farrell, R., Haber, S., Barry, J., Ipeirotis, P., Perona, P., and Belongie, S. Building a bird recognition app and large scale dataset with citizen scientists: The fine print in fine-grained dataset collection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 595–604, 2015.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., and Polosukhin, I. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

Veeling, B. S., Linmans, J., Winkens, J., Cohen, T., and Welling, M. Rotation equivariant cnns for digital pathology. In *Medical Image Computing and Computer Assisted Intervention–MICCAI 2018: 21st International Conference, Granada, Spain, September 16-20, 2018, Proceedings, Part II 11*, pp. 210–218. Springer, 2018.

Wah, C., Branson, S., Welinder, P., Perona, P., and Belongie, S. The caltech-ucsd birds-200-2011 dataset. 2011.

Wang, Y., Shi, B., Zhang, X., Li, J., Liu, Y., Dai, W., Li, C., Xiong, H., and Tian, Q. Adapting shortcut with normalizing flow: An efficient tuning framework for visual recognition. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 15965–15974. IEEE, 2023.

Xiao, J., Hays, J., Ehinger, K. A., Oliva, A., and Torralba, A. 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.

Yoo, S., Kim, E., Jung, D., Lee, J., and Yoon, S. Improving visual prompt tuning for self-supervised vision transformers. *arXiv preprint arXiv:2306.05067*, 2023.

Zaken, E. B., Ravfogel, S., and Goldberg, Y. Bitfit: Simple parameter-efficient fine-tuning for transformer-based masked language-models. *arXiv preprint arXiv:2106.10199*, 2021.

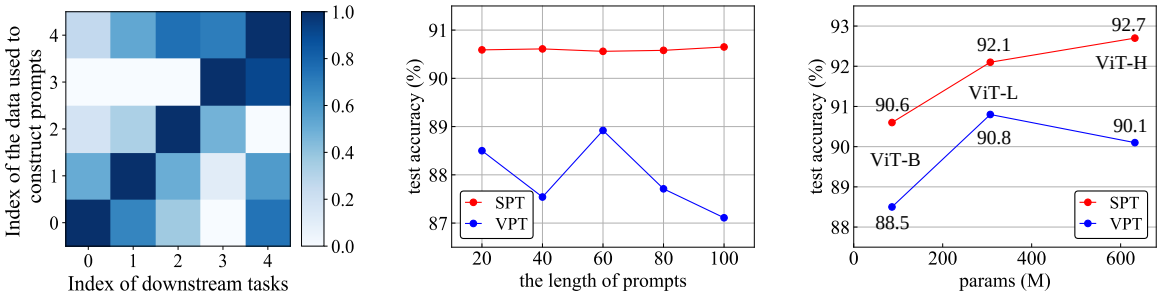
Zhai, X., Puigcerver, J., Kolesnikov, A., Ruyssen, P., Riquelme, C., Lucic, M., Djolonga, J., Pinto, A. S., Neumann, M., Dosovitskiy, A., et al. A large-scale study of representation learning with the visual task adaptation benchmark. *arXiv preprint arXiv:1910.04867*, 2019.

A. Detailed Descriptions for the Evaluation Datasets

We follow the practice of VPT (Jia et al., 2022) to perform the split of train/val/test. Tab. S1 summarizes the details of the evaluated datasets used in the paper.

Table S1. **Specifications of the downstream task datasets evaluated.** We randomly sampled the `train` and `val` sets since there are no public splits available.

Datasets	Description	# Classes	Train	Val	Test
	Fine-grained visual recognition tasks (FGVC)				
CUB-200-2011 (Wah et al., 2011)	Fine-grained bird species recognition	200	5,394*	600*	5,794
NABirds (Van Horn et al., 2015)	Fine-grained bird species recognition	555	21,536*	2,393*	24,633
Oxford Flowers (Nilsback & Zisserman, 2008)	Fine-grained flower species recognition	102	1,020	1,020	6,149
Stanford Dogs (Khosla et al., 2011)	Fine-grained dog species recognition	120	10,800*	1,200*	8,580
Stanford Cars (Gebru et al., 2017)	Fine-grained car classification	196	7,329*	815*	8,041
	Visual Task Adaptation Benchmark (VTAB-1k)				
Caltech101 (Fei-Fei et al., 2006)		102			6,084
CIFAR-100 (Krizhevsky et al., 2009)		100			10,000
DTD (Cimpoi et al., 2014)		47			1,880
Flowers102 (Nilsback & Zisserman, 2008)	Natural (7)	102	800 / 1000	200	6,149
Pets (Parkhi et al., 2012)		37			3,669
SVHN (Netzer et al., 2011)		10			26,032
Sun397 (Xiao et al., 2010)		397			21,750
Patch Camelyon (Veeling et al., 2018)		2			32,768
EuroSAT (Helber et al., 2019)	Specialized (4)	10	800 / 1000	200	5,400
Resisc45 (Cheng et al., 2017)		45			6,300
Retinopathy (Graham, 2015)		5			42,670
Clevr/count (Johnson et al., 2017)		8			15,000
Clevr/distance (Johnson et al., 2017)		6			15,000
DMLab (Beattie et al., 2016)		6			22,735
KITTI/distance (Geiger et al., 2013)	Structured (8)	4	800 / 1000	200	711
dSprites/location (Matthey et al., 2017)		16			73,728
dSprites/orientation (Matthey et al., 2017)		16			73,728
SmallNORB/azimuth (LeCun et al., 2004)		18			12,150
SmallNORB/elevation (LeCun et al., 2004)		9			12,150



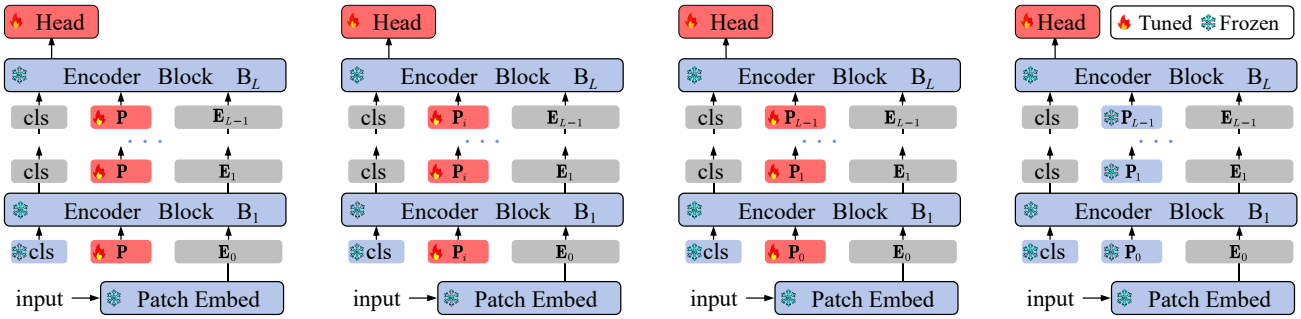
(a) construct prompts across task (b) prompt length of each block (c) scale up model size

Figure S1. **The IN-21K supervised** ViT-B counterpart of Fig. 3 on several major components.

B. Additional Results

B.1. Ablation with Supervised Pre-training

In Fig.S1, we present the counterpart of the ablation study conducted with ViT-B using supervised pre-training, complementing the analysis in Fig.3. The observations align with those obtained from MAE pre-training experiments. When compared to the baseline of VPT utilizing random initialization for prompts, SPT demonstrates superior behavior. Notably, SPT showcases robustness to prompt length variations and exhibits improved scaling behavior.



(a) random initialization (b) only use one-level's feature (c) SPT-deep (d) SPT-deep-frozen

Figure S2. **Prompt initialization strategies** with pretrained backbones. **(a)** Randomly initialize the parameters of prompts, which is adopted by VPT (Jia et al., 2022) and its variants (Yoo et al., 2023; Cheng et al., 2023; Tu et al., 2023). **(b)** Only use the first (i.e., $\mathbf{P}_i = \mathbf{P}_0$) or the last ($\mathbf{P}_i = \mathbf{P}_L$) level's output features to initialize all levels of prompts. **(c)** Use same-level image embedding features to construct prompts and insert the prompts into the input space of each block. We refer it as *self-prompt* strategy. **(d)** We found that after self-prompt initialization, even if the parameters of the prompt are frozen (which means that only the task head is learnable), our approach can still significantly improve the transfer performance (Table 4a). In all four cases, the encoders are frozen.

Table S2. **Ablation on prompt initialization strategies** with ViT-B as backbone evaluated on CUB-200-2011. The backbone is initialized with supervised pre-training on ImageNet-21K (left) and MAE (He et al., 2022) pre-trained on ImageNet-1K without labels (right). Two variants, namely *Shallow* and *Deep*, are depicted in Fig.1, with prompt lengths set to 100 and 20, respectively. Entries (a-c) correspond to Fig.S2 (a-c), with a comparison to the baseline represented by VPT with uniform initialization (Jia et al., 2022). The *SPT* strategy outperforms the baseline, particularly when using the MAE pre-trained backbone. For the shallow variant, *first* P_0 and *self-prompt* yield equivalent results.

		IN-21K, sup		IN-1K, MAE	
	prompt init.	Shallow	Deep	Shallow	Deep
(a)	random	86.7	88.5	42.2	68.3
(b)	first P_0	90.2 (+3.5)	89.2 (+0.7)	71.1 (+28.9)	76.5 (+8.2)
	last P_L	88.4 (+1.7)	89.9 (+1.4)	64.6 (+22.4)	76.8 (+8.5)
(c)	SPT (self-prompt)	**90.2** (+3.5)	**90.6** (+2.1)	**71.1** (+28.9)	**80.1** (+11.8)

B.2. Maximal Benefits via Layer-Specific Correspondence

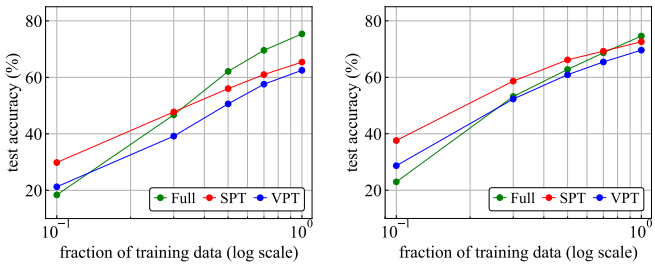
In Tab.S2, we present a comparison of prompt initialization strategies as illustrated in Fig.S2.

SPT demonstrates significant superiority over the baseline that randomly initialize the parameters of prompts, particularly in the case of Masked Autoencoder (MAE) pre-training. It increases accuracy by up to 28.9% and 11.8% for shallow and deep variants, respectively. It's important to note that prompts need to be inserted into the input space of the transformer block corresponding to the depth of SPT. To assess this design, we selectively use the output features of a specific level, such as the first or last layer's features, to construct prompts. These prompts are then inserted into all levels of the block, as depicted in Fig. S2 (b, c).

Tab. S2 (b, c) reveals that while using features from a single level to construct prompts achieves significant gains over the baseline with random initialization, it still lags notably behind the performance of SPT. We infer that the primary reason for this is the inconsistent semantic granularity between prompts and image embedding features at the same transformer depth.

B.3. Additional Results on Varying Training Data Size

In Sec 4.3, we employ ImageNet-1K, comprising approximately 1.28 million training images, as the downstream task. We systematically vary the quantity of available training images to investigate the optimal fine-tuning method selection, as illustrated in Fig. 4. Here, we extend our experiments to iNaturalist 2018, a dataset with approximately 437,000 training images distributed across 8,142 classes, showcasing a natural, long-tailed distribution. The results, presented in Fig. S3, mirror those of Fig. 4. Specifically, Self-Prompt Tuning (SPT) outperforms full fine-tuning when training data is limited, and the situation reverses when ample training data is available.



(a) MAE pre-training (b) Sup pre-training

Figure S3. **Varying available training data on iNaturelist 2018** and across two different pre-traing objectives.

B.4. Representation Similarity Analysis

We employ the Centered Kernel Alignment (CKA (Kornblith et al., 2019)) method to measure the representation similarity between prompts and image embeddings across various pre-training objectives. Our findings reveal that SPT consistently demonstrates a higher degree of similarity between prompts and image embeddings within the same layer, irrespective of whether the model is supervised or self-supervised pre-trained. Contrastingly, during self-supervised pre-training, VPT exhibits a relatively disordered phenomenon. Specifically, prompts from a certain layer in VPT may exhibit greater similarity to the image embeddings of other layers, rather than those of the same layer. This observed behavior could potentially be a primary factor contributing to the suboptimal performance of VPT under self-supervised pre-training.

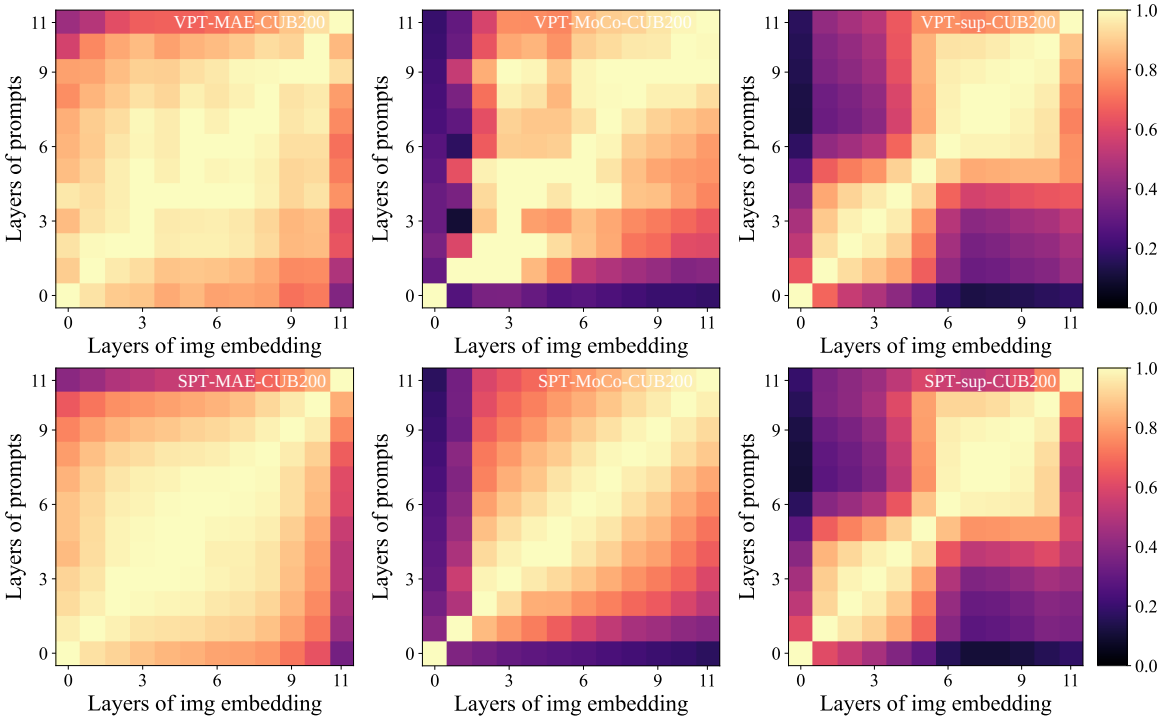


Figure S4. CKA similarities of prompts and patch tokens (i.e., image embedding features) for different layers and across different pre-training objectives.

C. Per-task Results on VTAB-1K and FGVC

Tab. S3 and S4 present pre-task results for 24 classification tasks evaluated in the main paper.

Table S3. Per-task fine-tuning results from Table. 2 for VTAB-1k benchmarks with pre-trained ViT-B/16 as backbone. The length of prompts is 100 and 20 for shallow and deep variants for all tasks respectively.

	Natural (7)							*Specialized* (4)				*Structured* (8)							
Methods	Caltech101	CIFAR-100	DTD	Flowers102	Pets	SVHN	Sun397	Patch Camelyon	EuroSAT	Resisc45	Retinopathy	Clevr/count	Clevr/distance	DMLab	KITTI/distance	dSprites/loc	dSprites/ori	SmallNORB/azi	SmallNORB/ele
ViT-B with MAE pretrained on ImageNet-1K																			
Full	84.2	24.6	56.9	72.7	74.4	**86.6**	15.8	81.8	**94.0**	72.3	70.6	67.0	59.8	45.2	75.3	72.5	47.5	**30.2**	33.0
SPT-Shallow (ours)	87.6	29.5	61.5	77.4	80.8	77.2	23.7	84.4	93.0	70.8	75.4	73.3	55.5	44.0	73.2	70.6	48.0	27.4	35.7
SPT-Deep (ours)	**88.9**	**37.7**	**64.1**	**84.5**	**83.7**	84.9	**26.5**	**85.2**	93.3	**78.5**	**75.6**	**76.8**	**63.1**	**47.9**	**76.7**	**82.3**	**49.3**	29.6	**48.1**
ViT-B with MoCo-V3 pretrained on ImageNet-1K																			
Full	91.0	57.6	64.6	91.6	79.9	**89.8**	29.1	85.1	**96.4**	83.1	74.2	55.2	56.9	44.6	**77.9**	63.8	**49.0**	**31.5**	36.9
SPT-Shallow (ours)	91.0	58.1	69.6	91.1	89.4	82.2	39.9	83.6	94.7	82.0	75.4	73.3	60.6	45.7	71.4	75.0	42.1	28.0	45.2
SPT-Deep (ours)	**92.2**	**62.3**	**70.1**	**93.2**	**89.4**	84.3	**41.9**	**85.6**	95.1	**83.3**	**75.8**	**76.1**	**62.5**	**49.0**	77.3	**77.7**	45.6	29.5	**49.2**
ViT-B with supervised pretrained on ImageNet-21K																			
Full	87.7	68.9	64.3	97.2	86.9	87.4	38.8	79.7	95.7	84.2	73.9	56.3	58.6	41.7	65.5	57.5	46.7	25.7	29.1
SPT-Shallow (ours)	91.2	78.9	71.2	99.4	90.7	86.4	**52.3**	82.5	94.9	82.6	74.6	68.6	61.2	48.0	68.8	72.6	41.9	24.4	37.4
SPT-Deep (ours)	**92.6**	**79.3**	**73.2**	**99.5**	**91.0**	**89.1**	51.2	**85.4**	**96.8**	**84.9**	**74.8**	**70.3**	**64.8**	**54.2**	**75.2**	**79.3**	**49.5**	**36.5**	**41.5**

Table S4. Per-task fine-tuning results for FGVC and VTAB-1K benchmarks with pre-trained ViT-B/16 as backbone. The length of prompts is 100 and 20 for shallow and deep variants for all tasks, respectively.

	Scope		FGVC (5)		VTAB-1K (19)			
Methods	Input	Backbone	Params (M)	Mean Acc	Params (M)	*Nature* (7)	*Specialized* (4)	*Structured* (8)
Full		✓	85.98	88.54	85.84	75.88	83.36	47.64
Linear			0.18	79.32	0.04	68.93	77.16	26.84
Bias (Zhai et al., 2019)		✓	0.28	88.41	0.14	73.30	78.25	44.09
Adapter (Houlsby et al., 2019)		✓	0.41	85.66	0.27	70.39	77.11	33.43
LoRA (Hu et al., 2021)		✓	-	-	0.32	79.49	84.55	**59.77**
SSF (Lian et al., 2022)		✓	0.39	90.72	0.24	81.57	**86.55**	58.96
SNF (Wang et al., 2023)		✓	0.25	90.74	0.29	**83.78**	86.13	59.61
VPT-shallow (Jia et al., 2022)	✓		0.25	84.62	0.11	76.81	79.68	46.98
VPT-Deep (Jia et al., 2022)	✓		0.85	89.11	0.64	78.48	82.43	54.98
E^2VPT (Cheng et al., 2023)	✓		0.56	89.22	0.31	80.01	84.43	57.39
SPT-Shallow (ours)	✓		0.25	90.10	0.11	81.44	83.65	52.86
SPT-Deep (ours)	✓		0.36	**91.40**	0.22	82.27	85.48	58.91