

Selective Visual Prompting in Vision Mamba

Yifeng Yao, Zichen Liu, Zhenyu Cui, Yuxin Peng, Jiahuan Zhou*

Wangxuan Institute of Computer Technology, Peking University, Beijing 100871, China
 {yaoyifeng, lzc20180720, cuizhenyu}@stu.pku.edu.cn, {pengyuxin, jiahuanzhou}@pku.edu.cn

Abstract

Pre-trained Vision Mamba (Vim) models have demonstrated exceptional performance across various computer vision tasks in a computationally efficient manner, attributed to their unique design of selective state space models. To further extend their applicability to diverse downstream vision tasks, Vim models can be adapted using the efficient fine-tuning technique known as visual prompting. However, existing visual prompting methods are predominantly tailored for Vision Transformer (ViT)-based models that leverage global attention, neglecting the distinctive sequential token-wise compression and propagation characteristics of Vim. Specifically, existing prompt tokens prefixed to the sequence are insufficient to effectively activate the input and forget gates across the entire sequence, hindering the extraction and propagation of discriminative information. To address this limitation, we introduce a novel Selective Visual Prompting (SVP) method specifically for the efficient fine-tuning of Vim. To prevent the loss of discriminative information during state space propagation, SVP employs lightweight selective prompters for token-wise prompt generation, ensuring adaptive activation of the update and forget gates within Mamba blocks to promote discriminative information propagation. Moreover, considering that Vim propagates both shared cross-layer information and specific inner-layer information, we further refine SVP with a dual-path structure: Cross-Prompting and Inner-Prompting. Cross-Prompting utilizes shared parameters across layers, while Inner-Prompting employs distinct parameters, promoting the propagation of both shared and specific information, respectively. Extensive experimental results on various large-scale benchmarks demonstrate that our proposed SVP significantly outperforms state-of-the-art methods.

Code —

<https://github.com/zhujiahuan1991/AAAI2025-SVP>

Introduction

Vision Mamba (Vim), a groundbreaking vision backbone featuring an input-dependent modeling mechanism termed selective state space, has demonstrated superior performance compared to established Vision Transformers (ViT) while maintaining computational efficiency (Zhu et al.

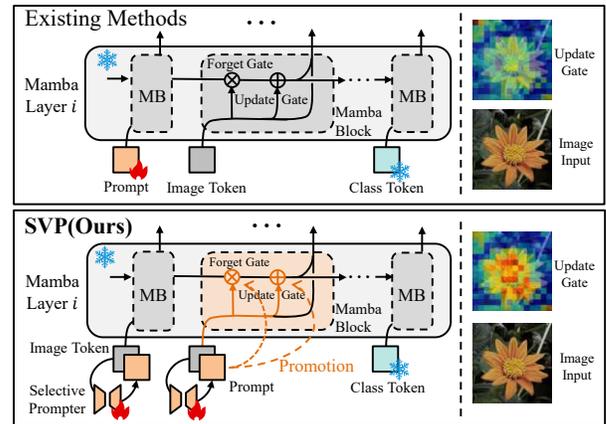


Figure 1: Existing visual prompting methods (Jia et al. 2022) use prompt sequences prefixed to the image tokens, which hinder discriminative feature propagation in Vim. In contrast, our SVP uses input-dependent selective prompts that better learn the input distribution, activating update and forget gates to enhance object-aware feature propagation.

2024). This advancement positions Vim as a potential next-generation foundation model architecture. However, as models scale up, directly applying pre-trained Vim models to downstream tasks through full fine-tuning leads to significant computational and storage overhead. This issue has spurred the development of Parameter-Efficient Fine-Tuning (PEFT) techniques (Fu et al. 2023), which adapt pre-trained models to downstream tasks by fine-tuning only a small subset of parameters or incorporating a few additional ones, thereby substantially reducing storage requirements. Among PEFT methods, visual prompting (Han et al. 2023) has shown promising performance by integrating a few additional learnable parameters into pre-trained models.

Therefore, visual prompting holds substantial potential for the efficient fine-tuning of Vim, enabling high performance with minimal resource consumption. However, current visual prompting methods are predominantly designed for ViT with global attention mechanisms and fail to account for the unique sequential characteristics of Vim, which processes visual information through token-by-token compression and propagation. As illustrated in Figure 1, existing ap-

*Corresponding Author

Copyright © 2025, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.

proaches employ prompt sequences prefixed to the token sequence, which inadequately adapts the model to the input distribution. This limitation poses a significant challenge, as it impedes the effective updating and propagation of discriminative sequence information when directly applied to Vim. There is a pressing need to explore visual prompting techniques specifically tailored to Vim, making it both a significant research endeavor and a practical necessity.

To address the unique requirements of Vim, we propose a novel Selective Visual Prompting (SVP) method designed to promote discriminative information propagation within the sequence. As depicted in Figure 1, our approach employs a lightweight prompter that dynamically produces token-level prompts based on varying inputs. These prompts are then integrated into the original image tokens, which adequately adapt the model to the input distribution. This mechanism ensures the selective activation of update and output gates, which are input-dependent parameters in Vim. Consequently, our SVP method enables the model to update and compress relevant discriminative features while propagating them through the network. Simultaneously, it identifies and discards irrelevant distracting information, preventing it from contaminating the compressed state of the sequence. This targeted approach enhances the model’s ability to retain and propagate discriminative information.

Furthermore, recognizing that Vim propagates both shared cross-layer information and specific inner-layer information, we introduce a dual-path structure in our selective prompting design, comprising Cross-Prompting and Inner-Prompting mechanisms. This design aims to optimize the propagation of both types of information within Vim. The Cross-Prompting facilitates the transfer of shared information between layers, while Inner-Prompting enhances the flow of layer-specific information within each layer. To account for the varying proportions of these two types of information at different layers, we implement an element-wise scaling factor that dynamically adjusts the emphasis between the two prompts. This dual-path structure ensures a balanced and effective leveraging of both information types, significantly enhancing the model’s overall performance.

To sum up, the main contributions of this work are: (1) To the best of our knowledge, this is the initial exploration of visual prompting within Vim. We introduce a selective prompting approach that leverages Vim’s input-dependent characteristics, adaptively activating its input and forget gates to enhance the propagation of discriminative information. (2) In our SVP, a dual-path prompting structure, termed Cross-Inner, is proposed to effectively utilize both cross-layer shared information and inner-layer specific features, ensuring comprehensive and efficient information propagation. (3) Extensive experiments on various datasets demonstrate that our SVP method significantly outperforms existing visual prompting approaches.

Related Work

State Space Model

The state space model (SSM) with linear complexity presented a promising approach for modeling long-range

dependencies. Moreover, the Structured State-Space Sequence model (Gu, Goel, and Ré 2021) improved computational efficiency while preserving theoretical strengths through novel parameterization. Expanding on this foundation, Mamba (Gu and Dao 2023) and Mamba2 (Dao and Gu 2024) introduced a data-dependent SSM layer with hidden state expansion, forming a language model backbone. Building on its success in sequence data, Vision Mamba (Zhu et al. 2024) applied pure Mamba layers to vision tasks, utilizing bidirectional scans for comprehensive modeling. Vim’s linear complexity and effective performance highlighted its suitability for a large pre-trained model. The previous paradigm for adapting pre-trained models to downstream tasks was full fine-tuning. However, as the scale of Vim models increased, this approach became inefficient, driving the development of Parameter-Efficient Fine-Tuning methods (PEFT) (Fu et al. 2023).

Parameter-Efficient Fine-Tuning

PEFT aimed at reducing learnable parameters while maintaining performance, categorized into partial-based, addition-based, and prompt-based methods (Xin et al. 2024b). Partial-based methods trained only select portions of model parameters, e.g., bias terms, attention, or MLP layers (Zaken, Ravfogel, and Goldberg 2021; Kornblith, Shlens, and Le 2019; Touvron et al. 2022; Basu et al. 2024). While these approaches are straightforward and simple to implement, they often lag in performance compared to full finetuning. Addition-based methods, such as Side-Tuning (Sung, Cho, and Bansal 2022) and adapters (Chen et al. 2022; Steitz and Roth 2024; Xin et al. 2024a; Dong et al. 2024), introduce auxiliary components for task-specific learning, yet their custom nature limits generalizability across different architectures.

Prompt Learning

Visual prompt learning techniques, which operated primarily on the input, offered better generalization and were more compatible with various models. Existing visual prompt learning methods could be categorized into two types. The first type, represented by the VPT series, appended prompt tokens to the image token sequence. E²VPT (Han et al. 2023) further refined these prompts by pruning ineffective ones, while more recent approaches like InsVP (Liu, Peng, and Zhou 2024) learned prompts more relevant to the instance. SPT (Wang et al. 2024) revisited the power of VPT and extended it by self-initializing with downstream token prototypes. However, these methods struggled to effectively capture the distribution across the entire sequence in the Vim sequence model. The second type directly overlaid frame-like prompts onto the original image, as seen in DAM-VP (Huang et al. 2023) and AutoVP (Tsao et al. 2024). These methods only applied prompts at the image level and lacked input dependency. As a result, they were not effective at activating the update and forget gates in the deeper layers of Vim. Therefore, directly applying these two types of prompt learning methods to Vim led to suboptimal performance. In contrast, our SVP method selectively activates

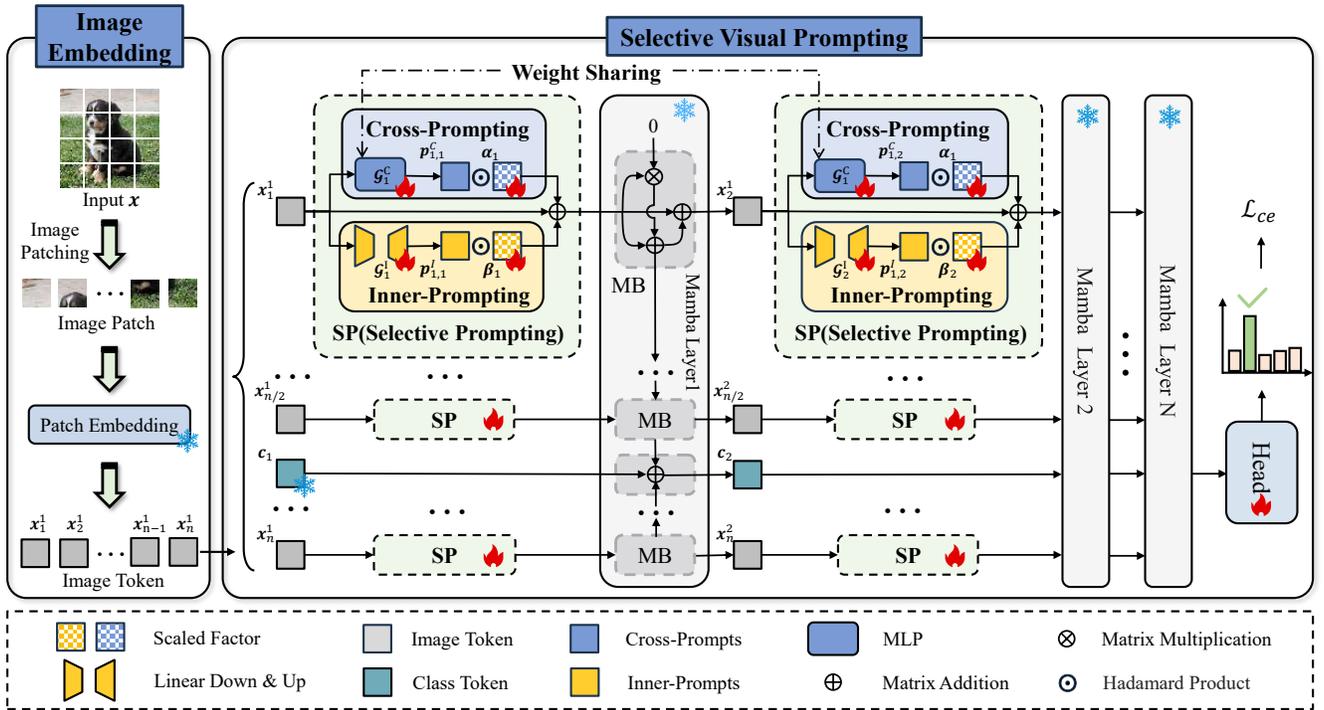


Figure 2: Our SVP employs a dual-path architecture: the Inner-Prompting pathway prompts specific information at each layer, while the Cross-Prompting pathway prompts shared information across layers. Both the Inner-Prompt p_i^c and Cross-Prompt p_i^c are selectively generated based on the input. They are subsequently coordinated by two element-wise dynamic factors α_j, β_j , and then superimposed onto the original input.

the update gate across the entire sequence, promoting discriminative information propagation.

The Proposed Method

SSM Preliminaries

The SSM-based models, Mamba (Gu and Dao 2023), and Vision Mamba (Vim) (Zhu et al. 2024) are inspired by the continuous system, which maps a one-dimensional function or sequence $x(t) \in \mathbb{R} \mapsto y(t) \in \mathbb{R}$ through a N-dimension hidden state $h(t) \in \mathbb{R}^N$. The hidden state evolves over time with parameters \mathbf{A} , \mathbf{B} , and \mathbf{C} , following linear ordinary differential equations:

$$\begin{aligned} h'(t) &= \mathbf{A}h(t) + \mathbf{B}x(t), \\ y(t) &= \mathbf{C}h(t), \end{aligned} \quad (1)$$

where $\mathbf{A} \in \mathbb{R}^{N \times N}$ is the state matrix, $\mathbf{B} \in \mathbb{R}^{N \times 1}$, and $\mathbf{C} \in \mathbb{R}^{1 \times N}$ are projection parameters.

To adapt SSM for deep learning, it is discretized using zero-order hold (Pechlivanidou and Karampetakis 2022). The continuous parameters \mathbf{A} , \mathbf{B} are transformed into their discrete counterparts $\bar{\mathbf{A}} \in \mathbb{R}^{N \times N}$, $\bar{\mathbf{B}} \in \mathbb{R}^{N \times 1}$ using a timescale parameter $\Delta \in \mathbb{R}$:

$$\begin{aligned} \bar{\mathbf{A}} &= \exp(\Delta \mathbf{A}), \\ \bar{\mathbf{B}} &= (\Delta \mathbf{A})^{-1}(\exp(\Delta \mathbf{A}) - \mathbf{I}) \cdot \Delta \mathbf{B} \approx \Delta \mathbf{B}. \end{aligned} \quad (2)$$

Thus, the discrete SSM can be written as:

$$\begin{aligned} h_i &= \bar{\mathbf{A}}h_{i-1} + \bar{\mathbf{B}}x_i, \\ y_i &= \mathbf{C}h_i, \end{aligned} \quad (3)$$

where $h_{i-1}, h_i \in \mathbb{R}^{N \times 1}, x_i$.

Selective Visual Prompting

To address the challenge of efficient fine-tuning in Vim, we propose a novel approach termed Selective Visual Prompting (SVP). Our method effectively adapts the model to the input distribution by selectively generating prompts at the token level. It adaptively activates the update and forget gates to promote effective information propagation. The overall structure of the proposed method is illustrated in Figure 2.

Specifically, the process begins with dividing the input image $\mathbf{x} \in \mathbb{R}^{H \times W \times C}$ into equally sized patches, where (H, W) represents the size of the image \mathbf{x} , and C is the number of channels. These patches are then embedded into d -dimensional latent space as $\{\mathbf{x}_i^p\}_{i=1}^n, \mathbf{x}_i \in \mathbb{R}^{1 \times d}$.

Given Vim’s 24-layer hierarchical architecture, each layer in Vim should focus not only on specific inner-layer features but also on shared features with adjacent layers. To facilitate both information propagation, we designed the lightweight Selective Prompting Module with a dual-path structure, incorporating Cross-Prompting and Inner-Prompting.

Cross-Prompting. To capture and propagate shared information across layers to ensure feature consistency, we de-

sign a Cross-Prompting Module. In this module, we utilize a fully connected cross-prompts generator (\mathcal{G}^C) with shared parameters across layers to generate cross-prompts $\mathbf{p}_i^C \in \mathbb{R}^{1 \times d}$ through $\mathbf{x}_i \in \mathbb{R}^{1 \times d}$. The number of layers sharing parameters is a hyperparameter set to 6, 8, or 12. The process is represented by the following formula:

$$\mathbf{p}_i^C = \mathcal{G}^C(\mathbf{x}_i). \quad (4)$$

Inner-Prompting. This module focuses on extracting and preserving layer-specific features to enhance the model’s discriminative power. To minimize tunable parameters while maintaining performance, we design a lightweight inner-prompts generator (\mathcal{G}^I) for each layer of Vim, enabling the generation of distinct inner-prompts $\mathbf{p}_i^I \in \mathbb{R}^{1 \times d}$. This generator includes a linear down layer (L^{down}), linear up layer (L^{up}) and a SiLU activation (Elfwing, Uchibe, and Doya 2018). The hidden dimension, which refers to both the reduced dimension in L^{down} and the dimension to be expanded in L^{up} , is set to 64. This process is represented by the following formula:

$$\begin{aligned} \mathbf{p}_i^I &= \mathcal{G}^I(\mathbf{x}_i) \\ &= \text{SiLU}(L^{up}(L^{down}(\mathbf{x}_i))). \end{aligned} \quad (5)$$

Otherwise, considering the differing importance of layer-specific features and shared information across layers, we designed two element-wise dynamic scaling factors (α , β) to balance the influence of these prompts on the input distribution.

$$\bar{\mathbf{p}}_i = \alpha \odot \mathbf{p}_i^C + \beta \odot \mathbf{p}_i^I, \quad (6)$$

where $\alpha \in \mathbb{R}^{1 \times d}$ and $\beta \in \mathbb{R}^{1 \times d}$ are learnable parameters which are initialized to zero, \odot denotes Hadamard product, $\bar{\mathbf{p}}_i \in \mathbb{R}^{1 \times d}$. As shown in Equation 7, the generated prompts are then overlaid onto the original input, effectively activating the update and forget gates in Vim to promote the propagation of shared and layer-specific information.

$$\mathbf{x}_i^p = \mathbf{x}_i + \bar{\mathbf{p}}_i. \quad (7)$$

Subsequently, all prompted image tokens $\{\mathbf{x}_i^p\}_{i=1}^n$ along with an additional classification token $\mathbf{c}_1 \in \mathbb{R}^{1 \times d}$ are fed into the N Mamba blocks $\{\mathcal{B}_j\}_{j=1}^N$ to extract features. Similar to the design of VPT-deep, we incorporate our selective prompts at the input of each layer. The output \mathbf{c}_{N+1} from the final Mamba block is then passed through a classification head \mathcal{H} to produce the predicted probability distribution \mathbf{y} .

Overall Optimization. As mentioned above, our SVP introduces only a few additional parameters:

$$\mathcal{M} = \{\mathcal{G}^C, \mathcal{G}^I, \alpha, \beta\}. \quad (8)$$

Following prior works (Jia et al. 2022; Huang et al. 2023; Wang et al. 2024), we keep the pre-trained model’s encoder frozen during training, allowing only the classification head \mathcal{H} and the newly added modules \mathcal{M} to be trainable. The optimization objective is defined as follows:

$$\arg \min_{\mathcal{M}, \mathcal{H}} \mathcal{L}_{ce}(\mathbf{y}, y_{gt}), \quad (9)$$

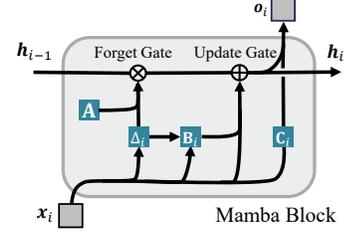


Figure 3: The internal structure of the Mamba block. The parameters Δ_i , \mathbf{B}_i , and \mathbf{C}_i are all input-dependent.

where \mathcal{L}_{ce} is the cross-entropy loss, and y_{gt} is the image label.

Discussion and Analysis

In this section, we discuss how our SVP facilitates the update and forget gates across the entire sequence, thereby promoting effective information propagation.

As shown in Figure 3, the Mamba architecture enhances the SSM by introducing the selective state space model. The parameters $\mathbf{B}_i \in \mathbb{R}^{h \times 1}$, $\mathbf{C}_i \in \mathbb{R}^{1 \times h}$, and $\Delta_i \in \mathbb{R}^{1 \times d}$ are generated from \mathbf{x}_i via functions \mathcal{S}_B , \mathcal{S}_C , and \mathcal{S}_Δ , thus becoming input-dependent:

$$\mathbf{B}_i = \mathcal{S}_B(\mathbf{x}_i), \mathbf{C}_i = \mathcal{S}_C(\mathbf{x}_i), \Delta_i = \mathcal{S}_\Delta(\mathbf{x}_i). \quad (10)$$

Mamba practically applies the state transition Equation 3 independently to each channel of input \mathbf{x}_i , leading to the following formulations:

$$\begin{aligned} \mathbf{h}_i &= \tilde{\mathbf{A}}_i \odot \mathbf{h}_{i-1} + \mathbf{B}_i(\Delta_i \odot \mathbf{x}_i) \\ &= \exp(\mathcal{S}_\Delta(\mathbf{x}_i) \tilde{\odot} \mathbf{A}) \odot \mathbf{h}_{i-1} + \mathcal{S}_B(\mathbf{x}_i)(\mathcal{S}_\Delta(\mathbf{x}_i) \odot \mathbf{x}_i), \end{aligned} \quad (11)$$

where \mathbf{A} , $\tilde{\mathbf{A}}_i$, \mathbf{h}_{i-1} , $\mathbf{h}_i \in \mathbb{R}^{h \times d}$, $\tilde{\odot}$ denotes extending the first dimension of the preceding matrix, followed by a Hadamard product with the subsequent matrix.

In our SVP, Vim’s parameters \mathbf{B}_i , \mathbf{C}_i and Δ_i are generated through prompted inputs \mathbf{x}_i^p as $\mathbf{B}_i^p \in \mathbb{R}^{h \times 1}$, $\mathbf{C}_i^p \in \mathbb{R}^{1 \times h}$ and $\Delta_i^p \in \mathbb{R}^{1 \times d}$. Then the state transition equation in Vim can be rewritten as:

$$\begin{aligned} \mathbf{h}_i &= \tilde{\mathbf{A}}_i^p \odot \mathbf{h}_{i-1} + \mathbf{B}_i^p(\Delta_i^p \odot \mathbf{x}_i^p) \\ &= \exp(\mathcal{S}_\Delta(\mathbf{x}_i + \bar{\mathbf{p}}_i) \tilde{\odot} \mathbf{A}) \odot \mathbf{h}_{i-1} \\ &\quad + \mathcal{S}_B(\mathbf{x}_i + \bar{\mathbf{p}}_i)(\mathcal{S}_\Delta(\mathbf{x}_i + \bar{\mathbf{p}}_i) \odot \mathbf{x}_i) \\ &\quad + \mathcal{S}_B(\mathbf{x}_i + \bar{\mathbf{p}}_i)(\mathcal{S}_\Delta(\mathbf{x}_i + \bar{\mathbf{p}}_i) \odot \bar{\mathbf{p}}_i). \end{aligned} \quad (12)$$

From Equations 11 and 12, our method directly activates the update gate ($\mathbf{B}_i(\Delta_i \odot \mathbf{x}_i)$) and forget gate ($\tilde{\mathbf{A}}_i$) in Vim, promoting the updation of discriminative information into the hidden state and its propagation across the sequence. This enhances the model’s adaptability to new tasks, improving overall performance. Additionally, unlike full fine-tuning, our approach keeps the pre-trained parameters of \mathcal{S}_B , \mathcal{S}_C , and \mathcal{S}_Δ fixed. This strategy leverages pre-trained knowledge effectively and mitigates catastrophic forgetting of pre-trained knowledge in downstream tasks.

Methods	Publication	Backbone	Param	Pre-train	Cifar	Cifar10	DTD	CUB	Dogs	GTSRB	Flowers	SVHN	Birds	Food	Average
Full	-	Vim-S	25M	1K	89.6	98.8	72.3	80.1	92.5	97.5	90.5	98.0	77.8	87.4	88.5
VP	<i>arXiv 2022</i>	ViT-B	85M	21K	78.7	94.2	59.5	84.6	84.5	89.4	97.7	87.6	77.7	80.5	83.4
VPT	<i>ECCV 2022</i>	ViT-B	85M	21K	78.8	96.8	65.8	88.5	90.2	90.7	99.0	78.1	84.2	83.3	85.5
E ² VPT	<i>ICCV 2023</i>	ViT-B	85M	21K	80.4	97.1	66.8	89.1	90.5	91.0	99.1	79.2	84.6	84.0	86.2
DAM-VP	<i>CVPR 2023</i>	ViT-B	85M	21K	88.1	97.3	73.1	87.5	92.3	90.6	99.2	87.5	82.1	86.9	88.5
SA ² VP	<i>AAAI 2024</i>	ViT-B	85M	21K	91.3	98.6	75.6	89.0	92.2	96.3	99.2	96.4	86.0	90.1	91.5
AutoVP	<i>ICLR 2024</i>	CLIP	85M	400M	77.9	95.2	62.5	85.4	90.3	93.1	90.4	92.9	83.5	82.3	85.4
SPT	<i>ICML 2024</i>	ViT-B	85M	21K	79.2	97.6	66.5	90.6	89.8	91.3	98.3	92.1	87.6	84.3	87.7
VPT	<i>ECCV 2022</i>	ViT-S	22M	1K	82.0	96.8	64.6	72.5	88.6	94.0	86.0	94.7	64.7	79.6	82.4
DAM-VP	<i>CVPR 2023</i>	ViT-S	22M	1K	86.5	97.4	69.0	78.3	86.2	96.9	86.9	96.8	<u>75.0</u>	<u>85.1</u>	85.8
AutoVP	<i>ICLR 2024</i>	ViT-S	22M	1K	72.6	92.9	56.7	49.9	78.5	87.2	67.8	89.7	38.3	60.6	69.4
SPT	<i>ICML 2024</i>	ViT-S	22M	1K	84.2	97.1	<u>70.6</u>	79.1	91.8	96.0	<u>89.1</u>	94.5	72.7	81.0	85.6
Linear	-	Vim-S	25M	1K	78.1	93.8	64.5	68.1	94.8	67.7	87.0	53.4	55.0	68.0	73.0
VPT	<i>ECCV 2022</i>	Vim-S	25M	1K	84.9	96.6	69.2	77.3	94.9	92.2	88.2	93.5	69.2	79.2	84.5
DAM-VP	<i>CVPR 2023</i>	Vim-S	25M	1K	<u>87.8</u>	<u>98.0</u>	67.4	<u>79.4</u>	89.2	96.5	87.9	<u>96.9</u>	74.9	79.3	<u>85.7</u>
AutoVP	<i>ICLR 2024</i>	Vim-S	25M	1K	76.3	95.0	56.4	54.3	87.0	82.2	73.1	82.8	52.5	62.1	72.2
SPT	<i>ICML 2024</i>	Vim-S	25M	1K	84.2	96.7	68.5	74.8	95.0	95.8	81.6	91.1	66.9	77.8	83.2
SVP(ours)	<i>This Paper</i>	Vim-S	25M	1K	89.8	98.6	74.0	82.8	95.0	97.5	95.2	97.9	78.5	88.2	89.8

Table 1: The comparison results against state-of-the-art methods on HTA benchmark. The best results are bolded, and second-best underlined. 1K and 21K refer to ImageNet-1K and ImageNet-21K, respectively, while 400M refers to the 400 million image-text pairs used in CLIP pre-training.

Experiments

Experiment Setup

Datasets and Baselines. Following prior works (Huang et al. 2023; Pei et al. 2024), our experiments are carried out on two image classification benchmarks HTA and VTAB.

HTA. The head tuning adaptation benchmark (Huang et al. 2023) comprises 10 datasets including CIFAR10 (Krizhevsky, Hinton et al. 2009), CIFAR100 (Krizhevsky, Hinton et al. 2009), DTD (Cimpoi et al. 2014), CUB-200 (Wah et al. 2011), NABirds (Van Horn et al. 2015), Stanford-Dogs (Khosla et al. 2011), Oxford-Flowers (Nilsback and Zisserman 2008), Food101 (Bossard, Guillaumin, and Van Gool 2014), GTSRB (Stallkamp et al. 2012) and SVHN (Netzer et al. 2011).

VTAB-1K. It collects 19 benchmarks from Visual Task Adaptation (Zhai et al. 2019), categorized into three groups: i) Natural, ii) Specialized, and iii) Structured, each with 1000 training examples. Following (Zhai et al. 2019; Jia et al. 2022), we use an 800-200 train/val split.

Comparison Methods. We compare our SVP with other visual prompting methods including VPT (Jia et al. 2022), DAM-VP (Huang et al. 2023), AutoVP (Tsao et al. 2024) and SPT (Wang et al. 2024). We apply these methods to Vision Mamba (Zhu et al. 2024). Additionally, We compare the performance of these prompting methods using ViT-Small (Dosovitskiy et al. 2020) as the backbone, with the same model size and pre-trained dataset to Vision Mamba. We also present the visual prompting results in ViT-B with much larger parameters for reference.

Implementation Details. Our experiments primarily involve three pre-trained vision models: ViT-Small/16 and

Vim-Small, both of which are pre-trained on ImageNet-1K (Russakovsky et al. 2015), and ViT-Base/16 (Dosovitskiy et al. 2020), which is pre-trained on ImageNet-21K (Krizhevsky, Sutskever, and Hinton 2012). Following (Huang et al. 2023), all methods are trained for 100 epochs across all datasets for a fair comparison. For the compared methods, we use the optimizers specified in the original papers to achieve better performance. In our approach, we utilize the AdamW (Loshchilov and Hutter 2017) optimizer for optimization and implement cosine annealing. The number of shared layers in Cross-Prompting is set to 4, 8, or 12, depending on the dataset, and the hidden dimension of the inner-prompts generator is set to 64.

Comparison with State-of-the-arts

Results on HTA. We first conduct experiments on HTA datasets using the ImageNet-1k supervised ViT-Small/16 and Vim-Small as the pre-trained models. As shown in Table 1, our SVP significantly surpasses the results of existing prompting methods that use pre-trained models with the same model size and pre-training dataset. It not only achieves SOTA performance in average accuracy but also excels across nine out of ten datasets. For example, SVP achieves a notable improvement of **5.3%** over VPT when applied to Vim, while also surpassing DAM-VP by **4.1%**. This is because our SVP effectively activates the update and forget gates of Vim across the whole sequence, promoting the propagation of discriminative information, which leads to enhanced performance.

Additionally, compared to full fine-tuning, our method outperforms in 7 out of 10 datasets and exceeds 1.3% in av-

Methods	Backbone	Natural							Specialized				Structured						Average					
		CIFAR100	Caltech101	DTD	Flowers102	Pets	SVHN	Sun397	Average	Camelyon	EuroSAT	Resisc45	Retinopathy	Average	Clevr-Count	Clevr-Dist	DMLab	KITTI-Dist		dSpr-Loc	dSpr-Ori	sNORB-Azim	sNORB-Ele	Average
Full	Vim-S	49.9	86.9	66.5	93.2	91.6	88.3	40.3	73.8	86.3	93.9	83.7	76.4	85.1	54.3	48.3	50.5	52.7	36.0	34.9	28.5	32.9	42.3	67.1
Linear	Vim-S	47.0	84.2	60.7	77.1	90.0	42.3	39.8	63.0	78.6	87.2	71.1	73.8	77.7	32.4	33.9	35.8	50.9	49.6	49.3	20.0	22.6	36.8	59.2
VPT	Vim-S	<u>57.1</u>	<u>86.4</u>	65.0	<u>86.8</u>	<u>90.8</u>	<u>78.0</u>	<u>42.0</u>	<u>72.3</u>	<u>80.4</u>	<u>90.3</u>	<u>78.1</u>	<u>74.4</u>	<u>80.8</u>	<u>36.2</u>	<u>40.2</u>	<u>36.3</u>	<u>46.6</u>	<u>44.8</u>	<u>29.5</u>	<u>20.0</u>	<u>29.0</u>	<u>35.3</u>	<u>62.8</u>
SPT	Vim-S	54.0	85.5	68.5	82.8	89.7	72.0	40.0	70.4	79.8	89.8	75.2	<u>74.5</u>	79.8	35.0	38.0	<u>38.2</u>	<u>51.2</u>	38.8	40.1	<u>20.9</u>	<u>24.1</u>	<u>35.8</u>	<u>62.0</u>
SVP(ours)	Vim-S	59.3	87.4	<u>67.0</u>	92.4	92.8	88.5	43.2	75.8	86.0	94.7	84.2	77.1	85.5	54.1	52.0	52.0	51.8	62.3	54.2	30.7	35.4	49.1	70.1

Table 2: The comparison results against state-of-the-art methods on VTAB-1K benchmark. The best results are bolded, and second-best underlined. Overall ‘‘Average’’ is the group-wise average accuracy over three groups.

Format	Position	Cifar	DTD	CUB	Flowers
Append	Pre	84.9	69.2	77.3	88.4
	Post	84.9	69.9	77.2	88.0
	Both Sides	85.0	69.7	77.4	89.3
	Uniform	84.4	69.5	77.0	88.0
	Middle	85.3	69.7	77.6	88.9
SVP(Ours)		89.8	74.0	82.8	95.2

Table 3: Ablation of prompt formation and position. Append indicates a prompt sequence inserted into the token sequence, with Position specifying the insertion location.

Component		Cifar	DTD	CUB	Flowers
IP	CP				
-	-	78.1	64.5	68.1	86.0
-	✓	87.2	72.5	80.0	91.8
✓	-	89.6	71.6	82.1	93.2
✓	✓	89.8	74.0	82.8	95.2

Table 4: Ablation results of Cross-Prompting (CP) and Inner-Prompting (IP). - and ✓ represent without or with the component.

erage. This further supports the discussion in our methodology section, demonstrating that our approach mitigates the issue of catastrophic forgetting commonly seen in full fine-tuning. Our SVP retains more pre-trained knowledge while efficiently adapting to downstream tasks.

Notably, our method achieves performance comparable to prompting methods that use ViT-B as the backbone. ViT-B has a much larger model size of 85 Million parameters and is pre-trained on the much larger dataset ImageNet-21K (Deng et al. 2009). In contrast, our method uses a significantly smaller model and a smaller pre-training dataset, yet still delivers comparable results. Our approach outperforms methods such as DAM-VP (Huang et al. 2023) and SPT (Wang et al. 2024). This can be attributed to the dual-path Selective Prompting design in Vim, which effectively promotes the propagation of both shared inter-layer information and

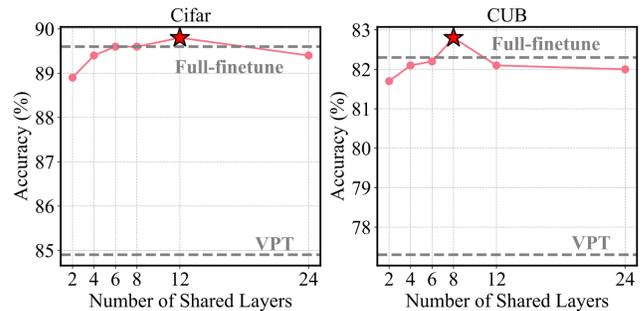


Figure 4: Ablation results of the number of shared layers in Cross-Prompting.

specific intra-layer information.

Results on VTAB-1K. Following (Jia et al. 2022; Pei et al. 2024), we also conduct experiments on another widely used VTAB-1k (Zhai et al. 2019) benchmark. As shown in Table 2, compared to the method VPT (Jia et al. 2022), our SVP achieves improvements of **3.5%**, **4.7%**, and **13.8%** in the three different tasks Natural, Specialized, and Structured, respectively. This further illustrates the effectiveness of our SVP in promoting the information flow and the robust adaptability of our SVP.

Ablation Study

Ablation of Prompt Format and Position. Given the specificity of Vim’s linear sequence model, where token impact varies by position, we first explore appending prompts to the image sequence and assess the effect of changing their position. As shown in Table 3, placing the prompt tokens in the middle yields a slight average improvement of 0.5% over in the pre, likely due to its closeness to the class token. However, this approach does not fully consider the sequential token-wise compression and propagation characteristics of the Vim sequence model. This limitation makes it ineffective in learning the input distribution across the entire sequence and in activating Vim’s update and forget gates. In contrast, our SVP attains an average **5.2%** improvement. It is because our SVP generates token-wise prompts based on the input,

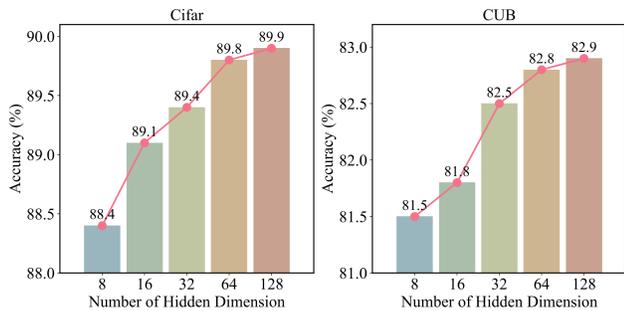


Figure 5: Ablation of hidden dimension in Inner-Prompting.

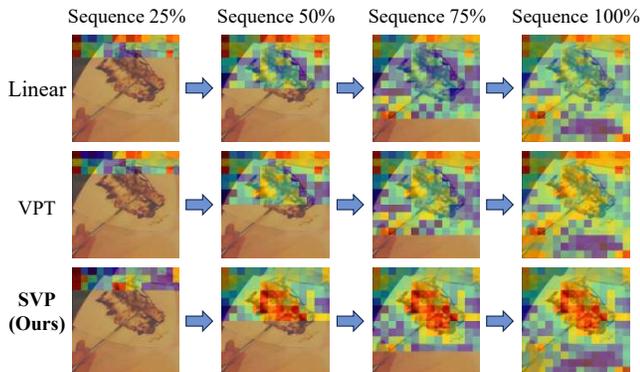


Figure 6: Visualization of normalized update gate of Vim over 24 layers during the sequence propagation. It reveals the information retained in Vim’s hidden states as sequence propagates to 25%, 50%, 75%, and the end of sequence.

more effectively learning the input distribution. This selective change activates the update and forget gates in Vim, promoting the propagation of discriminative information.

Influence of Different Components. To evaluate the effectiveness of dual-path prompting in our proposed SVP, we conduct ablation experiments on four datasets, as shown in Table 4. When no prompts are used, SVP reduces to a frozen pre-trained Vim model with a learnable classifier. On the CUB dataset, employing only the inner-prompts p^I boosts performance by 14%. Furthermore, using both the inner-prompts p^I and cross-prompts p^C together yields an additional 0.7% improvement.

This is because our dual-path SVP method captures more discriminative and richer information compared to the single-path approach. This enhancement arises from the synergy between the shared information provided by Cross-Prompting and the layer-specific details from Inner-Prompting, enabling more precise extraction and propagation of discriminative information.

Influence of Hyper-Parameters. The shared layers in Cross-Prompting and the hidden dimension of the inner-prompts generator (G^I) are important hyper-parameters in our SVP. To assess their impact, we conduct extensive ablation experiments. As shown in Figure 4, the model’s performance initially improves but then declines as the number of

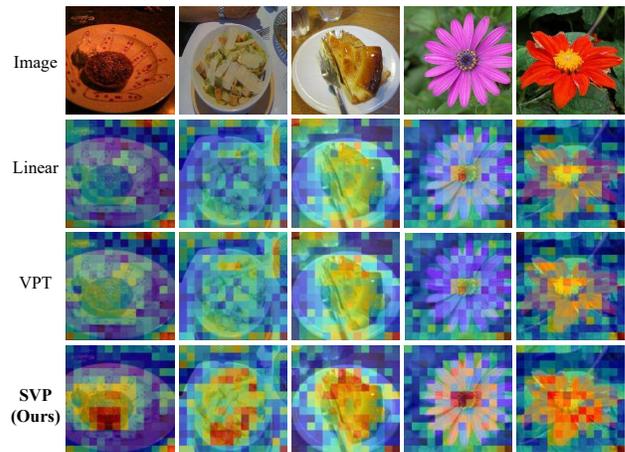


Figure 7: Visualization of the average values of the update gate over 24 layers in Vim.

shared layers increases, corresponding to the shallow, intermediate, and deep layers, which tend to share information respectively. For hidden dimension experiments in Figure 5, performance continues to improve with increasing rank, but the rate of improvement slows. To balance performance and tunable parameters, we select 64, resulting in 1.6M tunable parameters. For those willing to trade a small amount of performance, a dimension of 32, with 0.9M tunable parameters, is also an option.

Visualization Results of Update Gate in Vim. As shown in Figure 6, we first visualize the normalized update gate of Vim during the sequence propagation. This visualization reveals the information retained in Vim’s hidden states as the sequence progresses, highlighting that our method improves the extraction and transmission of discriminative features. Figure 7 shows more examples of the update gate within Vim. Our SVP retains more discriminative information related to key elements like flowers and food, while less discriminative backgrounds, such as plates, are retained to a lesser extent. The visualization results demonstrate that our method effectively activates the update gate in downstream tasks, enabling Vim to focus more on discriminative regions and thereby enhancing feature extraction and propagation.

Conclusion

In this paper, we introduce Selective Visual Prompting (SVP), an efficient and novel visual prompting method tailored for Vision Mamba (Vim). To the best of our knowledge, this is the initial exploration of visual prompting within Vim. Unlike existing approaches, our SVP leverages a dual-path strategy to achieve superior performance by leveraging both shared and layer-specific information. We find that prompts selectively generated based on input are more effective in activating Vim’s update and forget gates, promoting discriminative information propagation. Visualization results further validate our approach. We believe SVP will serve as a valuable benchmark that will drive future research in visual prompting for Vim.

Acknowledgments

This work was supported by the grants from the National Natural Science Foundation of China (62376011, 61925201, 62132001, 62432001) and Beijing Natural Science Foundation (L247006).

References

- Basu, S.; Hu, S.; Massiceti, D.; and Feizi, S. 2024. Strong Baselines for Parameter-Efficient Few-Shot Fine-Tuning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, 11024–11031.
- Bossard, L.; Guillaumin, M.; and Van Gool, L. 2014. Food-101—mining discriminative components with random forests. In *Computer vision—ECCV 2014: 13th European conference, Zurich, Switzerland, September 6–12, 2014, proceedings, part VI 13*, 446–461. Springer.
- Chen, S.; Ge, C.; Tong, Z.; Wang, J.; Song, Y.; Wang, J.; and Luo, P. 2022. Adaptformer: Adapting vision transformers for scalable visual recognition. *Advances in Neural Information Processing Systems*, 35: 16664–16678.
- Cimpoi, M.; Maji, S.; Kokkinos, I.; Mohamed, S.; and Vedaldi, A. 2014. Describing textures in the wild. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 3606–3613.
- Dao, T.; and Gu, A. 2024. Transformers are SSMs: Generalized Models and Efficient Algorithms Through Structured State Space Duality. In *Forty-first International Conference on Machine Learning*.
- Deng, J.; Dong, W.; Socher, R.; Li, L.-J.; Li, K.; and Fei-Fei, L. 2009. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, 248–255. Ieee.
- Dong, W.; Yan, D.; Lin, Z.; and Wang, P. 2024. Efficient adaptation of large vision transformer via adapter recomposing. *Advances in Neural Information Processing Systems*, 36.
- Dosovitskiy, A.; Beyer, L.; Kolesnikov, A.; Weissenborn, D.; Zhai, X.; Unterthiner, T.; Dehghani, M.; Minderer, M.; Heigold, G.; Gelly, S.; et al. 2020. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*.
- Elfwing, S.; Uchibe, E.; and Doya, K. 2018. Sigmoid-weighted linear units for neural network function approximation in reinforcement learning. *Neural networks*, 107: 3–11.
- Fu, Z.; Yang, H.; So, A. M.-C.; Lam, W.; Bing, L.; and Collier, N. 2023. On the effectiveness of parameter-efficient fine-tuning. In *Proceedings of the AAAI conference on artificial intelligence*, volume 37, 12799–12807.
- Gu, A.; and Dao, T. 2023. Mamba: Linear-time sequence modeling with selective state spaces. *arXiv preprint arXiv:2312.00752*.
- Gu, A.; Goel, K.; and Ré, C. 2021. Efficiently modeling long sequences with structured state spaces. *arXiv preprint arXiv:2111.00396*.
- Han, C.; Wang, Q.; Cui, Y.; Cao, Z.; Wang, W.; Qi, S.; and Liu, D. 2023. E² 2VPT: An Effective and Efficient Approach for Visual Prompt Tuning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 17491–17502.
- Huang, Q.; Dong, X.; Chen, D.; Zhang, W.; Wang, F.; Hua, G.; and Yu, N. 2023. Diversity-aware meta visual prompting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 10878–10887.
- Jia, M.; Tang, L.; Chen, B.-C.; Cardie, C.; Belongie, S.; Hariharan, B.; and Lim, S.-N. 2022. Visual prompt tuning. In *European Conference on Computer Vision*, 709–727. Springer.
- Khosla, A.; Jayadevaprakash, N.; Yao, B.; and Li, F.-F. 2011. Novel dataset for fine-grained image categorization: Stanford dogs. In *Proc. CVPR workshop on fine-grained visual categorization (FGVC)*, volume 2.
- Kornblith, S.; Shlens, J.; and Le, Q. V. 2019. Do better imagenet models transfer better? In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2661–2671.
- Krizhevsky, A.; Hinton, G.; et al. 2009. Learning multiple layers of features from tiny images.
- Krizhevsky, A.; Sutskever, I.; and Hinton, G. E. 2012. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25.
- Liu, Z.; Peng, Y.; and Zhou, J. 2024. InsVP: Efficient Instance Visual Prompting from Image Itself. In *ACM Multimedia 2024*.
- Loshchilov, I.; and Hutter, F. 2017. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*.
- Netzer, Y.; Wang, T.; Coates, A.; Bissacco, A.; Wu, B.; Ng, A. Y.; et al. 2011. Reading digits in natural images with unsupervised feature learning. In *NIPS workshop on deep learning and unsupervised feature learning*, volume 2011, 4. Granada.
- Nilsback, M.-E.; and Zisserman, A. 2008. Automated flower classification over a large number of classes. In *2008 Sixth Indian conference on computer vision, graphics & image processing*, 722–729. IEEE.
- Pechlivanidou, G.; and Karampetakis, N. 2022. Zero-order hold discretization of general state space systems with input delay. *IMA Journal of Mathematical Control and Information*, 39(2): 708–730.
- Pei, W.; Xia, T.; Chen, F.; Li, J.; Tian, J.; and Lu, G. 2024. SA²VP: Spatially Aligned-and-Adapted Visual Prompt. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, 4450–4458.
- Russakovsky, O.; Deng, J.; Su, H.; Krause, J.; Satheesh, S.; Ma, S.; Huang, Z.; Karpathy, A.; Khosla, A.; Bernstein, M.; et al. 2015. Imagenet large scale visual recognition challenge. *International journal of computer vision*, 115: 211–252.

Stallkamp, J.; Schlipsing, M.; Salmen, J.; and Igel, C. 2012. Man vs. computer: Benchmarking machine learning algorithms for traffic sign recognition. *Neural networks*, 32: 323–332.

Steitz, J.-M. O.; and Roth, S. 2024. Adapters Strike Back. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 23449–23459.

Sung, Y.-L.; Cho, J.; and Bansal, M. 2022. Lst: Ladder side-tuning for parameter and memory efficient transfer learning. *Advances in Neural Information Processing Systems*, 35: 12991–13005.

Touvron, H.; Cord, M.; El-Nouby, A.; Verbeek, J.; and Jégou, H. 2022. Three things everyone should know about vision transformers. In *European Conference on Computer Vision*, 497–515. Springer.

Tsao, H.-A.; Hsiung, L.; Chen, P.-Y.; Liu, S.; and Ho, T.-Y. 2024. AutoVP: An Automated Visual Prompting Framework and Benchmark. In *The Twelfth International Conference on Learning Representations*.

Van Horn, G.; Branson, S.; Farrell, R.; Haber, S.; Barry, J.; Ipeirotsis, P.; Perona, P.; and Belongie, S. 2015. 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*, 595–604.

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

Wang, Y.; Cheng, L.; Fang, C.; Zhang, D.; Duan, M.; and Wang, M. 2024. Revisiting the Power of Prompt for Visual Tuning. In *Forty-first International Conference on Machine Learning*.

Xin, Y.; Du, J.; Wang, Q.; Lin, Z.; and Yan, K. 2024a. Vmt-adapter: Parameter-efficient transfer learning for multi-task dense scene understanding. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, 16085–16093.

Xin, Y.; Luo, S.; Zhou, H.; Du, J.; Liu, X.; Fan, Y.; Li, Q.; and Du, Y. 2024b. Parameter-efficient fine-tuning for pre-trained vision models: A survey. *arXiv preprint arXiv:2402.02242*.

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

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

Zhu, L.; Liao, B.; Zhang, Q.; Wang, X.; Liu, W.; and Wang, X. 2024. Vision Mamba: Efficient Visual Representation Learning with Bidirectional State Space Model. In *Forty-first International Conference on Machine Learning*.