Parameter-Efficient Fine-Tuning for Pre-Trained Vision Models: A Survey

Yi Xin[1] , **Siqi Luo**[1,2] , **Haodi Zhou**[1] , **Junlong Du**[3] , **Xiaohong Liu**[2] ,
Yue Fan[4] , **Qing Li**[4] , **Yuntao Du**[4*]

[1] Nanjing University, [2] Shanghai Jiao Tong University, [3] Youtu Lab, Tencent,
[4] Beijing Institute for General Artificial Intelligence (BIGAI)

{xinyi, siqiluo, haodizhou}@smail.nju.edu.cn, jeffdu@tencent.com
xiaohongliu@sjtu.edu.cn, {liqing, fanyue, duyuntao}@bigai.ai

Abstract

Large-scale pre-trained vision models (PVMs) have shown great potential for adaptability across various downstream vision tasks. However, with state-of-the-art PVMs growing to billions or even trillions of parameters, the standard full fine-tuning paradigm is becoming unsustainable due to high computational and storage demands. In response, researchers are exploring parameter-efficient fine-tuning (PEFT), which seeks to exceed the performance of full fine-tuning with minimal parameter modifications. This survey provides a comprehensive overview and future directions for visual PEFT, offering a systematic review of the latest advancements. First, we provide a formal definition of PEFT and discuss model pre-training methods. We then categorize existing methods into three categories: addition-based, partial-based, and unified-based. Finally, we introduce the commonly used datasets and applications and suggest potential future research challenges. A comprehensive collection of resources is available at https://github.com/synbol/Awesome-Parameter-Efficient-Transfer-Learning.

1 Introduction

With the development of available datasets [Deng and et al., 2009], model architectures [Dosovitskiy and et al., 2021], and training algorithms [He and et al., 2022], a significant number of *vision foundation models* have been developed. Particularly, transformer-based pre-trained vision models (PVMs) [Khan and et al., 2022] have demonstrated remarkable performance across various computer vision tasks, such as image classification [Dosovitskiy and et al., 2021] and semantic segmentation [Kirillov and et al., 2023].

Owing to the powerful representational abilities of PVMs, it has become a popular paradigm to fine-tune PVMs for learning downstream tasks. However, traditional full fine-tuning, though effective, requires substantial computational and memory resources. This becomes particularly costly for models with billions or even trillions of parameters. Additionally, there is a requirement to maintain separate model weights for each dataset, which becomes impractical as the number of tasks increases, especially in the case of large PVMs.

As a promising solution, parameter-efficient fine-tuning (PEFT), which was originally proposed in NLP, overcomes the above challenges by updating a minimal number of parameters while potentially achieving comparable or superior performance to full fine-tuning [Hu and et al., 2021; Yu and et al., 2022]. These approaches hinge on recent advances showing that large pre-trained models trained with rich data have strong generalisability and most parameters in the PVMs could be shared for the new tasks [Kornblith and et al., 2019; Yu and et al., 2022]. PEFT methods could reduce learnable parameters, which not only facilitates more effective adaptation to novel tasks but also safeguards the pre-existing knowledge within the PVMs. Taking into account the prospects of PEFT and the fast-paced development of large-scale vision models, a survey that provides a detailed and up-to-date investigation of PEFT in the vision domain is in urgent demand.

This paper aims to provide a comprehensive and systematic study of PEFT methods in the vision domain, particularly focusing on transformer-based pre-trained models ranging from the year 2019 to the year 2023. As shown in Fig. 1, existing visual PEFT methods could be categorized into addition-based tuning, partial-based tuning, and unified-based tuning. In section 2, we will define the problem of PEFT, introduce popular backbones, and discuss pre-training methods. In section 3, a detailed taxonomy and in-depth analysis of the PEFT methods will be presented. The real-world applications of PEFT will be introduced in section 4. Finally, in section 5, we will point out future research challenges.

2 Preliminaries

2.1 Problem Definition

Definition 1. (Parameter-efficient Fine-tuning). *Given a pre-trained model M parametrized by θ, and a downstream task $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^{|\mathcal{D}|}$, where (x_i, y_i) serves as a ground-truth input-output pair of task $\mathcal{D}$, parameter-efficient fine-tuning aims to adapt θ to task $\mathcal{D}$, where task-specific parameters increment $\Delta\theta$ is introduced with $|\Delta\theta| \ll |\theta|$. The optimal parameters are found by optimizing the losses $\mathcal{L}$ on task $\mathcal{D}$:*

$$\min_{\Delta\theta} \mathbb{E}_{(x_i, y_i) \in \mathcal{D}} \mathcal{L}(M_{\theta+\Delta\theta}(\hat{y}_i|x_i), y_i). \quad (1)$$
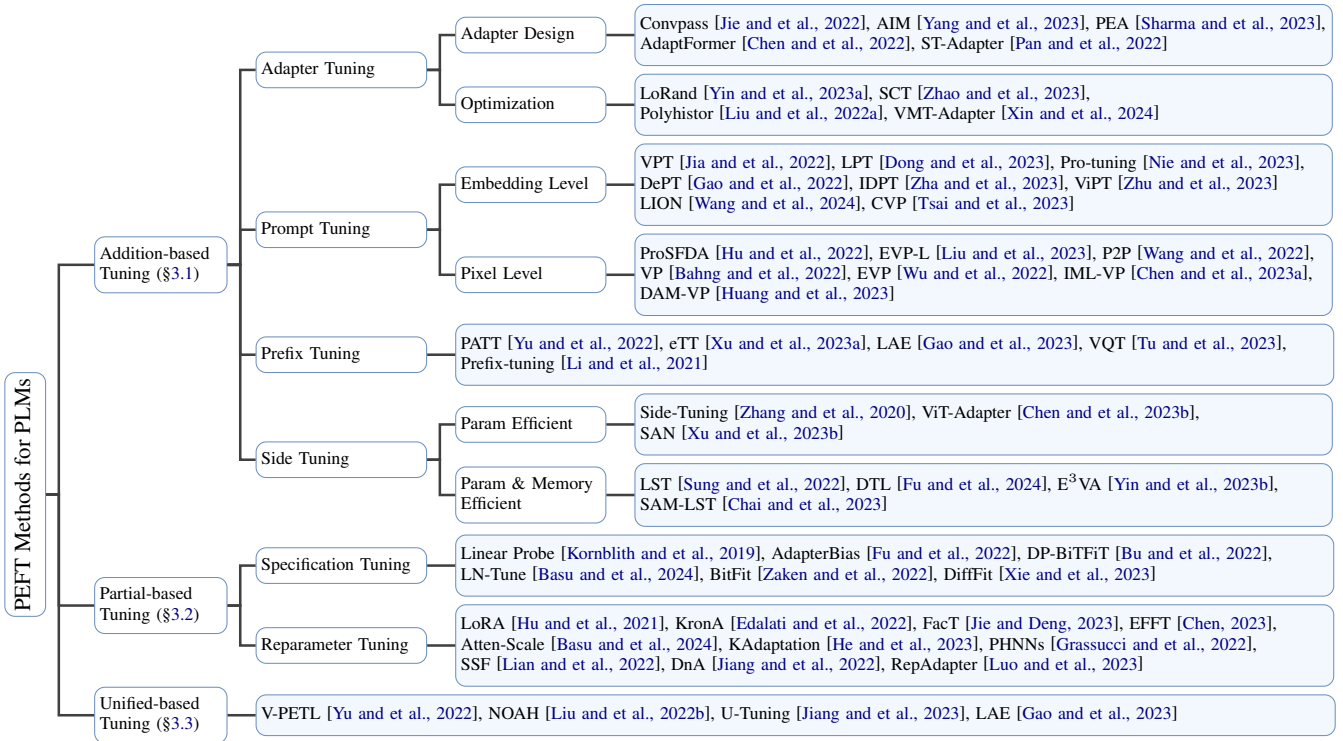
---

*Corresponding author

Figure 1: Taxonomy of Parameter-Efficient Fine-Tuning Methods for Pre-trained Vision Models.

2.2 Vision Transformer

The standard Vision Transformer [Dosovitskiy and et al., 2021] consists of a patch embedding layer and L Transformer layers. Given an image $x \in \mathbb{R}^{H \times W \times C}$, the patch embedding layer first splits and flatten the image x into sequential patches $x_p \in \mathbb{R}^{N \times (P^2 C)}$, where (H, W) represents the height and width of the input image, (P, P) is the resolution of each image patch, C denotes the number of channels, and $N = HW/P^2$ is the number of image tokens. Then, x_p is mapped to $x_0 \in \mathbb{R}^{N \times d}$ with a trainable linear projection. The combination of a prepended $[cls]$ token and x_0 are the inputs of Transformer encoders.

Each Transformer layer consists of a multi-head attention (MHA) and a multilayer perceptron (MLP) module. In MHA, attention scores are computed using query (Q), key (K), and value (V) representations, along with projection matrices $W_q, W_k, W_v \in \mathbb{R}^{d \times d}$. Given an input $x_{\ell-1}$ at the ℓ-th layer, the attention is calculated as follows:

$$Q = x_{\ell-1}W_q, \ K = x_{\ell-1}W_k, \ V = x_{\ell-1}W_v, \quad (2)$$

$$x'_\ell = Attention(Q, K, V) = softmax(\frac{QK^T}{\sqrt{d}})V. \quad (3)$$

The output tokens x'_ℓ are further sent to a LayerNorm(LN) and an MLP block, which is formulated as follows:

$$x_\ell = MLP(LN(x'_\ell)) + x'_\ell, \quad (4)$$

where x_ℓ is the output of the ℓ-th encoder layer.

Recent advancements in vision Transformer architectures have significantly enhanced performance in vision tasks. A line of works improves the standard ViT by integrating additional or contextual information, with notable models like Pyramid DeiT [Touvron and et al., 2021] and Token to Token (T2T) ViT [Yuan and et al., 2021]. Another line of work focuses on multi-scale ViTs using hierarchical designs to capture spatial details at varying scales, which is a capability limited in standard ViTs due to fixed token numbers and dimensions. Key models in this category include Pyramid ViT (PVT) [Wang and et al., 2021] and Swin Transformer [Liu and et al., 2021]. A more comprehensive survey can be found in this literature [Khan and et al., 2022].

2.3 Model Pre-training

Recently, many pre-training methods with innovative backbones for training PVMs have emerged. The pre-training methods can be mostly categorized into supervised learning and self-supervised learning.

Supervised pre-training. These methods use classification losses for pre-training on large annotated datasets, *e.g.*, ImageNet [Deng and et al., 2009]. A renowned pre-trained model that applies supervised pre-training is SAM [Kirillov and et al., 2023], which is trained on pixel-level annotated datasets and achieves excellent results in segmentation tasks.

Self-supervised pre-training. Self-supervised learning, now a leading pre-training paradigm, includes 1) Contrastive learning methods, which focus on attracting similar (positive) and repelling dissimilar (negative) samples. This includes both image-based approaches such as SimCLR [Chen and et al., 2020], MoCo [He and et al., 2020], DINO [Caron and et al.,

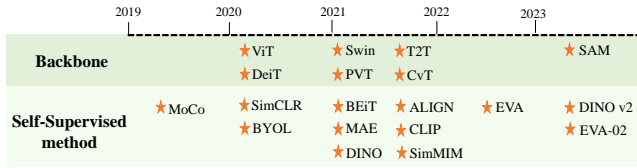|	2019	2020	2021	2022	2023	
Backbone		★ViT ★DeiT	★Swin ★PVT	★T2T ★CvT		★SAM
Self-Supervised method	★MoCo	★SimCLR ★BYOL	★BEiT ★MAE ★DINO	★ALIGN ★CLIP ★SimMIM	★EVA	★DINO v2 ★EVA-02

Figure 2: The representative backbones and pre-training methods.

2021], and multi-modality based approaches like CLIP [Radford and et al., 2021] and ALIGN [Jia and et al., 2021]. Note that we exclusively focus on the image-related modules for multimodal self-supervised models, ignoring other modalities. 2) Mask image modeling methods, including MAE [He and et al., 2022], SimMIM [Xie and et al., 2021], and EVA [Fang and et al., 2022], which involve masking parts of images and reconstructing them.

3 Methodology

3.1 Addition-based Methods

Addition-based methods involve incorporating additional trainable modules or parameters into original PVMs to learn task-specific information. This subsection discusses four primary branches of representative addition-based methods: adapter tuning, prompt tuning, prefix tuning, and side tuning.

Adapter Tuning. As a pioneering work, the adapter was initially introduced in the NLP domain by [Houlsby and et al., 2019] to achieve PEFT. Owing to its remarkable effectiveness, it has been successfully adopted in the CV field as well. This method integrates small neural modules, termed adapters, into the Transformer layers. During the adaptation process, only these adapters are fine-tuned. The adapter architecture consists of a down-projection layer parameterized by $W_{down} \in \mathbb{R}^{d \times k}$ and an up-projection layer parameterized by $W_{up} \in \mathbb{R}^{k \times d}$. Here, k (with $k << d$) serves to reduce the dimension of the representation into a lower rank. Furthermore, a ReLU layer is positioned between two layers to enable non-linear projection. For a given input feature map $x_\ell \in \mathbb{R}^{N \times d}$, the adapter generates optimized features as follows:

$$\hat{x}_\ell = \text{ReLU}\left(x_\ell W_{down}\right) W_{up}, \tag{5}$$

where $W = [W_{down}; W_{up}^T] \in \mathbb{R}^{d \times 2k}$ denotes all the trainable parameters in the adapter.

Adapter tuning methods in CV domain can be broadly divided into two categories: **1) design specific adapter architectures for various vision tasks** (*e.g.*, image classification, video understanding, etc.), and **2) employ advanced optimization techniques to reduce the trainable parameters in the adapter.**

In the first category, AdaptFormer [Chen and et al., 2022] serves as a typical example. It marks the first instance of adapting vision transformers to a broad array of downstream visual recognition tasks using adapters. Notably, AdaptFormer doesn't modify the structure of the adapter but demonstrates that parallel insertion of adapters is more efficacious for vision tasks than the sequential insertion typically employed in NLP tasks. Another key contribution is Convpass [Jie and et al., 2022], which highlights that current adapters are hindered by a lack of strong inductive bias, limiting their performance.

To overcome this, Convpass incorporates trainable convolutional blocks, thereby enhancing the adapter's capabilities by integrating the strengths of convolutional neural networks. Additionally, AIM [Yang and et al., 2023] introduces adapters specialized in spatial, temporal, and joint domains, while ST-Adapter [Pan and et al., 2022] offers a spatiotemporal adapter. Both methods are tailored to improve a vision model's spatiotemporal reasoning for video understanding tasks. In the field of robotic manipulation, Rob-Adapter [Sharma and et al., 2023] applies the classic bottleneck architecture, commonly used in image classification, for lossless adaptation.

In the second category, these methods focus on optimizing the adapter's architecture to reduce trainable parameters. One such example is LoRand [Yin and et al., 2023a], which creates compact adapter structures through a low-rank synthesis approach. This method achieves a reduction in parameters by parameterizing both the down-projection layer W_{down} and the up-projection layer W_{up} through the multiplication of three low-rank matrices. Another distinct approach is presented in SCT [Zhao and et al., 2023], which opts for a selective channel tuning strategy, focusing on specific task-relevant channels to lower parameter costs. Furthermore, Polyhistor [Liu and et al., 2022a] adopts a unique method by decomposing a hyper-network into two separate hyper-networks and factorizing an adapter's weight matrix into two kernels. This technique is particularly beneficial in multi-task architectures, contributing to a reduction in the number of parameters. Expanding on the ideas of Polyhistor, VMT-Adapter [Xin and et al., 2024] integrates knowledge extraction modules to adapt to multiple vision tasks efficiently, demonstrating both parameter and training efficiency.

Prompt Tuning. Visual prompt tuning methods provide an alternative to injecting learnable modules into the Transformer model. In such a method, the original input, whether an image embedding or the actual image, is wrapped with visual prompts. These prompts consist of additional trainable parameters or perturbations. They are uniquely adaptable parameters and can be optimized according to the specific task and the training data. The primary goal is to align the input distribution to original pre-training data with task-specific prompts. Research in visual prompt tuning typically falls into two main categories: **1) inject a set of learnable parameters into the image embedding space**, and **2) inject learnable perturbations around the border of the original input image.**

In the first category, VPT [Jia and et al., 2022] is a pioneering work. It presents two variants: VPT-Shallow (see Fig. 3(c)) and VPT-Deep. VPT-Shallow integrates additional l learnable prompts, denoted as $P = [P_1], [P_2], ...[P_l] \in \mathbb{R}^{l \times d}$, into the input patch embeddings $x_0 \in \mathbb{R}^{N \times d}$. These prompts are then concatenated with path embedding to form the final input. This process can be expressed as follows:

$$x_0 = concat(P, x_0) = [P, x_0] \in \mathbb{R}^{(l+N) \times d}, \tag{6}$$

where $[\cdot, \cdot]$ is the concatenation along the token dimension. VPT-Deep advances VPT-Shallow by adding prompts to every Transformer layer's input space, updating only these prompts during fine-tuning while keeping pre-trained parameters frozen. The cost of VPT-Deep depends on the prompt
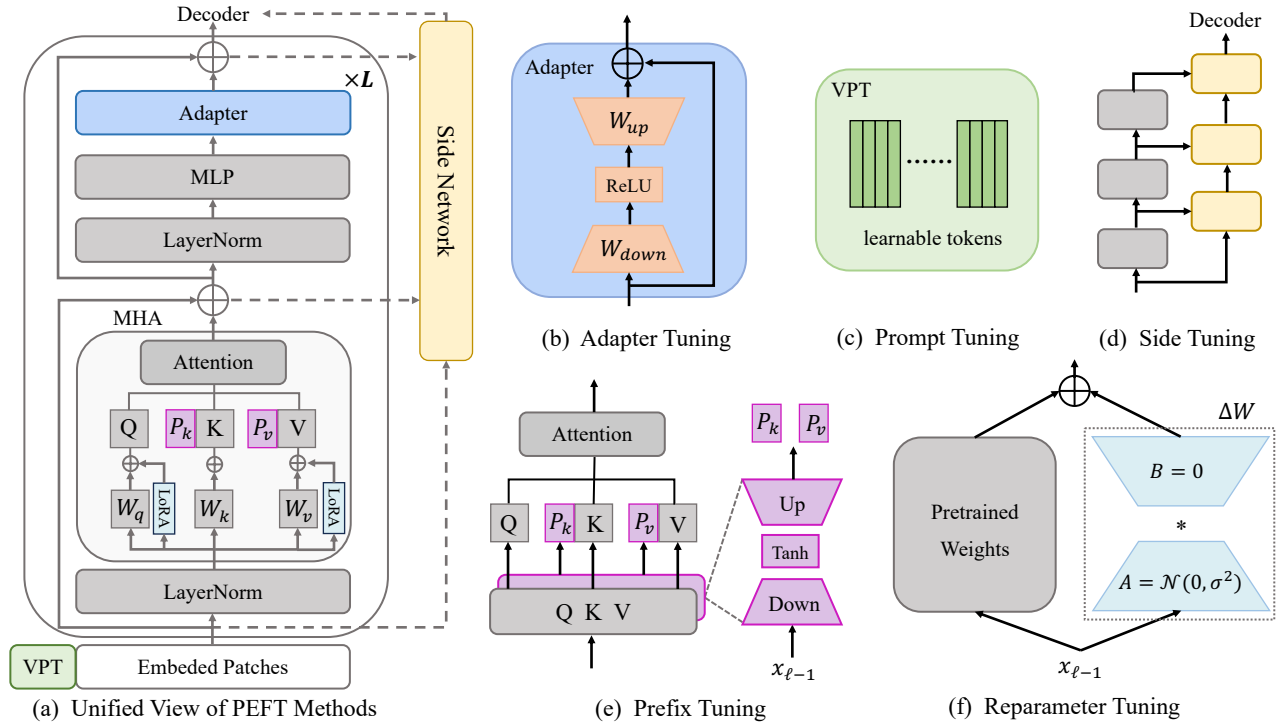
Figure 3: The detailed architecture of various PEFT methods.

length and token embedding dimension, and the experiments show that longer prompts yielding better performance. Similarly, DePT [Gao and et al., 2022] introduces learnable visual prompts into the vision Transformer, specifically for data-efficient test-time domain adaptation. Similarly, CVP [Tsai and et al., 2023] proposes a self-supervised convolutional prompt for robust visual perception. Additionally, LPT [Dong and et al., 2023] optimizes shared prompts to extract general features for the long-tailed dataset. IDPT [Zha and et al., 2023] ventures into applying visual prompt tuning on pre-trained point cloud models. Moreover, some works focused on designing sub-networks to produce visual prompts. Pro-Tuning [Nie and et al., 2023] designs lightweight prompt blocks, comprising three lightweight convolutional layers, to generate task-specific discriminative prompts for each downstream input image. LION [Wang and et al., 2024] adds two implicit layers, positioned at the beginning and end of the PVMs, serving as visual prompts to enrich the visual input and representation. Lastly, ViPT [Zhu and et al., 2023] takes both RGB and auxiliary modal inputs, which are initially processed by the patch embed to generate corresponding RGB and prompt tokens.

In the second category, research focuses on optimizing task-specific prompts at the pixel level, integrating these prompts directly with input images. A key method in this category is VP [Bahng and et al., 2022], which modifies learnable perturbations around image borders and does not require PVMs access at test time. Building on VP, EVP [Wu and et al., 2022] employs a strategy where images are shrunk and subjected to data augmentations, followed by padding the area around the image with the prompt. DAM-VP [Huang and et al., 2023] adopts a divide-and-conquer strategy. It segments high-diversity datasets into subsets and learns separate prompts for

each subset, addressing issues related to large data diversity. Furthermore, this category of methods is particularly effective for pixel-level tasks, such as image segmentation and point cloud analysis. For instance, EVP-L [Liu and et al., 2023] employs the high-frequency components of the input as prompts for low-level structure segmentation tasks. ProSFDA [Hu and et al., 2022] adds a zero-initialized learnable prompt to target images in medical image segmentation. P2P [Wang and et al., 2022] converts point cloud data into colorful images, which are then used as vision prompts to adapt PVMs for various point cloud analysis tasks. Additionally, to further understand and improve visual prompt effectiveness, ILM-VP [Chen and et al., 2023a] automatically remaps source labels to target labels, enhancing the target task accuracy of visual prompting.

Prefix Tuning. Inspired by the success of prompt tuning, Prefix-tuning [Li and et al., 2021] introduces learnable prefix matrices to the MHA module of the PVMs. It involves prepending two randomly initialized prefix matrices $P_k, P_v \in \mathbb{R}^{l \times d}$ to the keys and values in the MHA, leading the attention calculation in Eq. 3 to:

$$Attention(Q, K, V) = softmax(\frac{Q[P_k, K]^T}{\sqrt{d}})[P_v, V] \quad (7)$$

However, random initialization may bring random noise, impacting the convergence of the fine-tuning downstream tasks. To address this, PATT [Yu and et al., 2022] proposes a parallel attention mechanism to the original attention module without random initialization and uses two linear layers (with parameters $W_{down} \in \mathbb{R}^{d \times k}$ and $W_{up} \in \mathbb{R}^{k \times l}$) and Tanh layers to transform prefix matrices (see Fig. 3e). Specifically, for the l-th Transformer layer, given previous layer's output $x_{\ell-1}$, and we get a pair of prefix matrices via:

$$P_k, P_v = Tanh(x_{\ell-1}W_{down})W_{up}. \quad (8)$$

Following PATT, eTT [Xu and et al., 2023a] uses the latest innovations in attentive prefix tuning (*i.e.*, generating new key-value pairs) for few-shot learning and LAM [Gao and et al., 2023] includes prefix tuning as part of the framework for continual learning. In contrast to original Prefix-tuning, VQT [Tu and et al., 2023] only appends additional prefix vectors to the query Q, not to both the value V and key K.

Side Tuning. Different from previous PEFT methods that typically involve inserting additional modules or parameters inside PVMs, side tuning employs a side network, a smaller and separate network that operates in parallel with the PVMs, as shown in Fig. 3(d).

Earlier side tuning methods concentrated on parameter efficiency, with a key focus on how to design the side network. Side-Tuning [Zhang and et al., 2020] utilizes a four-layer convolutional network as the additive side network. The outputs of the network are summed with the representations from the PVMs in the final layer, facilitating the resolution of various tasks. Furthermore, SAN [Xu and et al., 2023b] proposes a two-branch side adapter network. One branch is dedicated to predicting mask proposals, while the other focuses on predicting attention biases that are applied to the self-attention blocks for mask class recognition. More recently, ViT-Adapter [Chen and et al., 2023b] designs a spatial prior module along with two feature interaction operations, which enables integrating image priors into the architecture of ViT without necessitating a redesign. Such an arrangement is particularly beneficial for dense prediction tasks as it supplements missing local information and reorganizes fine-grained, multi-scale features.

Besides prioritizing parameter efficiency, **following works discovered that side tuning can lead to GPU memory efficiency through innovative designs**. LST [Sung and et al., 2022] proposes to separate trainable parameters from the backbone model to create a small Transformers network. This separation completely obviates the need for costly backpropagation through a large backbone network, resulting in significant GPU memory savings. Building upon the ideas in LST, SAM-LST [Chai and et al., 2023] incorporates an additional convolutional neural network as a complementary encoder within the SAM. This integration leads to faster training and reduced resource demands. However, as LST is not directly applicable to some PVMs like the Swin Transformer, E^{3}VA [Yin and et al., 2023b] provides a gradient backpropagation highway for low-rank adapters. This method is compatible with all PVMs and further enhances efficiency. More recently, DTL [Fu and et al., 2024] designs a compact side network specifically for ViT to achieve both parameter and GPU memory efficiency.

3.2 Partial-based Methods

Partial-based methods concentrate on updating only a small subset of inherent parameters while maintaining the majority of the model's parameters unchanged during the adaptation process. These methods do not seek to change the internal structure of the model. This section will cover two strategies: specification tuning and reparameter tuning.

Specification Tuning. Specification tuning is an efficient approach that directly modifies a specific subset of parameters in PVMs, such as bias and LayerNorm, which are crucial for downstream tasks. This method concentrate on important parameters while discarding those deemed less relevant. The concept, while straightforward, has proven to be surprisingly effective. One of the earliest examples is Linear Probe [Kornblith and et al., 2019], which introduces a linear layer as the classifier on the top of PVMs. In this method, all parameters of the PVMs are frozen, allowing for an exploration of the pre-training capabilities of the PVMs. This technique has become a standard baseline in various PEFT methods. Moreover, BitFit [Zaken and et al., 2022] empirically demonstrates that optimizing only the bias terms within a model can be effective, which could be represented as follows:

$$x_\ell = x_{\ell-1}W_\ell + b_\ell, \tag{9}$$

where the weight parameters W_ℓ are kept frozen, and only the bias b_ℓ is optimized during the tuning process. Remarkably, this approach enables the model to retain over 95% of its performance across several benchmarks. Building on the principles of BitFit, DP-BiTFiT [Bu and et al., 2022] combines the efficiency of the standard BiTFit approach to address downstream tasks involving sensitive data and achieves state-of-the-art accuracy for differentially private algorithms. Similarly, DiffFit [Xie and et al., 2023] only fine-tunes the bias term and newly added scaling factors in specific layers of diffusion models, and this strategy results in training speed-ups and reduced model storage costs. Meanwhile, AdapterBias [Fu and et al., 2022] presents a unique approach that avoids altering the bias of the PVMs. Instead, it targets the bias term at the MLP layer by using a linear layer with a weight α and a tunable vector v, which is expressed as follows:

$$x_\ell = x_{\ell-1}W_\ell + b_\ell + \alpha \otimes v. \tag{10}$$

Instead of tuning bias terms, LN-Tune [Basu and et al., 2024] introduces a strong PEFT baseline that fine-tunes only the LayerNorm parameters of the PVMs.

Reparameter Tuning. Reparameter tuning methods also introduce new learnable parameters during the training stage, while these parameters can be integrated into the original PVMs through reparameterization during the inference phase. LoRA [Hu and et al., 2021] is a prominent example, where trainable low-rank matrices are injected into Transformer layers to approximate updates to the weights. For a pre-trained weight matrix W_ℓ, LoRA represents its update with a low-rank decomposition:

$$W_l^{'} = W_\ell + \triangle W = W_\ell + BA, \tag{11}$$

where B and A are trainable parameters. Generally, LoRA updates the query and value projection matrices in multi-head attention. Since then, there has been plenty of following research in this area. KronA [Edalati and et al., 2022] shares structural similarities with LoRA but differs in replacing LoRA's low-rank decomposition with Kronecker product decomposition, expressed as $\triangle W = B \otimes A$. This modification enhances computational efficiency and reduces the number of required floating-point operations (FLOPs). Building on these concepts, KAdaptation [He and et al., 2023] decomposes update weights through a sum of n Kronecker products between shared slow weights A_i and independent fast weights B_i. and it further

takes an additional step by decomposing B_i into the product of two low-rank matrices u_i and v_i:

$$W + \triangle W = W + \sum_{i=1}^{n} A_i \otimes B_i = \sum_{i=1}^{n} A_i \otimes (u_i v_i^T). \quad (12)$$

Thus, the trainable parameters are now substantially reduced in this manner. Delving deeper, FacT [Jie and Deng, 2023] proposes a tensorization-decomposition framework, which involves tensorizing the weights of PVMs into a single 3D tensor and then decomposing their increments into lightweight factors. This approach efficiently stores weight increments, offering a novel way to handle the parameters of PVMs. Following FacT, EFFT [Chen, 2023] aims to minimize redundancies both within and across layers, without increasing computational latency. This method exemplifies how tensor decomposition can be leveraged for more efficient model tuning. Beyond pre-trained weight matrices, other works have explored different parameters of PVMs. SSF [Lian and et al., 2022] integrates learnable scale and shift parameters to adjust features and then reparameterizes these into the MLP layer. RepAdapter [Luo and et al., 2023] demonstrates that adapter modules can be seamlessly integrated into PVMs via structural reparameterization, thereby achieving zero cost during inference.

3.3 Unified-based Tuning

Unified-based tuning approaches offer a unified framework to integrate various fine-tuning methods into a single, harmonized architecture. This approach streamlines the process and enhances the overall efficiency and effectiveness of the fine-tuning. For instance, NOAH [Liu and et al., 2022b] incorporates Adapter, LoRA, and VPT into each Transformer block and employs Neural Architecture Search (NAS) to determine the best design for specific downstream tasks. This method represents a comprehensive approach to optimizing fine-tuning by combining multiple techniques. LAM [Gao and et al., 2023] proposes a unified framework for continual learning. This framework is designed to be adaptable, allowing any PEFT method to be reconfigured into a competitive approach for continual learning. Additionally, V-PEFT [Yu and et al., 2022] provides a unified analysis of PEFT techniques for video tasks. This research investigates the critical aspects of fine-tuning positions, offering a cohesive view of these techniques. Similarly, U-Tuning [Jiang and et al., 2023] rethinks PEFT from an integrated perspective, re-evaluating existing tuning paradigms. It identifies a parallel form for mainstream tuning methods, including adapter, prefix, and prompt tuning, which effectively reduces the coupling in tuning structures.

3.4 Discussion

Characteristic Analysis. We summarize the characteristics of all PEFT methods, as shown in Tab. 1. The methods are compared in 4 aspects. 1) No-Additional Modules (**NAM**): Specification tuning is the only method that does not introduce new modules, while others introduce additional modules or parameters more or less. 2) Structure Preserving (**SP**): Adapter Tuning changes the structure of the PVMs. By contrast, prompt tuning, prefix tuning, side tuning, and reparameter tuning maintain the structure of the original PVM while introducing new modules. Specification Tuning directly optimizes a subset of parameters of PVMs, so it does not change

Category	Characteristic				Representative Method	#Trainable Params
	NAM	SP	IE	ME		
ADAPTER	✗	✗	✗	✗	AdaptFormer	$L \times (2dk)$
PROMPT	✗	✓	✗	✗	VPT-Deep	$L \times (ld)$
PREFIX	✗	✓	✗	✗	PATT	$L \times 2 \times (dk + kl)$
SIDE	✗	✓	✗	✓	LST	#Params of subnetwork
SPECIFICATION	✓	✓	✓	✗	BitFit	$L \times (7 \times d)$
REPARAMETER	✗	✓	✓	✗	LoRA	$L \times 2 \times (2dk)$

Table 1: Comparison between different tuning methods.

the model structure. 3) Inference Efficient (**IE**): Additional modules typically increase inference latency, with reparameter tuning being an exception due to its mitigating reparameterization technique. 4) Memory Efficient (**ME**): Side tuning uniquely achieves memory efficiency as its gradient backpropagation does not involve PVMs. Overall, each PEFT method presents unique advantages and limitations, and there is no completely perfect PEFT method.

Parameter Analysis. To accurately calculate the number of trainable parameters, we select a specific, representative work for each taxonomy, as shown in Tab. 1. It is observed that BitFit has the smallest number of trainable parameters since it only updates the bias terms in PVMs. In contrast, LST has the largest trainable parameters due to its parallel subnetwork, but it can achieve memory efficient. Optimization of the subnetwork structure may be crucial in the future. Additionally, AdaptFormer, PATT, and LoRA share similar parameter magnitudes, as they all inject comparable structures into each Transformer layer. VPT-Deep has a slightly higher parameter count than BitFit. In practical applications, compared to full fine-tuning, these methods possess only 0.05% to 10% of the trainable parameter, yet they achieve comparable or even better performance on downstream tasks.

4 Datasets and Applications

In this section, we briefly discuss the popular datasets and applications in visual PEFT, as shown in Tab. 2. Image recognition is the primary benchmark and application for PEFT, exemplified by datasets such as FGVC [Jia and et al., 2022] (5 downstream tasks) and VTAB-1k [Zhai and et al., 2019] (19 downstream tasks). PEFT is also influential in other domains. Beyond image classification, video action recognition is another key application area, involving datasets like Kinetics-400 [Kay and et al., 2017], SSv2 [Goyal and et al., 2017], HMDB51 [Kuehne and et al., 2011] and Driving-48 [Li and et al., 2018]. Additionally, PEFT has been utilized for dense prediction tasks, using datasets like COCO [Lin and et al., 2014], ADE20K [Zhou and et al., 2019] and PASCAL VOC [Everingham and et al., 2015]. Furthermore, the use of PEFT is expanding into new fields, including point cloud analysis and robotic manipulation. It's evident that PEFT is increasingly applied across various domains and prevailing in diverse downstream tasks.

Application	Dataset	Description	#Classes	Train size	Val size	Test size
	Fine-Grained Visual Classification (FGVC) [Jia and et al., 2022]					
	CUB-200-2011	Fine-grained bird species recognition	200	5,394	600	5,794
	NABirds	Fine-grained bird species recognition	55	21,536	2,393	24,633
	Oxford Flowers	Fine-grained flower species recognition	102	1,020	1,020	6,149
	Stanford Dogs	Fine-grained dog species recognition	120	10,800	1,200	8,580
	Stanford Cars	Fine-grained car classification	196	7,329	815	8,041
	Visual Task Adaptation Benchmark (VTAB-1k) [Zhai and et al., 2019]					
Image Recognition	CIFAR-100	*Natural*-tasks that contain natural images captured using standard cameras.	100	800	200	10,000
	Caltech101		102			6,084
	DTD		47			1,880
	Flowers102		102			6,149
	Pets		37			3,669
	SVHN		10			26,032
	Sun397		397			21,750
	Patch Camelyon	*Specialized*-tasks that contain images captured via specialized equipment, such as medical and satellite imagery.	2	800	200	32,768
	EuroSAT		10			5,400
	Resisc45		45			6,300
	Retinopathy		5			42,670
	Clevr/count	*Structured*-tasks that require geometric comprehension like object counting.	8	800	200	15,000
	Clevr/distance		6			15,000
	DMLab		6			22,735
	KITTI/distance		4			711
	dSprites/location		16			73,728
	dSprites/orientation		16			73,728
	SmallNORB/azimuth		18			12,150
	SmallNORB/elevation		9			12,150
Video Recognition	Kinetics-400 [Kay and et al., 2017]	Video action recognition	400	240,436	N/A	19,787
	SSv2 [Goyal and et al., 2017]		174	168,913	24,777	27,157
	HMDB51 [Kuehne and et al., 2011]		51	3,500	1,500	1,849
	Diving-48 [Li and et al., 2018]		48	15,900	N/A	2,000
	UCF-101 [Soomro and et al., 2012]		101	9,537	N/A	3,783
Dense Prediction	MS COCO [Lin and et al., 2014]	Instance segmentation	80	118,000	N/A	5,000
	ADE20K [Zhou and et al., 2019]	Semantic segmentation	150	20,210	N/A	2,000
	PASCAL VOC [Everingham and et al., 2015]	Semantic segmentation	21	1,464	N/A	1,449

Table 2: Several popular datasets and applications of visual PEFT.

5 Future Research Challenges

Explainability of Visual PEFT Methods.   Despite significant advancements, the underlying reasons for the effectiveness of visual PEFT methods remain unclear, especially in terms of the interpretability of visual prompts. In the NLP domain, we can explain the prompt as a better description, which is more intuitive. While in CV domain, the main challenge is that visual prompts are learned as unordered token-based prompts, which are difficult to translate into an understandable format. Other tuning techniques like adapter and prefix also confront challenges in interpretability. These methods strive to reduce the number of parameters required for adapting large models to specific tasks. Therefore, improving the interpretability of PEFT is a crucial area for future research.

PEFT for Generative and Multimodal Models.   On one side, within the CV domain, most PEFT methods are tailored for discriminative tasks, such as image classification and video action recognition. Yet, exploring their application in generative tasks is highly promising. With the help of adapter and prompt, researchers have developed several PEFT methods for pre-trained generative models [Xie and et al., 2023], particularly stable diffusion models. Nonetheless, these models still have much room for deeper exploration. On the other side, large multimodal models typically require more computational and memory resources compared to single-modal models. Therefore, investigating PEFT methods in the mul-

timodal domain is also desirable. Moreover, PEFT methods could facilitate cross-modality alignment, leading to significant improvements in downstream multimodal tasks. Consequently, further exploration in both these domains represents a promising direction for future research.

Building Visual PEFT Library.   While numerous PEFT methods for the vision domain have been proposed, their direct employment or comparison is not conventional. In contrast, the NLP domain has developed comprehensive libraries like PEFT library[1], which integrate various PEFT methods and large language models (LLMs) to facilitate their application in downstream tasks. Thus, it is desired to develop a library for the vision domain and even integrate the multimodal domain, which could boost the development of PEFT.

6 Conclusion

In this paper, we conduct a comprehensive review of the visual parameter-efficient fine-tuning domain by offering an in-depth analysis of existing methods, datasets, and applications. We conclude with a detailed comparison of these methods and identify several potential research challenges in the field. Our goal is for this survey to serve as a valuable resource for researchers interested in parameter-efficient fine-tuning, providing insights that could inspire further advancements.

---

[1]https://github.com/huggingface/peft

References

[Bahng and et al., 2022] Hyojin Bahng and et al. Exploring visual prompts for adapting large-scale models. *Arxiv*, 2022.

[Basu and et al., 2024] Samyadeep Basu and et al. Strong baselines for parameter efficient few-shot fine-tuning. *AAAI*, 2024.

[Bu and et al., 2022] Zhiqi Bu and et al. Differentially private bias-term only fine-tuning of foundation models. *NeurIPS*, 2022.

[Caron and et al., 2021] Mathilde Caron and et al. Emerging properties in self-supervised vision transformers. *ICCV*, 2021.

[Chai and et al., 2023] Shurong Chai and et al. Ladder fine-tuning approach for sam integrating complementary network. *Arxiv*, 2023.

[Chen and et al., 2020] Ting Chen and et al. A simple framework for contrastive learning of visual representations. *ICML*, 2020.

[Chen and et al., 2022] Shoufa Chen and et al. Adaptformer: Adapting vision transformers for scalable visual recognition. *NeurIPS*, 2022.

[Chen and et al., 2023a] Aochuan Chen and et al. Understanding and improving visual prompting: A label-mapping perspective. *CVPR*, 2023.

[Chen and et al., 2023b] Zhe Chen and et al. Vision transformer adapter for dense predictions. *ICLR*, 2023.

[Chen, 2023] Dongping Chen. Aggregate, decompose, and fine-tune: A simple yet effective factor-tuning method for vision transformer. *Arxiv*, 2023.

[Deng and et al., 2009] Jia Deng and et al. Imagenet: A large-scale hierarchical image database. *CVPR*, 2009.

[Dong and et al., 2023] Bowen Dong and et al. Lpt: Long-tailed prompt tuning for image classification. *ICLR*, 2023.

[Dosovitskiy and et al., 2021] Alexey Dosovitskiy and et al. An image is worth 16x16 words: Transformers for image recognition at scale. *ICLR*, 2021.

[Edalati and et al., 2022] Ali Edalati and et al. Krona: Parameter efficient tuning with kronecker adapter. *Arxiv*, 2022.

[Everingham and et al., 2015] Mark Everingham and et al. The pascal visual object classes challenge: A retrospective. *IJCV*, 2015.

[Fang and et al., 2022] Yuxin Fang and et al. Eva: Exploring the limits of masked visual representation learning at scale. *CVPR*, 2022.

[Fu and et al., 2022] Chin-Lun Fu and et al. Adapterbias: Parameter-efficient token-dependent representation shift for adapters in nlp tasks. *NAACL*, 2022.

[Fu and et al., 2024] Minghao Fu and et al. Dtl: Disentangled transfer learning for visual recognition. *AAAI*, 2024.

[Gao and et al., 2022] Yunhe Gao and et al. Visual prompt tuning for test-time domain adaptation. *Arxiv*, 2022.

[Gao and et al., 2023] Qiankun Gao and et al. A unified continual learning framework with general parameter-efficient tuning. *ICCV*, 2023.

[Goyal and et al., 2017] Raghav Goyal and et al. The" something something" video database for learning and evaluating visual common sense. *ICCV*, 2017.

[Grassucci and et al., 2022] Eleonora Grassucci and et al. Phnns: Lightweight neural networks via parameterized hypercomplex convolutions. *TNNLS*, 2022.

[He and et al., 2020] Kaiming He and et al. Momentum contrast for unsupervised visual representation learning. *CVPR*, 2020.

[He and et al., 2022] Kaiming He and et al. Masked autoencoders are scalable vision learners. *CVPR*, 2022.

[He and et al., 2023] Xuehai He and et al. Parameter-efficient model adaptation for vision transformers. *AAAI*, 2023.

[Houlsby and et al., 2019] Neil Houlsby and et al. Parameter-efficient transfer learning for nlp. *ICML*, 2019.

[Hu and et al., 2021] Edward J Hu and et al. Lora: Low-rank adaptation of large language models. *ICLR*, 2021.

[Hu and et al., 2022] Shishuai Hu and et al. Prosfda: Prompt learning based source-free domain adaptation for medical image segmentation. *Arxiv*, 2022.

[Huang and et al., 2023] Qidong Huang and et al. Diversity-aware meta visual prompting. *CVPR*, 2023.

[Jia and et al., 2021] Chao Jia and et al. Scaling up visual and vision-language representation learning with noisy text supervision. *ICML*, 2021.

[Jia and et al., 2022] Menglin Jia and et al. Visual prompt tuning. *ECCV*, 2022.

[Jiang and et al., 2022] Ziyu Jiang and et al. Dna: Improving few-shot transfer learning with low-rank decomposition and alignment. *ECCV*, 2022.

[Jiang and et al., 2023] Zeyinzi Jiang and et al. Rethinking efficient tuning methods from a unified perspective. *Arxiv*, 2023.

[Jie and Deng, 2023] Shibo Jie and Zhi-Hong Deng. Fact: Factor-tuning for lightweight adaptation on vision transformer. *AAAI*, 2023.

[Jie and et al., 2022] Shibo Jie and et al. Convolutional bypasses are better vision transformer adapters. *Arxiv*, 2022.

[Kay and et al., 2017] Will Kay and et al. The kinetics human action video dataset. *Arxiv*, 2017.

[Khan and et al., 2022] Salman Khan and et al. Transformers in vision: A survey. *CSUR*, 2022.

[Kirillov and et al., 2023] Alexander Kirillov and et al. Segment anything. *Arxiv*, 2023.

[Kornblith and et al., 2019] Simon Kornblith and et al. Do better imagenet models transfer better? *CVPR*, 2019.

[Kuehne and et al., 2011] Hildegard Kuehne and et al. Hmdb: a large video database for human motion recognition. *ICCV*, 2011.

[Li and et al., 2018] Yingwei Li and et al. Resound: Towards action recognition without representation bias. *ECCV*, 2018.

[Li and et al., 2021] Xiang Lisa Li and et al. Prefix-tuning: Optimizing continuous prompts for generation. *ACL*, 2021.

[Lian and et al., 2022] Dongze Lian and et al. Scaling & shifting your features: A new baseline for efficient model tuning. *NeurIPS*, 2022.

[Lin and et al., 2014] Tsung-Yi Lin and et al. Microsoft coco: Common objects in context. *ECCV*, 2014.

[Liu and et al., 2021] Ze Liu and et al. Swin transformer: Hierarchical vision transformer using shifted windows. *ICCV*, 2021.

[Liu and et al., 2022a] Yen-Cheng Liu and et al. Polyhistor: Parameter-efficient multi-task adaptation for dense vision tasks. *NeurIPS*, 2022.

[Liu and et al., 2022b] Yuanhan Liu and et al. Neural prompt search. *Arxiv*, 2022.

[Liu and et al., 2023] Weihuang Liu and et al. Explicit visual prompting for low-level structure segmentations. *CVPR*, 2023.

[Luo and et al., 2023] Gen Luo and et al. Towards efficient visual adaption via structural re-parameterization. *Arxiv*, 2023.

[Nie and et al., 2023] Xing Nie and et al. Pro-tuning: Unified prompt tuning for vision tasks. *TCSVT*, 2023.

[Pan and et al., 2022] Junting Pan and et al. St-adapter: Parameter-efficient image-to-video transfer learning. *NeurIPS*, 2022.

[Radford and et al., 2021] Alec Radford and et al. Learning transferable visual models from natural language supervision. *ICML*, 2021.

[Sharma and et al., 2023] Mohit Sharma and et al. Lossless adaptation of pretrained vision models for robotic manipulation. *ICLR*, 2023.

[Soomro and et al., 2012] Khurram Soomro and et al. Ucf101: A dataset of 101 human actions classes from videos in the wild. *Arxiv*, 2012.

[Sung and et al., 2022] Yi-Lin Sung and et al. Lst: Ladder side-tuning for parameter and memory efficient transfer learning. *NeurIPS*, 2022.

[Touvron and et al., 2021] Hugo Touvron and et al. Training data-efficient image transformers & distillation through attention. *ICML*, 2021.

[Tsai and et al., 2023] Yun-Yun Tsai and et al. Convolutional visual prompt for robust visual perception. *NeurIPS*, 2023.

[Tu and et al., 2023] Cheng-Hao Tu and et al. Visual query tuning: Towards effective usage of intermediate representations for parameter and memory efficient transfer learning. *CVPR*, 2023.

[Wang and et al., 2021] Wenhai Wang and et al. Pyramid vision transformer: A versatile backbone for dense prediction without convolutions. *ICCV*, 2021.

[Wang and et al., 2022] Ziyi Wang and et al. P2p: Tuning pretrained image models for point cloud analysis with point-to-pixel prompting. *NeurIPS*, 2022.

[Wang and et al., 2024] Haixin Wang and et al. Lion: Implicit vision prompt tuning. *AAAI*, 2024.

[Wu and et al., 2022] Junyang Wu and et al. Unleashing the power of visual prompting at the pixel level. *Arxiv*, 2022.

[Xie and et al., 2021] Zhenda Xie and et al. Simmim: a simple framework for masked image modeling. *CVPR*, 2021.

[Xie and et al., 2023] Enze Xie and et al. Difffit: Unlocking transferability of large diffusion models via simple parameter-efficient fine-tuning. *Arxiv*, 2023.

[Xin and et al., 2024] Yi Xin and et al. Vmt-adapter: Parameter-efficient transfer learning for multi-task dense. *AAAI*, 2024.

[Xu and et al., 2023a] Chengming Xu and et al. Exploring efficient few-shot adaptation for vision transformers. *TMLR*, 2023.

[Xu and et al., 2023b] Mengde Xu and et al. Side adapter network for open-vocabulary semantic segmentation. *CVPR*, 2023.

[Yang and et al., 2023] Taojiannan Yang and et al. Aim: Adapting image models for efficient video action recognition. *ICLR*, 2023.

[Yin and et al., 2023a] Dongshuo Yin and et al. 1% vs 100%: Parameter-efficient low rank adapter for dense predictions. *CVPR*, 2023.

[Yin and et al., 2023b] Dongshuo Yin and et al. Parameter-efficient is not sufficient: Exploring parameter, memory, and time efficient adapter tuning for dense predictions. *Arxiv*, 2023.

[Yu and et al., 2022] Bruce XB Yu and et al. Towards a unified view on visual parameter-efficient transfer learning. *Arxiv*, 2022.

[Yuan and et al., 2021] Li Yuan and et al. Tokens-to-token vit: Training vision transformers from scratch on imagenet. *ICCV*, 2021.

[Zaken and et al., 2022] Elad Ben Zaken and et al. Bitfit: Simple parameter-efficient fine-tuning for transformer-based masked language-models. *ACL*, 2022.

[Zha and et al., 2023] Yaohua Zha and et al. Instance-aware dynamic prompt tuning for pre-trained point cloud models. *ICCV*, 2023.

[Zhai and et al., 2019] Xiaohua Zhai and et al. A large-scale study of representation learning with the visual task adaptation benchmark. *Arxiv*, 2019.

[Zhang and et al., 2020] Jeffrey Zhang and et al. Side-tuning: a baseline for network adaptation via additive side networks. *ECCV*, 2020.

[Zhao and et al., 2023] Henry Hengyuan Zhao and et al. Sct: A simple baseline for parameter-efficient fine-tuning via salient channels. *IJCV*, 2023.

[Zhou and et al., 2019] Bolei Zhou and et al. Semantic understanding of scenes through the ade20k dataset. *IJCV*, 2019.

[Zhu and et al., 2023] Jiawen Zhu and et al. Visual prompt multimodal tracking. *CVPR*, 2023.