VFM-Adapter: Adapting Visual Foundation Models for Dense Prediction with Dynamic Hybrid Operation Mapping

Zheng Chen[1], Yu Zeng[1], Zehui Chen[1], Hongzhi Gao[1], Lin Chen[1], Jiaming Liu[2], Feng Zhao[1]*

[1] MoE Key Laboratory of Brain-inspired Intelligent Perception and Cognition, University of Science and Technology of China
[2] Peking University
{chenzheng, zy18803425504, lovesnow, hongzhigao, chlin}@mail.ustc.edu.cn
jiamingliu@stu.pku.edu.cn, fzhao956@ustc.edu.cn

Abstract

Although pre-trained large vision foundation models (VFM) yield superior results on various downstream tasks, full fine-tuning is often impractical due to its high computational cost and storage requirements. Recent advancements in parameter-efficient fine-tuning (PEFT) of VFM for image classification show significant promise. However, the application of PEFT techniques to dense prediction tasks remains largely unexplored. Our analysis of existing methods reveals that the underlying premise of utilizing low-rank parameter matrices, despite their efficacy in specific applications, may not be adequately suitable for dense prediction tasks. To this end, we propose a novel PEFT learning approach tailored for dense prediction tasks, namely VFM-Adapter. Specifically, the VFM-Adapter introduces a hybrid operation mapping technique that seamlessly integrates local information with global modeling to the adapter module. It capitalizes on the distinct inductive biases inherent in different operations. Additionally, we dynamically generate parameters for the VFM-Adapter, enabling flexibility of feature extraction given specific inputs. To validate the efficacy of VFM-Adapter, we conduct extensive experiments across object detection, semantic segmentation, and instance segmentation tasks. Results on multiple benchmarks consistently demonstrate the superiority of our method over previous approaches. Notably, with only three percent of the trainable parameters of the SAM-Base backbone, our approach achieves competitive or even superior performance compared to full fine-tuning. The code will be available.

Introduction

With the rapid development of computer vision, a wide spectrum of deep neural networks emerged, promoting massive progress in various applications, e.g., object detection (Ren et al. 2015; Lin et al. 2017; Wang, Bochkovskiy, and Liao 2023; Tian et al. 2019; Carion et al. 2020; Chen et al. 2024), semantic segmentation (Long, Shelhamer, and Darrell 2015; Chen et al. 2017; Cheng et al. 2022; Xiao et al. 2018; Xie et al. 2021), and depth estimation (Patil et al. 2022; Yang et al. 2022; Mi, Di, and Xu 2022; Guizilini et al. 2023; Ma et al. 2022). Recently, there has been an increasing research interest in vision foundation models (VFM) (Fang et al. 2023; Oquab et al. 2023; Kirillov et al. 2023; Liu et al. 2021; Zou et al. 2023) due to the impressive demonstrations from both language (Floridi and Chiriatti 2020; Touvron et al. 2023; Zeng et al. 2022) and vision (Chen et al. 2023; Radford et al. 2021) communities.

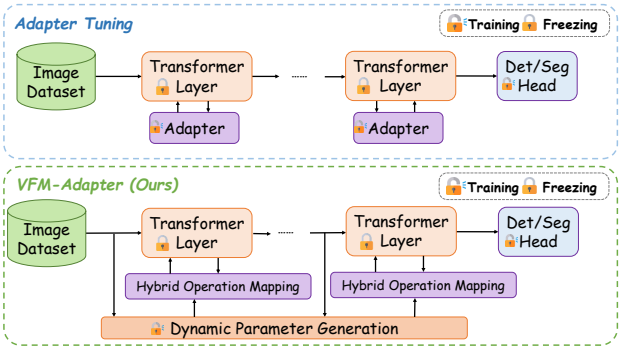

Figure 1: **Characteristics of Our Proposed VFM-adapter.** Adapter tuning freezes the parameters of the pre-trained model while exclusively training the parameters of the adapter and prediction head. In contrast, we harness the power of Hybrid Operation Mapping (HOM) to seamlessly aggregate global and local features, thereby effectively capturing comprehensive contextual information. Moreover, we employ Dynamic Parameter Generation (DPG) to dynamically synthesize adaptive parameters tailored to the HOM module. This innovative approach confers exceptional flexibility in our feature extraction process, transcending the constraints imposed by previous adapter-based methods.

Despite the claim of strong zero-shot abilities by these models (Fang et al. 2023; Oquab et al. 2023; Kirillov et al. 2023; Liu et al. 2021), we empirically find that simply utilizing the output features derived from these VFM still suffers inferior performance on various downstream tasks, especially on dense visual predictions. Fully fine-tuning the entire backbone and task-specific heads can revitalize VFM's performance. However, as network parameters continue to scale up, this strategy becomes unsustainable due to the enormous computational demands, including GPU memory and extended training periods. Moreover, dedicating a unique, parameter-heavy VFM to each dataset is both inefficient and impractical. Consequently, devising an effective

*Corresponding author.

method to adapt VFM for broader visual tasks emerges as a critical challenge.

Parameter-efficient fine-tuning (PEFT) learning, which is first investigated in natural language processing (NLP) (Houlsby et al. 2019; Zaken, Ravfogel, and Goldberg 2021), seeks to replicate the results of full fine-tuning by updating only a fraction of the backbone parameters or adding lightweight structures. Some recent work (Chen et al. 2022; Jia et al. 2022) introduces the PEFT concept to computer vision for efficiently fine-tuning VFM to downstream tasks, especially the visual classification task. However, how to adapt VFM for dense prediction tasks, *e.g.,* object detection, semantic segmentation, and instance segmentation remains under-explored. Contrary to visual classification, which assigns a single category to the entire image, dense prediction tasks require detailed region/pixel-level analysis. The model must accurately classify or segment each region/pixel, necessitating advanced perceptual and reasoning capabilities to detect subtle nuances and interpret semantic details across various image regions. Consequently, designing PEFT methods tailored for the unique demands of dense prediction tasks presents a significant challenge.

We perform a detailed analysis to understand why current PEFT methods are unsuitable for dense prediction tasks. Many methods assume that the full fine-tuning weight matrix is low-rank, but this assumption may not hold for dense prediction tasks. In Figure 2, we observe that the singular values distribution in the weight matrix for dense prediction tasks is not concentrated around zero, unlike what is seen in classification tasks. Instead, the singular values distribution for dense prediction tasks is scattered, suggesting that the parameter matrix lacks a low-rank structure.

Therefore, PEFT methods predicated solely on low-rankness are inadequate for dense prediction tasks. Taking LoRA (Hu et al. 2021) as an example, a straightforward idea is that we can increase the rank in the model. However, this results in a diminishing improvement in performance and a disproportionate increase in parameters, contradicting the foundational principle of PEFT methods. In addition, the parameters in the current PEFT method are independent of the input image, which prevents the adapter from learning semantic information from complex images.

To address these challenges, we propose **VFM-Adapter**, an innovative and effective adapter specialized for dense visual predictions. To expand the representation space generated by visual adapters while ensuring efficient parameters, we introduce the hybrid operation mapping module to seamlessly combine the local information and global modeling within adapters. Besides, to fully enhance the representation ability of our VFM-Adapter, we propose a dynamic adapter generation mechanism conditioned on the input images, enabling the flexibility of the feature extraction in our inserted adapters.

To validate the generality and superiority of our method, we choose three representative VFM *i.e.,* DINOv2 (Oquab et al. 2023), SAM (Kirillov et al. 2023), and Swin Transformer (Liu et al. 2021) to conduct experiments on common dense prediction tasks, including object detection, instance segmentation, and semantic segmentation. Extensive experimental results demonstrate the superiority and generalization of our approach. With only 3% of learnable parameters, VFM-Adapter even exceeds the performance of full fine-tuning on the SAM-Base backbone.

In a nutshell, our contributions are three-fold:

- We highlight that the unsuitability of current PEFT methods for dense prediction tasks stems from the non-low-rank nature of their parameter matrices.
- We introduce the VFM-Adapter, a simple yet effective approach that incorporates hybrid operation mapping and dynamically generates input-aware parameters for the adapter module.
- Extensive experimental results show the effectiveness and generality of our approach on various dense prediction tasks, providing an efficient solution for this field.

Related Work

Large Visual Foundation Models

In recent years, there has been a rapid growth in large vision foundation models. Models based on ViT (Dosovitskiy et al. 2020) or Swin Transformer (Liu et al. 2021) have been extended and trained on larger datasets, such as ImageNet-21K (Deng et al. 2009) and JFT-300M citesun2017revisiting. DINOv2 (Oquab et al. 2023), for instance, employs a self-supervised approach (He et al. 2022) to train a ViT model with 1 billion tunable parameters. Furthermore, the 1B model can be compressed into a series of smaller models through unsupervised distillation, making it suitable for various tasks. SAM (Kirillov et al. 2023), on the other hand, is trained on the SA-1B (Kirillov et al. 2023) dataset and exhibits the ability to learn a wide range of visual generic features, including the concept of objects. SAM can be applied to different downstream tasks in a zero-shot manner. However, when these models are directly applied to dense predict tasks (freeze the backbone) such as object detection and semantic segmentation, their performance may not meet expectations. Hence, there is a need for efficient parameter fine-tuning to optimize their performance in downstream tasks.

Parameter-Efficient Fine-Tuning

Parameter-efficient fine-tuning has emerged as a widely adopted technique in the field of NLP. Adapter (Houlsby et al. 2019) proposes the integration of a bottleneck-like module into the transformer blocks. During training, the original backbone is frozen, and only this module is trained. BitFit (Zaken, Ravfogel, and Goldberg 2021) presents a straightforward approach that fine-tunes solely the bias parameter in the backbone. LoRA (Hu et al. 2021) introduces a parallel structure to the original weights, formed by the multiplication of two low-rank matrices. In the domain of computer vision, Adapterformer (Chen et al. 2022) and ConvPass (Jie and Deng 2022) follow LoRA's paradigm by incorporating a parallel adapter into the original structure. However, Adapterformer is parallel to the entire FFN (Feed-Forward Network) model and introduces nonlinearities in the adapter, while ConvPass (Jie and Deng 2022) adds a

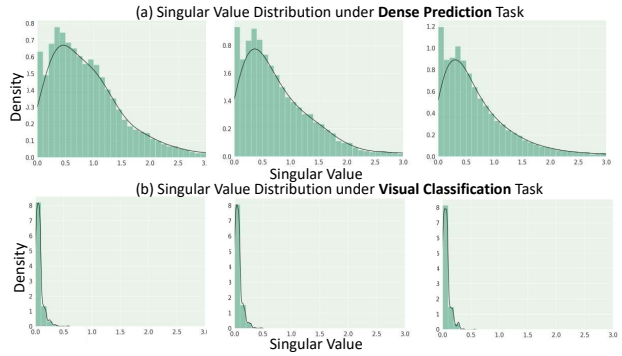

Figure 2: **Singular Value Distribution of Parameter Matrices under Dense Task and Visual Classification Task.** We select the 1-st layer, 5-th layer, and 9-th layer of SAM-Base's MHA module as representative examples. Comparable observations can be extended to the remaining layers, including the FFN layers. The first row illustrates the singular value distribution of a fully fine-tuning MHA parameter matrix under a dense prediction task, while the second row depicts the same matrix under a visual classification task. The horizontal axis represents the singular values of the parameter matrix, while the vertical axis depicts their associated density distribution.

convolutional operation to the adapter, leveraging the inductive bias of the visual task itself. SSF (Lian et al. 2022) and ARC (Dong et al. 2023) enable efficient parameter fine-tuning without any increase in inference time through reparameterization techniques. VPT (Jia et al. 2022) employs additional learnable tokens integrated into the input space as prompts for model learning. Although the aforementioned methods have demonstrated relatively favorable outcomes in the field of vision classification, the investigation of parameter-efficient fine-tuning (PEFT) methods for dense prediction tasks remains scarce. Lorand (Yin et al. 2023), designed specifically for dense prediction tasks, utilizes a low-rank synthesis approach. In light of the current state of research, it inspires us to explore more effective approaches in this realm.

Dynamic Parameter Generation

A general neural network employs a fixed set of network parameters during the testing phase, utilizing them for inference on all test samples. In contrast, dynamic parameter generation involves the neural network dynamically adapting its parameters for each individual sample during the inference phase. In order to achieve dynamic parameterization during inference, ConvLora (Zhong et al. 2024) leverages the concept of Mixture of Experts (MoE). Other methods (Chen et al. 2020; Ma et al. 2020; Jia et al. 2016; Yang et al. 2019) follow a more direct approach by using networks to predict parameters. In our work, we employ the hypernet framework (Ha, Dai, and Le 2016) to dynamically generate parameters based on the inputs. This approach retains the generality of the foundational model while enhancing its expressive power and adaptivity.

Pilot Observation

In this section, we present our motivation by carrying out pilot experiments and show the hazards of previous low-rank design-based methods directly applied to dense prediction tasks.

We select SAM-Base and apply it as the backbone to the object detection task on the COCO dataset for full fine-tuning. We conduct a detailed analysis of the singular value distribution of weight matrices $W \in \mathbb{R}$, as shown in Figure 2. The first row illustrates the singular value distribution of a fully fine-tuning MHA parameter matrix under a dense prediction task, while the second row depicts the same matrix under a visual classification task. This indicates that the parameter matrix demonstrates low-rank characteristics in visual classification tasks, whereas dense prediction tasks do not exhibit such a phenomenon. Consequently, prior methods such as LoRA (Hu et al. 2021) and ARC (Dong et al. 2023), which rely on the low rank of the parameter matrix, may not yield optimal results in dense prediction tasks. This valuable insight suggests that we should not confine the search space of the adapter solely to the low-rank subspace.

Method

According to our pilot study, we propose the VFM-Adapter as an efficient method for transferring large vision foundation models to dense prediction tasks. First, we introduce the difference between adapter tuning and full fine-tuning. Then, we detail our approach, explaining the key components and mechanisms of the VFM-Adapter. Subsequently, we conduct a careful parametric analysis to demonstrate the efficiency of our method.

Preliminary

Full Tuning. When we apply large foundation models to dense prediction tasks, full tuning means that all parameters are learnable and are changed during tuning. This process can be described by the equation:

$$L(D, \theta) = \sum_{i=1}^{N} \text{loss}\left(f_\theta\left(x_i\right), y_i\right), \tag{1}$$

$$\theta \leftarrow \arg\min_{\theta} L(D, \theta). \tag{2}$$

where $D = \{(x_i, y_i)\}_{i=1}^{N}$ denotes the dataset, f_θ represents the model forward function. The parameters θ in the model are updated by Eq. 2.

Adapter Tuning. In the training paradigm of adapter tuning, the parameters in the original backbone are frozen, and only the parameters of the additionally added adapter structure are learnable. We will refer to the parameter in the original backbone as θ_o and the parameter in the adapter as θ_a. The process of adapter tuning can be described by the following equation:

$$L(D, \theta_o, \theta_a) = \sum_{i=1}^{N} \text{loss}\left(f_{(\theta_o, \theta_a)}\left(x_i\right), y_i\right), \tag{3}$$

$$\theta_a \leftarrow \arg\min_{\theta_a} L(D, \theta_a). \tag{4}$$

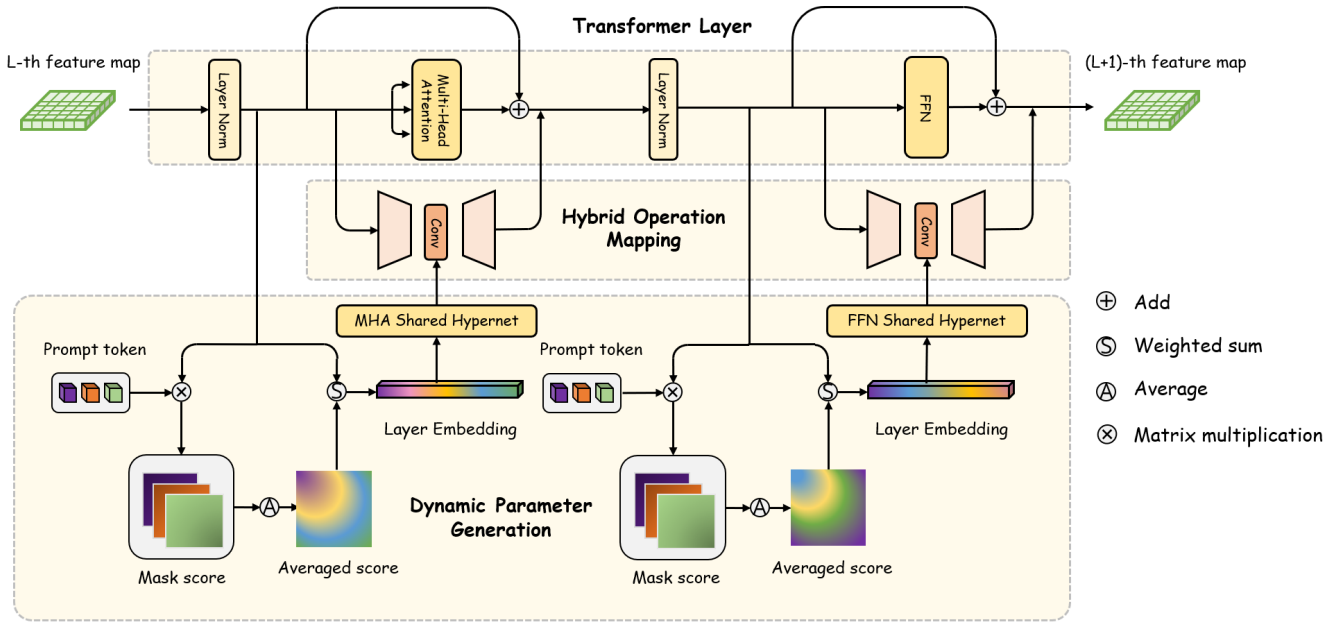

Figure 3: **Framework of VFM-Adapter.** Each VFM-Adapter contains two HOM modules in parallel with the FFN and MHA modules, respectively. Each HOM module includes hybrid linear mapping and convolutional mapping to aggregate the global and local information. We use the shared hypernet to dynamically generate the parameters of the convolution kernel. The input of the hypernet is the input-aware layer embedding, which is dynamically generated based on the input, and the hypernet outputs dynamic convolution kernel parameters for HOM.

VFM-Adapter

Building on the observations in the pilot study, the core bottleneck on VFM dense prediction lies in the insufficient representation ability of the adapter module after channel compression. We address this problem in the following two aspects. Firstly, to enhance the representation ability of ViT backbone, we introduce complex mapping relations that seamlessly combine the inductive bias and global modeling ability from convolution and attention modules. Secondly, inspired by dynamic filtering networks (Jia et al. 2016) and CondConv (Yang et al. 2019), for each sample, a controller sub-network, termed HyperNet, dynamically generates the parameters for the adapter, enabling the flexibility of our VFM-Adapter based on the input image.

Hybrid Operation Mapping. To increase the representation space, we introduce hybrid operation mapping to seamlessly combine the local inductive bias and global modeling during feature extraction. Considering the number of parameters and the nature of dense prediction tasks, we adopt convolution operation to enhance the local information omitted by attention modeling. In contrast to simple linear mappings similar to LoRA (Hu et al. 2021), convolutions enable weight sharing within the feature space, thereby effectively capturing local patterns and structures of the features. In addition, the convolution operation can introduce image-related inductive biases into the transformer-based backbone, making it more suitable for dense vision tasks. Given the input feature x^l, the entire process of the dynamic adapter can be expressed as follows:

$$x^l_{out} = ReLU(x^l_{in}W^l_d)C^lW^l_u * s, \tag{5}$$

where $W^l_d \in \mathbb{R}^{d\times r}$,$W^l_u \in \mathbb{R}^{r\times d}$,$C^l$ stands for a 3×3 convolution. s is a hyperparameter that controls how much the features in the adapter affect the original features. We insert the hybrid operation mapping adapters to the FFN (feed-forward network) module and MHA (multi-head attention) modules in the transformer layer in parallel for task-specific fine-tuning as shown in Fig. 3.

Dynamic Adapter Generation via HyperNet. In order to further expand the representation space in the adapter, we leverage the dynamic parameter mechanism via HyperNet for adapter generation. Note that in most previous methods, the parameters of adapters are independent of the input, which means the adapter lacks the ability to dynamically adjust the feature extraction manner given different samples. Based on this, we propose the dynamic adapter generation module which fully investigates the prior information from the input image and enhances the representation ability of adapters.

Specifically, we initially introduce T prompt tokens described by $\boldsymbol{E} \in \mathbb{R}^{T\times C}$. These tokens represent specific information that we want the adapter to consider during its operation. By calculating the similarity between each prompt token and every pixel point in the feature map, we obtain the importance of each pixel point:

$$M^l = softmax(F^lE_t), \tag{6}$$

where $F^l \in \mathbb{R}^{H\times W\times C}$ represents the feature map of the l-th layer, and $M^l \in \mathbb{R}^{H\times W}$ represents the importance of each pixel point on the current feature map. This similarity-based calculation allows us to assess the relevance and significance

Backbone	Method	Update Params	ATSS			SOLO			Venue
			mAP	mAP_{50}	mAP_{75}	mAP	mAP_{50}	mAP_{75}	
DINOv2	Full tuning	98.0M	53.5	73.9	58.5	43.7	67.0	46.8	-
	Fix	9.15M	43.5	67.0	47.0	38.0	62.1	39.6	-
	BitFit	9.25M	45.6	67.5	49.7	39.1	62.8	41.0	ACL'22
	LoRA	9.88M	47.3	68.8	51.7	40.0	63.5	42.5	ICLR'22
	SSF	9.35M	45.8	67.7	50.0	39.1	62.7	41.0	NeurIPS'22
	Lorand	12.15M	47.5	68.8	51.7	39.8	63.4	42.0	CVPR'23
	ARC	9.37M	49.1	68.4	50.8	39.4	63.6	41.6	NeurIPS'23
	Ours	11.93M	**51.6**	**72.0**	**56.3**	**42.1**	**66.0**	**44.9**	-
SAM	Full tuning	97.3M	43.4	61.5	47.2	35.4	55.7	37.7	-
	Fix	7.64M	31.1	46.7	36.1	27.4	44.3	28.8	-
	BitFit	7.74M	34.0	50.0	37.0	29.1	46.3	31.0	ACL'22
	LoRA	8.38M	41.2	58.1	44.9	34.5	53.6	36.9	ICLR'22
	SSF	7.85M	29.4	44.5	31.4	24.4	40.3	25.4	NeurIPS'22
	Lorand	10.68M	38.5	54.9	42.0	31.5	49.9	33.5	CVPR'23
	ARC	7.87M	37.8	54.4	41.1	32.6	51.1	35.0	NeurIPS'23
	Ours	10.42M	**43.5**	**60.7**	**47.5**	**35.9**	**55.8**	**38.7**	-
Swin	Full tuning	98.0M	44.7	63.4	48.7	38.2	60.5	40.8	-
	Fix	9.15M	34.1	55.3	35.9	31.2	52.7	31.9	-
	BitFit	9.28M	36.2	56.9	38.4	29.5	47.0	31.5	ACL'22
	LoRA	11.22M	43.3	63.4	47.2	36.6	28.7	38.7	ICLR'22
	SSF	9.42M	35.0	53.6	37.6	32.0	52.9	33.2	NeurIPS'22
	Lorand	12.23M	39.6	58.8	42.6	33.6	54.4	35.2	CVPR'23
	ARC	9.36M	41.1	61.4	44.1	35.0	57.1	36.7	NeurIPS'23
	Ours	12.17M	**44.6**	**64.3**	**48.1**	**37.4**	**59.6**	**39.7**	-

Table 1: **Main Results of Object Detection and Instance Segmentation Tasks.** We report bbox mAP and segm mAP for object detection and instance segmentation, respectively. In addition, we report the learnable parameters corresponding to each approach. Updated params contain the parameters in the neck and head.

of different areas within the feature map. By assigning importance values to individual pixel points, we can focus on the areas of the feature map that contribute the most to the desired outcome. Based on the obtained weights, we compress the original feature map through a weighted sum operation.

$$H^l = \frac{1}{T}\sum_{t=1}^{T}\sum_{i=1}^{H}\sum_{j=1}^{W} F_{ij}^l M_{tij}^l, \tag{7}$$

where H^l stands for the input-aware layer embedding, which encompasses the compressed feature map information. This embedding is then fed into the shared hypernet (Ha, Dai, and Le 2016), which generates the parameters of conv2d for each layer. The hypernet can be conceptualized as a network that generates these parameters, allowing us to obtain the specific parameters required for the current layer's conv2d operation. By leveraging the hypernet, we can dynamically and adaptively generate layer-specific parameters based on the input-aware layer embedding, enabling effective parameter customization for each layer in the network.

$$Kernal_{conv2d}^l = Hypernet(H^l), \tag{8}$$

where $H^l \in \mathbb{R}^{1\times Z}$ represents the input-aware layer embedding,$Kernal_{conv2d}^l \in \mathbb{R}^{k\times k\times C_{in}\times C_{out}}$ stands for parameters of the convolution kernel. Here k means the size of the convolution kernel, C_{in} and C_{out} denote the input and output channels of the convolution kernel that are equivalent to r. Hypernet contains two linear layers like (Ha, Dai, and Le 2016), which transform the input-aware layer embedding into the desired conv parameters.

During training, the original backbone model's parameters are frozen, and only the adapter's parameters are updated. Through the described designs, we expand the parameter search space in the adapter, enabling easier transfer of pre-trained knowledge to downstream tasks. The shared hypernet is crucial, generating parameters for each conv2d layer based on input-aware embeddings from the layer's feature map. By sharing the hypernet across layers, we ensure consistent parameter generation and promote knowledge transfer. This method enables efficient adapter adaptation while preserving valuable information from the original backbone.

Method	DINOv2			SAM			SWIN		
	mIoU	mAcc	aAcc	mIoU	mAcc	aAcc	mIoU	mAcc	aAcc
Full tuning	52.8	64.0	84.5	42.9	53.2	81.0	45.8	57.4	81.7
Fix	50.2	61.9	83.0	25.0	33.1	72.0	36.7	47.3	75.5
BitFit	50.8	62.4	83.4	29.8	39.5	75.3	41.2	52.2	78.4
LoRA	50.9	62.6	83.5	40.0	51.2	80.2	42.4	54.2	79.3
SSF	51.2	63.1	83.5	36.7	40.1	78.6	41.5	53.1	78.6
Lorand	50.6	62.3	83.5	**41.5**	**53.0**	**80.7**	43.4	55.4	80.0
ARC	50.9	63.0	83.4	38.5	49.4	79.6	42.1	53.6	79.0
Ours	**52.1**	**63.6**	**84.3**	40.7	51.9	80.6	**44.2**	**55.8**	**80.4**

Table 2: **Main Results of Semantic Segmentation.** We report three metrics in our evaluation: mIoU (mean Intersection over Union), mAcc (mean Accuracy), and aAcc (average Accuracy).

Parameter Analysis

In this section, we will conduct an analysis of the parameter count in our method to demonstrate that our incorporation of the hypernet not only enables input-aware parameters but also leads to a reduction in the overall number of parameters.

We define d as the input dimension in the adapter, r as the middle dimension, and the transformer has a total of l layers. When we don't use hypernet, the number of parameters in the adapter section can be written as:

$$2l(2dr + 9r^2), \tag{9}$$

where $9r^2$ denotes the parameters of the 3*3 convolution. The space complexity of our method can be abbreviated as $O(r^2)$. After we use the shared hypernet to generate the parameters of the convolution kernel, the number of parameters in the adapter section can be written as:

$$2(2ldr + 9zr + z^2r). \tag{10}$$

The space complexity of our method can be abbreviated as $O(r)$. By introducing a shared hypernet, we not only realize the input-aware parameter design but also reduce the space complexity of the adapter from $O(r^2)$ to $O(r)$.

Experiments

We conduct comprehensive experiments on mainstream dense prediction tasks to demonstrate the effectiveness and advantages of our method, including instance segmentation, object detection, and semantic segmentation. We also organize a wealth of ablation experiments to help better understand our methodology.

HOM	DPG (input-independent)	DPG (input-aware)	mAP
×	×	×	40.1
✓	×	×	42.2
✓	✓	×	42.2
✓	×	✓	**43.5**

Table 3: **Ablation on Main Components**. We use SAM-Base as the backbone to validate the effectiveness of the different components. HOM stands for Hybrid Operation Mapping and DPG represents Dynamic Parameter Generation.

Experimental Settings

Dataset. We conduct extensive experiments on the COCO (Lin et al. 2014) dataset and the ADE20K (Zhou et al. 2019) dataset. The COCO dataset comprises 118k training images and 5k validation images, which can be used for object detection and instance segmentation. For the object detection task, we use ATSS (Zhang et al. 2020) as our detector and for instance segmentation, we use SOLO (Wang et al. 2020) as our framework. ADE20K is a widely used semantic segmentation benchmark, including 20k training and 2k validation images. We use the SegFormer (Xie et al. 2021) as the framework for semantic segmentation.

Pretrained VFM. We select three representative VFM, namely DINOv2 (Oquab et al. 2023), SAM (Kirillov et al. 2023), and Swin Transformer (Liu et al. 2021). Code and pre-trained models are from MMPretrain (Contributors 2023). If not specified, we use the base version for our experiments.

Implementation Details. As the pre-trained large models used in our study *e.g.,* DINOv2 and SAM are based on traditional ViT (Dosovitskiy et al. 2020) architectures, they lack inherent multi-scale features commonly used in dense prediction tasks. To address this limitation, we adopt a similar approach to ViTDet (Li et al. 2022), where we construct a feature pyramid based on the last layer of features extracted from the pre-trained models. This allows us to incorporate multi-scale information into the models, enhancing their performance on diverse visual tasks. All our experiments are conducted with 8× NVIDIA Tesla A100 GPUs.

Baselines. We compare our approach with current state-of-the-art methods.

- Full tuning: Update all parameters in the framework.
- Fix: Fix the backbone and only update other parameters.
- BitFit: Only tune the bias parameters in the backbone and other parameters are fixed.

- LoRA: Inject trainable rank decomposition matrices into each layer of the Transformer architecture.
- ARC: Leverage shared symmetric down/up projections to construct bottleneck operations and learn low-dimensional re-scaling coefficients to re-compose layer-adaptive adapters.
- SSF: Scale and shift the deep features extracted by the pre-trained model after every operation.
- Lorand: Add LoRand layers after the MHA/FFN layers of each block.

Main Results

The zero-shot ability of VFM is not as good as claimed. Contrary to claims, VFM demonstrates limited generalizability when apply directly to downstream tasks without fine-tuning. By freezing the VFM backbone and training only the neck and head sections, we observe a notable decline in performance across various tasks. Specifically, in object detection, freezing the VFM backbone results in more than a 10-point reduction in mean Average Precision (mAP). Similarly, in instance segmentation and semantic segmentation tasks, freezing the backbone leads to varying degrees of performance degradation across different VFM. These findings suggest that the zero-shot capabilities of pre-trained VFM on downstream tasks are not as robust as claimed. While fully fine-tuning VFM shows enhanced performance, the associated high costs have been a deterrent.

Low-rank-based designs are not suitable for dense prediction tasks. LoRA and ARC are parameter-efficient fine-tuning methods based on low-rank property. We enlarge their intermediate dimensions for better performance with comparable trainable parameters. ARC's shared projection matrix keeps parameter count low despite dimension increase. Our method outperforms low-rank based methods (Tables 1 and 2). Low-rank structures may not suit dense prediction tasks. Our method adds visual domain-specific bias to low-rank structures and uses hypernet for input-aware parameter generation, proving more suitable for dense prediction tasks.

Our approach acquires a balance between training parameters and performance. BitFit and SSF are not designed for dense prediction tasks, and even though these methods have smaller trainable parameters compared to our method, their performance is far inferior to our method. Regarding LoRA and Lorand, we observe that increasing the intermediate dimension can lead to performance improvement. However, the associated increase in parameter count is unacceptable for our purposes. In contrast, our approach strikes a balance between the number of parameters and performance. When using SAM as the backbone, we exceed the effect of full fine-tuning on the object detection and instance segmentation tasks by using only about 3% of the number of parameters of the full backbone. When using other backbones on these three classical dense prediction tasks, our method also achieves notable results, beating all other methods.

Method	Param	GPU Memory	Inference Latency	Training Time	mAP
Full tuning	663.1M	63G	3.8FPS	268h	51.5
Fix	11.7M	5G	3.8FPS	76h	32.9
BitFit	12.2M	19G	3.8FPS	150h	42.3
LoRA	44.5M	33G	3.6FPS	160h	49.2
SSF	12.6M	24G	3.8FPS	172h	42.9
Lorand	64.8M	31G	3.5FPS	254h	50.4
ARC	13.1M	20G	3.6FPS	163h	47.2
Ours	39.5M	31G	3.6FPS	198h	**52.6**

Table 4: Comparison of parameters, GPU memory, inference time, training time, and COCO box mAP on ViT-Huge.

Ablation Studies

We perform ablation experiments on our method to investigate what properties are useful and observe several intriguing properties.

Ablation on main components. To assess our method's components, we conduct ablation experiments with different setups: without using HOM, with HOM, and with a hypernet generating HOM parameters. We also explore the impact of input-aware HOM parameters by comparing an input-aware hypernet with learnable layer-wise embeddings alone. Results in Table 3 show that using HOM yields roughly a 2 mAP increase compared to its absence, likely due to its ability to introduce image-related biases to the transformer-based backbone. By employing a hypernet for HOM parameters, we reduce parameter count while maintaining performance. Additionally, an input-aware hypernet further boosts performance.

Comparisons with Huge Backbone

To further validate the effectiveness and efficiency of our method, we provide more dimensional comparisons on a larger backbone. We compare our method with other state-of-the-art methods in terms of parameters, GPU memory usage, inference time, training time, and COCO box mAP. As shown in Table 4, our approach reduces the number of parameters by about 94%, the amount of GPU memory by 51%, and the training time by 26% compared to full fine-tuning, but achieves almost the same performance.

Conclusion

In this paper, we propose VFM-Adapter, a novel PEFT method. It effectively transfers pre-trained VFM to dense prediction tasks. Experiments show our approach outperforms other PEFT methods and full fine-tuning in representative dense prediction tasks. We hope our work inspires others and advances the PEFT field.

Acknowledgments

This work was supported by the Anhui Provincial Natural Science Foundation under Grant 2108085UD12. We acknowledge the support of GPU cluster built by MCC Lab of Information Science and Technology Institution, USTC.

References

Carion, N.; Massa, F.; Synnaeve, G.; Usunier, N.; Kirillov, A.; and Zagoruyko, S. 2020. End-to-end object detection with transformers. In *European Conference on Computer Vision*, 213–229. Springer.

Chen, L.; Li, J.; Dong, X.; Zhang, P.; He, C.; Wang, J.; Zhao, F.; and Lin, D. 2023. ShareGPT4V: Improving large multi-modal models with better captions. *arXiv preprint arXiv:2311.12793*.

Chen, L.-C.; Papandreou, G.; Schroff, F.; and Adam, H. 2017. Rethinking atrous convolution for semantic image segmentation. *arXiv preprint arXiv:1706.05587*.

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.

Chen, Y.; Dai, X.; Liu, M.; Chen, D.; Yuan, L.; and Liu, Z. 2020. Dynamic convolution: Attention over convolution kernels. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 11030–11039.

Chen, Z.; Yang, C.; Chang, J.; Zhao, F.; Zha, Z.-J.; and Wu, F. 2024. DDOD: Dive deeper into the disentanglement of object detector. *IEEE Transactions on Multimedia*, 26: 284–298.

Cheng, B.; Misra, I.; Schwing, A. G.; Kirillov, A.; and Girdhar, R. 2022. Masked-attention mask transformer for universal image segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 1290–1299.

Contributors, M. 2023. OpenMMLab's Pre-training Toolbox and Benchmark. https://github.com/open-mmlab/mmpretrain.

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. 2023. Efficient adaptation of large vision transformer via adapter recomposing. *arXiv preprint arXiv:2310.06234*.

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*.

Fang, Y.; Sun, Q.; Wang, X.; Huang, T.; Wang, X.; and Cao, Y. 2023. Eva-02: A visual representation for neon genesis. *arXiv preprint arXiv:2303.11331*.

Floridi, L.; and Chiriatti, M. 2020. GPT-3: Its nature, scope, limits, and consequences. *Minds and Machines*, 30: 681–694.

Guizilini, V.; Vasiljevic, I.; Chen, D.; Ambruș, R.; and Gaidon, A. 2023. Towards zero-shot scale-aware monocular depth estimation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 9233–9243.

Ha, D.; Dai, A.; and Le, Q. V. 2016. HyperNetworks. *arXiv e-prints*, arXiv:1609.09106.

He, K.; Chen, X.; Xie, S.; Li, Y.; Dollár, P.; and Girshick, R. 2022. Masked autoencoders are scalable vision learners. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 16000–16009.

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

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

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.

Jia, X.; De Brabandere, B.; Tuytelaars, T.; and Gool, L. V. 2016. Dynamic filter networks. *Advances in Neural Information Processing Systems*, 29.

Jie, S.; and Deng, Z.-H. 2022. Convolutional bypasses are better vision transformer adapters. *arXiv preprint arXiv:2207.07039*.

Kirillov, A.; Mintun, E.; Ravi, N.; Mao, H.; Rolland, C.; Gustafson, L.; Xiao, T.; Whitehead, S.; Berg, A. C.; Lo, W.-Y.; et al. 2023. Segment anything. *arXiv preprint arXiv:2304.02643*.

Li, Y.; Mao, H.; Girshick, R.; and He, K. 2022. Exploring plain vision transformer backbones for object detection. In *European Conference on Computer Vision*, 280–296. Springer.

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

Lin, T.-Y.; Goyal, P.; Girshick, R.; He, K.; and Dollár, P. 2017. Focal loss for dense object detection. In *Proceedings of the IEEE International Conference on Computer Vision*, 2980–2988.

Lin, T.-Y.; Maire, M.; Belongie, S.; Hays, J.; Perona, P.; Ramanan, D.; Dollár, P.; and Zitnick, C. L. 2014. Microsoft coco: Common objects in context. In *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13*, 740–755. Springer.

Liu, Z.; Lin, Y.; Cao, Y.; Hu, H.; Wei, Y.; Zhang, Z.; Lin, S.; and Guo, B. 2021. Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 10012–10022.

Long, J.; Shelhamer, E.; and Darrell, T. 2015. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 3431–3440.

Ma, J.; Lei, X.; Liu, N.; Zhao, X.; and Pu, S. 2022. Towards comprehensive representation enhancement in semantics-guided self-supervised monocular depth estimation. In *European Conference on Computer Vision*, 304–321. Springer.

Ma, N.; Zhang, X.; Huang, J.; and Sun, J. 2020. Weightnet: Revisiting the design space of weight networks. In *European Conference on Computer Vision*, 776–792. Springer.

Mi, Z.; Di, C.; and Xu, D. 2022. Generalized binary search network for highly-efficient multi-view stereo. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 12991–13000.

Oquab, M.; Darcet, T.; Moutakanni, T.; Vo, H.; Szafraniec, M.; Khalidov, V.; Fernandez, P.; Haziza, D.; Massa, F.; El-Nouby, A.; et al. 2023. Dinov2: Learning robust visual features without supervision. *arXiv preprint arXiv:2304.07193*.

Patil, V.; Sakaridis, C.; Liniger, A.; and Van Gool, L. 2022. P3depth: Monocular depth estimation with a piecewise planarity prior. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 1610–1621.

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

Ren, S.; He, K.; Girshick, R.; and Sun, J. 2015. Faster r-cnn: Towards real-time object detection with region proposal networks. *Advances in Neural Information Processing Systems*, 28.

Tian, Z.; Shen, C.; Chen, H.; and He, T. 2019. Fcos: Fully convolutional one-stage object detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 9627–9636.

Touvron, H.; Martin, L.; Stone, K.; Albert, P.; Almahairi, A.; Babaei, Y.; Bashlykov, N.; Batra, S.; Bhargava, P.; Bhosale, S.; et al. 2023. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.

Wang, C.-Y.; Bochkovskiy, A.; and Liao, H.-Y. M. 2023. YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 7464–7475.

Wang, X.; Kong, T.; Shen, C.; Jiang, Y.; and Li, L. 2020. Solo: Segmenting objects by locations. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XVIII 16*, 649–665. Springer.

Xiao, T.; Liu, Y.; Zhou, B.; Jiang, Y.; and Sun, J. 2018. Unified perceptual parsing for scene understanding. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 418–434.

Xie, E.; Wang, W.; Yu, Z.; Anandkumar, A.; Alvarez, J. M.; and Luo, P. 2021. SegFormer: Simple and efficient design for semantic segmentation with transformers. *Advances in Neural Information Processing Systems*, 34: 12077–12090.

Yang, B.; Bender, G.; Le, Q. V.; and Ngiam, J. 2019. Condconv: Conditionally parameterized convolutions for efficient inference. *Advances in neural information processing systems*, 32.

Yang, Z.; Ren, Z.; Shan, Q.; and Huang, Q. 2022. Mvs2d: Efficient multi-view stereo via attention-driven 2d convolutions. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 8574–8584.

Yin, D.; Yang, Y.; Wang, Z.; Yu, H.; Wei, K.; and Sun, X. 2023. 1% VS 100%: Parameter-efficient low rank adapter for dense predictions. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 20116–20126.

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*.

Zeng, A.; Liu, X.; Du, Z.; Wang, Z.; Lai, H.; Ding, M.; Yang, Z.; Xu, Y.; Zheng, W.; Xia, X.; et al. 2022. Glm-130b: An open bilingual pre-trained model. *arXiv preprint arXiv:2210.02414*.

Zhang, S.; Chi, C.; Yao, Y.; Lei, Z.; and Li, S. Z. 2020. Bridging the gap between anchor-based and anchor-free detection via adaptive training sample selection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 9759–9768.

Zhong, Z.; Tang, Z.; He, T.; Fang, H.; and Yuan, C. 2024. Convolution meets LoRA: Parameter efficient finetuning for segment anything model. *arXiv preprint arXiv:2401.17868*.

Zhou, B.; Zhao, H.; Puig, X.; Xiao, T.; Fidler, S.; Barriuso, A.; and Torralba, A. 2019. Semantic understanding of scenes through the ade20k dataset. *International Journal of Computer Vision*, 127: 302–321.

Zou, X.; Dou, Z.-Y.; Yang, J.; Gan, Z.; Li, L.; Li, C.; Dai, X.; Behl, H.; Wang, J.; Yuan, L.; et al. 2023. Generalized decoding for pixel, image, and language. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 15116–15127.