VIT-ADAPTER: EXPLORING PLAIN VISION TRANSFORMER FOR ACCURATE DENSE PREDICTIONS

Anonymous authors
Paper under double-blind review

ABSTRACT

This work investigates a simple yet powerful dense prediction task adapter for Vision Transformer (ViT). Unlike recently advanced variants that incorporate vision-specific inductive biases into their architectures, the plain ViT suffers inferior performance on dense predictions due to weak prior assumptions. To address this issue, we propose the ViT-Adapter, which allows plain ViT to achieve comparable performance to vision-specific transformers. Specifically, the backbone in our framework is a plain ViT that can learn powerful representations from large-scale multi-modal data. When transferring to downstream tasks, a **pre-training-free adapter** is used to introduce the image-related inductive biases into the model, making it suitable for these tasks. We verify ViT-Adapter on multiple dense prediction tasks, including object detection, instance segmentation, and semantic segmentation. Notably, without using extra detection data, our ViT-Adapter-L yields state-of-the-art **60.9** box AP and **53.0** mask AP on COCO test-dev. We hope that the ViT-Adapter could serve as an alternative for vision-specific transformers and facilitate future research. The code and models will be released.

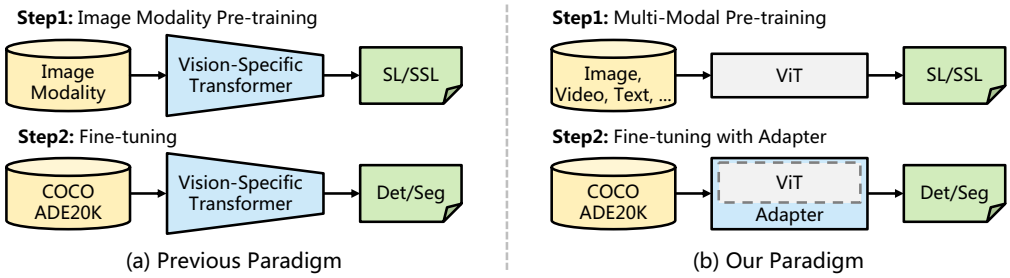
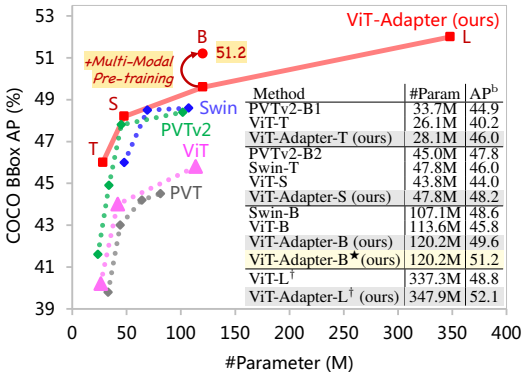


Figure 1: **Previous paradigm *vs.* our paradigm.** (a) Previous paradigm designs vision-specific models and pre-trains on large-scale image datasets via supervised learning (SL) or self-supervised learning (SSL) and then fine-tunes them on downstream tasks. (b) We propose a pre-training-free adapter to close the performance gap between the plain ViT (Dosovitskiy et al., 2020) and vision-specific transformers (*e.g.*, Swin Transformer (Liu et al., 2021b), PVT (Wang et al., 2021)) for dense prediction tasks. Compared to the previous paradigm, our method preserves the flexibility of ViT and thus could benefit from advanced multi-modal pre-training.

1 INTRODUCTION

Recently, transformers have witnessed remarkable success in a broad range of computer vision fields. Benefiting from the dynamic modeling capability and the long-range dependence of the attention mechanism, various vision transformers (Dosovitskiy et al., 2020; Chen et al., 2021; Han et al., 2021; Li et al., 2021c; Wu et al., 2022b) soon rose in many computer vision tasks such as object detection and semantic segmentation, surpassing CNN models and reaching state-of-the-art performance. These models are mainly divided into two families, *i.e.* the plain ViT (Dosovitskiy et al., 2020; Touvron et al., 2021), and its hierarchical variants (Dong et al., 2021; Liu et al., 2021b; Wang et al., 2021; 2022a). In general, the latter can produce better results and is believed to introduce vision-specific inductive biases into their architectures by using local spatial operations.



Method	#Param	AP^b
PVTv2-B1	33.7M	44.9
ViT-T	26.1M	40.2
ViT-Adapter-T (ours)	28.1M	46.0
PVTv2-B2	45.0M	47.8
Swin-T	47.8M	46.0
ViT-S	43.8M	44.0
ViT-Adapter-S (ours)	47.8M	48.2
Swin-B	107.1M	48.6
ViT-B	113.6M	45.8
ViT-Adapter-B (ours)	120.2M	49.6
ViT-Adapter-B★(ours)	120.2M	51.2
ViT-L†	337.3M	48.8
ViT-Adapter-L† (ours)	347.9M	52.1

Figure 2: **Object detection performance on COCO val2017 using Mask R-CNN.** We see that the proposed ViT-Adapter brings significant improvements to plain ViTs. ★ indecates using multi-modal pre-trained ViT from (Zhu et al., 2021). Backbones pre-trained on ImageNet-22K are marked with †, otherwise ImageNet-1K.

Nonetheless, the plain ViT (*i.e.*, vanilla transformer) still has some nonnegligible advantages. A typical example lies in multi-modal pre-training (Zhu et al., 2021; 2022; Wang et al., 2022b). Stemming from the natural language processing (NLP) field, transformer has no assumption of input data. Equipping with different tokenizers, *e.g.* patch embedding (Dosovitskiy et al., 2020), 3D patch embedding (Liu et al., 2021c), and token embedding (Vaswani et al., 2017), vanilla transformers such as *plain ViT can use massive multi-modal data for pre-training*, including image, video, and text, which encourages the model to learn semantic-rich representations. However, the plain ViT has conclusive defects in dense predictions compared to vision-specific transformers. Lacking image-related prior knowledge results in slower convergence and lower performance, and thus plain ViTs are hard to compete with vision-specific transformers (Huang et al., 2021b; Xie et al., 2021; Wang et al., 2022a) on dense prediction tasks. Inspired by the adapters (Houlsby et al., 2019; Stickland & Murray, 2019) in the NLP field, *this work aims to develop an adapter to close the performance gap between the plain ViT and vision-specific backbones for dense prediction tasks.*

To this end, we propose the Vision Transformer Adapter (ViT-Adapter), which is a *pre-training-free* additional network that can efficiently adapt the plain ViT to downstream dense prediction tasks without modifying its original architecture. Specifically, to introduce the vision-specific inductive biases into the plain ViT, we design three tailored modules for ViT-Adapter, including (1) a spatial prior module for capturing the local semantics (spatial prior) from input images, (2) a spatial feature injector for incorporating spatial prior into the ViT, and (3) a multi-scale feature extractor to reconstruct the multi-scale features required by dense prediction tasks.

As shown in Figure 1, compared to the previous paradigm that pre-trains on large-scale image datasets (*e.g.*, ImageNet (Deng et al., 2009)) then fine-tunes on other tasks, our paradigm is more flexible. In our framework, the backbone network is a general-propose model (*e.g.*, plain ViT) that can be pre-trained with not only images but also multi-modal data. For the transfer learning of dense prediction tasks, we use a randomly initialized adapter to introduce the image-related prior knowledge (inductive biases) into the pre-trained backbone, making the model suitable for these tasks. In this way, using ViT as the backbone, our framework achieves comparable or even better performance than vision-specific transformers such as Swin (Liu et al., 2021b).

Our main contributions are as follows:

• We explore a new paradigm to introduce vision-specific inductive biases into the plain ViT. It helps ViT achieve on-par performance to recent transformer variants (Liu et al., 2021b; Wang et al., 2022a) with regular ImageNet pre-training and further benefits from advanced multi-modal pre-training.

• We design a spatial prior module and two feature interaction operations, to inject the image prior without redesigning the architecture of ViT. They can supplement the missing local information and reorganize fine-grained multi-scale features for dense prediction tasks.

• We evaluate the ViT-Adapter on multiple challenging benchmarks, including COCO (Lin et al., 2014) and ADE20K (Zhou et al., 2017). As shown in Figure 2, our models consistently achieve improved performance compared to the prior arts *under the fair pre-training strategy*. For instance, when using only ImageNet-1K pre-training, ViT-Adapter-B reports 49.6 box AP on COCO val, outperforming Swin-B by 1.0 points. Benefiting from multi-modal pre-training (Peng et al., 2022), our ViT-Adapter-L yields 60.9 box AP, which is the best record on COCO test-dev without training on extra detection data such as Object365 (Shao et al., 2019).

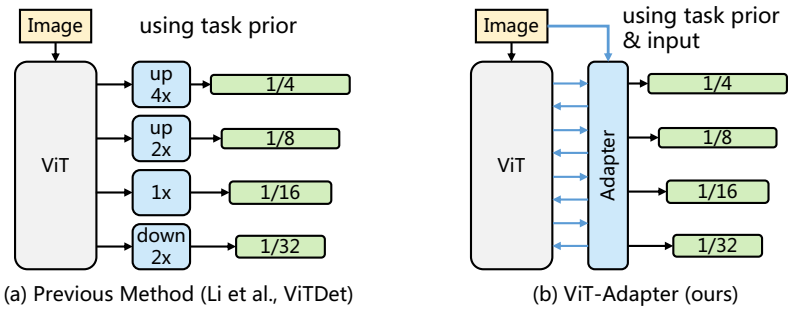


Figure 3: **Overview of our ViT-Adapter and two related approaches.** Li et al. (2021b) and ViTDet (Li et al., 2022b) build simple feature pyramid to adapt plain ViT for object detection, which only consider task prior. Differently, our adapter utilizes both task prior and the input image.

2 Related Work

Transformers. In recent years, transformers have dominated various tasks across multiple modalities, such as natural language processing, computer vision, and speech recognition. The vanilla transformer (Vaswani et al., 2017) was initially proposed for machine translation and remains the state-of-the-art architecture for NLP tasks today. ViT (Dosovitskiy et al., 2020) is the first work to generalize the vanilla transformer to the image classification task without much modification. PVT (Wang et al., 2021) and Swin (Liu et al., 2021b) introduce more vision-specific inductive biases by incorporating the pyramid structure from CNNs. Afterward, Conformer (Peng et al., 2021) proposed the first dual network to combine CNN with transformer. Recently, BEiT (Bao et al., 2022) and MAE (He et al., 2021) extended the scope of ViT to self-supervised learning with masked image modeling (MIM), demonstrating the powerful potential of the plain ViT architecture. Many works (Li et al., 2021b; Zhu et al., 2021; 2022; Wang et al., 2022b) have shown that designing vision-specific models is an important direction, but the general-propose architectures (*e.g.*, plain ViT) are more flexible and essential for masked data modeling and multi-modal pre-training. Therefore, we develop a pre-training-free adapter to introduce the image prior without modifying the architecture of ViT, preserving its flexibility and enjoying advanced multi-modal pre-training.

Decoders for ViT. The architecture for dense prediction commonly follows an encoder-decoder pattern, in which the encoder generates rich features and the decoder aggregates and translates them to the final predictions. Recently, illuminated by the global receptive fields of ViT, many works employ it as the encoder and design task-specific decoders. SETR (Zheng et al., 2021) is the first work to adopt ViT as the backbone and develop several CNN decoders for semantic segmentation. Segmenter (Strudel et al., 2021) also extends ViT to semantic segmentation, but differs in that it equips a transformer-based decoder. DPT (Ranftl et al., 2021) further applies ViT to the monocular depth estimation task via a CNN decoder and yields remarkable improvements. In summary, these works improve the dense prediction performance of ViT by designing modality- and task-specific decoders, but remain ViT's weakness of single-scale and low-resolution representation.

Adapters. To date, adapters have been widely used in the NLP field. PALs (Stickland & Murray, 2019) and Adapters (Houlsby et al., 2019) introduce new modules in transformer encoders for task-specific fine-tuning, making the pre-trained model quickly adapt to downstream NLP tasks. In the field of computer vision, some adapters have been proposed for incremental learning (Rosenfeld & Tsotsos, 2018) and domain adaptation (Rebuffi et al., 2017; 2018). With the advent of CLIP (Radford et al., 2021), many CLIP-based adapters (Gao et al., 2021; Sung et al., 2021; Zhang et al., 2021) were presented to transfer pre-trained knowledge to zero-shot or few-shot downstream tasks. Recently, Li et al. (2021b) and ViTDet (Li et al., 2022b) employed some upsampling and downsampling modules to adapt the plain ViT for object detection, as shown in Figure 3. However, under regular training settings (*i.e.*, apply ImageNet supervised pre-training and fine-tune for 36 epochs), their detection performance is still inferior[1] to recent models (Chu et al., 2021b; Dong et al., 2021; Wang et al., 2022a; Wu et al., 2022b) that well combine image prior. Therefore, it is still challenging to design a powerful dense prediction task adapter for ViT.

[1]In ViTDet, using regular ImageNet-22K pre-training instead of MAE (He et al., 2021) drops 4.0 AP^b.

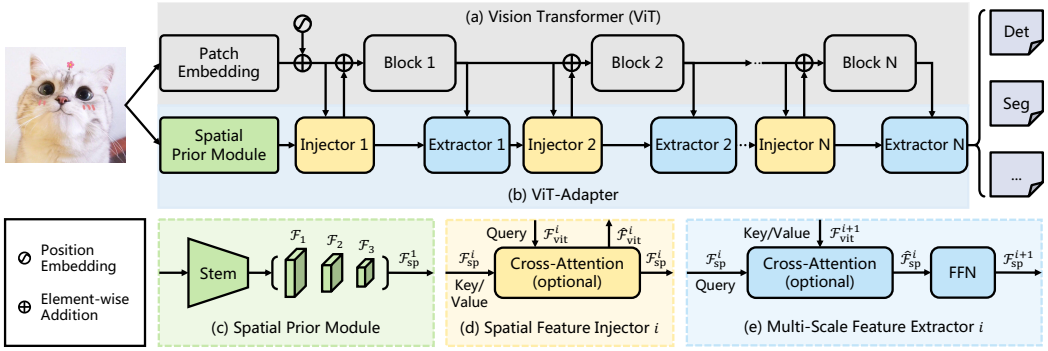


Figure 4: **Overall architecture of ViT-Adapter.** (a) The ViT, whose encoder layers are divided into N (usually $N = 4$) equal blocks for feature interaction. (b) Our ViT-Adapter, which contains three key designs, including (c) a spatial prior module for modeling local spatial contexts from the input image, (d) a spatial feature injector for introducing spatial priors into the ViT, and (e) a multi-scale feature extractor for reconstructing multi-scale features from the single-scale features of ViT.

3 Vision Transformer Adapter

3.1 Overall Architecture

As illustrated in Figure 4, our model can be divided into two parts. The first part is the plain ViT (Dosovitskiy et al., 2020) that consists of a patch embedding followed by L transformer encoder layers (see Figure 4(a)). The second part is the proposed ViT-Adapter as shown in Figure 4(b), which contains (1) a spatial prior module to capture spatial features from the input image, (2) a spatial feature injector to inject spatial priors into the ViT, and (3) a multi-scale feature extractor to extract hierarchical features from the single-scale features of ViT.

For the ViT, the input image is first fed into the patch embedding, where the image is divided into 16×16 non-overlapping patches. After that, these patches are flattened and projected to D-dimensional tokens, and the feature resolution is reduced to 1/16 of the original image. Then, these tokens added with the position embedding, are passed through L encoder layers.

For the ViT-Adapter, we first feed the input image into the spatial prior module. D-dimensional spatial features of three target resolutions (*i.e.*, 1/8, 1/16, and 1/32) will be collected. Then, these feature maps are flattened and concatenated as the input for feature interaction. Specifically, given the number of interactions N (usually $N = 4$), we evenly split the transformer encoders of ViT into N blocks, each containing L/N encoder layers. For the i-th block, we first inject spatial priors $\mathcal{F}_{\text{sp}}^{i}$ into the block via a spatial feature injector, and then extract hierarchical features from the output of the block by a multi-scale feature extractor. After N feature interactions, we obtain high-quality multi-scale features, and then we split and reshape the features into three target resolutions 1/8, 1/16, and 1/32. Finally, we build the 1/4-scale feature map by upsampling the 1/8-scale feature map using a 2×2 transposed convolution. In this way, we obtain a feature pyramid of similar resolutions to ResNet (He et al., 2016), which can be used in various dense prediction tasks.

3.2 Spatial Prior Module

Recent studies (Wang et al., 2022a; Wu et al., 2021; Fang et al., 2022b; Park & Kim, 2022) show convolutions can help transformers better capture the local spatial information. Inspired by this, we introduce the *Spatial Prior Module* (SPM). It is designed to model the local spatial contexts of images parallel with the patch embedding layer, so as not to alter the original architecture of ViT.

As shown in Figure 4(c), a standard convolutional stem borrowed from ResNet (He et al., 2016) is employed, which consists of three convolutions and a max-pooling layer. Then, we use a stack of stride-2 3×3 convolutions to double the number of channels and reduce the size of feature maps. Finally, several 1×1 convolutions are applied at the end to project the feature maps to D dimensions. In this way, we obtain a feature pyramid $\{\mathcal{F}_1, \mathcal{F}_2, \mathcal{F}_3\}$, which contains D-dimensional feature maps

with resolutions of 1/8, 1/16, and 1/32. Then, we flatten and concatenate these feature maps into feature tokens $\mathcal{F}_{\text{sp}}^{1} \in \mathbb{R}^{(\frac{HW}{8^2}+\frac{HW}{16^2}+\frac{HW}{32^2})\times D}$, as the input for feature interaction.

3.3 Feature Interaction

Due to weak prior assumptions, the plain ViT suffers sub-optimal performance on dense prediction tasks compared to vision-specific transformers (Chu et al., 2021a; Dong et al., 2021; Liu et al., 2021b; Wang et al., 2022a). To alleviate this issue, we propose two feature interaction modules to bridge the feature maps of our SPM and the ViT. To be specific, the two modules are mainly based on cross-attention, namely *Spatial Feature Injector* and *Multi-Scale Feature Extractor*.

Spatial Feature Injector. As shown in Figure 4(d), this module is used to inject the spatial priors into ViT. Specifically, for the i-th block of the ViT, we take the input feature $\mathcal{F}_{\text{vit}}^{i} \in \mathbb{R}^{\frac{HW}{16^2}\times D}$ as the query, and the spatial feature $\mathcal{F}_{\text{sp}}^{i} \in \mathbb{R}^{(\frac{HW}{8^2}+\frac{HW}{16^2}+\frac{HW}{32^2})\times D}$ as the key and value. We use cross-attention to inject spatial feature $\mathcal{F}_{\text{sp}}^{i}$ into the input feature $\mathcal{F}_{\text{vit}}^{i}$, which can be written as Eqn. 1.

$$\hat{\mathcal{F}}_{\text{vit}}^{i} = \mathcal{F}_{\text{vit}}^{i} + \gamma^{i}\text{Attention}(\text{norm}(\mathcal{F}_{\text{vit}}^{i}), \text{norm}(\mathcal{F}_{\text{sp}}^{i})), \tag{1}$$

where the $\text{norm}(\cdot)$ is LayerNorm (Ba et al., 2016), and the attention layer $\text{Attention}(\cdot)$ suggests using sparse attention. In addition, we apply a learnable vector $\gamma^{i} \in \mathbb{R}^{D}$ to balance the attention layer's output and the input feature $\mathcal{F}_{\text{vit}}^{i}$, which is initialized with $\mathbf{0}$. This initialization strategy ensures that the feature distribution of $\mathcal{F}_{\text{vit}}^{i}$ will not be modified drastically due to the injection of spatial priors, thus making better use of the pre-trained weights of ViT.

Multi-Scale Feature Extractor. After injecting the spatial priors into the ViT, we obtain the output feature $\mathcal{F}_{\text{vit}}^{i+1}$ by passing $\hat{\mathcal{F}}_{\text{vit}}^{i}$ through the encoder layers of the i-th block. Then, we apply a module consisting of a cross-attention layer and a feed-forward network (FFN), to extract multi-scale features, as shown in Figure 4(e). This process can be formulated as:

$$\mathcal{F}_{\text{sp}}^{i+1} = \hat{\mathcal{F}}_{\text{sp}}^{i} + \text{FFN}(\text{norm}(\hat{\mathcal{F}}_{\text{sp}}^{i})), \tag{2}$$

$$\hat{\mathcal{F}}_{\text{sp}}^{i} = \mathcal{F}_{\text{sp}}^{i} + \text{Attention}(\text{norm}(\mathcal{F}_{\text{sp}}^{i}), \text{norm}(\mathcal{F}_{\text{vit}}^{i+1})), \tag{3}$$

in which we use the spatial feature $\mathcal{F}_{\text{sp}}^{i} \in \mathbb{R}^{(\frac{HW}{8^2}+\frac{HW}{16^2}+\frac{HW}{32^2})\times D}$ as the query, and the output feature $\mathcal{F}_{\text{vit}}^{i+1} \in \mathbb{R}^{\frac{HW}{16^2}\times D}$ as the key and value for cross-attention. As same as the spatial feature injector, we adopt sparse attention here to reduce computational cost. The generated spatial feature $\mathcal{F}_{\text{sp}}^{i+1}$ will be used as the input of the next spatial feature injector.

3.4 Architecture Configurations

We build our ViT-Adapter for 4 different sizes of ViT, including ViT-T, ViT-S, ViT-B, and ViT-L. For these models, the parameter numbers of our adapters are 2.5M, 5.8M, 14.0M, and 23.7M, respectively. We employ deformable attention (Zhu et al., 2020) as the default sparse attention in our method, where the number of sampling points is fixed to 4, and the number of attention heads is set to 6, 6, 12, and 16. The number of interactions N is 4. Besides, we set the FFN ratio in our adapter to 0.25 to save computational overhead, *i.e.* the hidden sizes of FFN are 48, 96, 192, and 256 for 4 different adapters. More details of each configuration are shown in Table 10 in Appendix A.2.

4 Experiments

Previous work (Wang et al., 2021) has shown that the pyramid prior is beneficial to dense prediction, but brings little gains to image classification. Therefore, in this study, we focus on how to better adapt readily available pre-trained ViTs to dense prediction tasks. We hope this method will also help decouple the model design of upstream pre-training and downstream fine-tuning.

4.1 Object Detection And Instance Segmentation

Settings. Our detection experiments are based on MMDetection (Chen et al., 2019b) and the COCO (Lin et al., 2014) dataset. We use 4 mainstream detectors to evaluate our ViT-Adapter, including

Method	#Param (M)	Mask R-CNN 1× schedule						Mask R-CNN 3×+MS schedule					
		AP^b	AP^b_{50}	AP^b_{75}	AP^m	AP^m_{50}	AP^m_{75}	AP^b	AP^b_{50}	AP^b_{75}	AP^m	AP^m_{50}	AP^m_{75}
PVT-Tiny (Wang et al., 2021)	32.9	36.7	59.2	39.3	35.1	56.7	37.3	39.8	62.2	43.0	37.4	59.3	39.9
PVTv2-B1 (Wang et al., 2022a)	33.7	41.8	64.3	45.9	38.8	61.2	41.6	44.9	67.3	49.4	40.8	64.0	43.8
ViT-T (Li et al., 2021b)	26.1	35.5	58.1	37.8	33.5	54.9	35.1	40.2	62.9	43.5	37.0	59.6	39.0
ViTDet-T (Li et al., 2022b)	26.6	35.7	57.7	38.4	33.5	54.7	35.2	40.4	63.3	43.9	37.1	60.1	39.3
ViT-Adapter-T (ours)	28.1	41.1	62.5	44.3	37.5	59.7	39.9	46.0	67.6	50.4	41.0	64.4	44.1
PVT-Small (Wang et al., 2021)	44.1	40.4	62.9	43.8	37.8	60.1	40.3	43.0	65.3	46.9	39.9	62.5	42.8
PVTv2-B2 (Wang et al., 2022a)	45.0	45.3	67.1	49.6	41.2	64.2	44.4	47.8	69.7	52.6	43.1	66.8	46.7
Swin-T (Liu et al., 2021b)	47.8	42.7	65.2	46.8	39.3	62.2	42.2	46.0	68.1	50.3	41.6	65.1	44.9
ConvNeXt-T (Liu et al., 2022)	48.1	44.2	66.6	48.3	40.1	63.3	42.8	46.2	67.9	50.8	41.7	65.0	44.9
Focal-T (Yang et al., 2021)	48.8	44.8	67.7	49.2	41.0	64.7	44.2	47.2	69.4	51.9	42.7	66.5	45.9
ViT-S (Li et al., 2021b)	43.8	40.2	63.1	43.4	37.1	59.9	39.3	44.0	66.9	47.8	39.9	63.4	42.2
ViTDet-S (Li et al., 2022b)	45.7	40.6	63.3	43.5	37.1	60.0	38.8	44.5	66.9	48.4	40.1	63.6	42.5
ViT-Adapter-S (ours)	47.8	44.7	65.8	48.3	39.9	62.5	42.8	48.2	69.7	52.5	42.8	66.4	45.9
PVTv2-B5 (Wang et al., 2022a)	101.6	47.4	68.6	51.9	42.5	65.7	46.0	48.4	69.2	52.9	42.9	66.6	46.2
Swin-B (Liu et al., 2021b)	107.1	46.9	-	-	42.3	-	-	48.6	70.0	53.4	43.3	67.1	46.7
ViT-B (Li et al., 2021b)	113.6	42.9	65.7	46.8	39.4	62.6	42.0	45.8	68.2	50.1	41.3	65.1	44.4
ViTDet-B (Li et al., 2022b)	121.3	43.2	65.8	46.9	39.2	62.7	41.4	46.3	68.6	50.5	41.6	65.3	44.5
ViT-Adapter-B (ours)	120.2	47.0	68.2	51.4	41.8	65.1	44.9	49.6	70.6	54.0	43.6	67.7	46.9
ViT-L† (Li et al., 2021b)	337.3	45.7	68.9	49.4	41.5	65.6	44.6	48.3	70.4	52.9	43.4	67.9	46.6
ViTDet-L† (Li et al., 2022b)	350.1	46.2	69.2	50.3	41.4	65.8	44.1	49.1	71.5	53.8	44.0	68.5	47.6
ViT-Adapter-L† (ours)	347.9	48.7	70.1	53.2	43.3	67.0	46.9	52.1	73.8	56.5	46.0	70.5	49.7

Table 1: **Object detection and instance segmentation with Mask R-CNN on COCO val2017.** For fair comparison, we initialize all ViT-T/S/B models with the regular ImageNet-1K pre-training (Touvron et al., 2021), and ViT-L† with the ImageNet-22K weights from (Steiner et al., 2021).

Method	AP^b	AP^b_{50}	AP^b_{75}	#P
Cascade Mask R-CNN 3×+MS schedule				
Swin-T (Liu et al., 2021b)	50.5	69.3	54.9	86M
Shuffle-T (Huang et al., 2021b)	50.8	69.6	55.1	86M
PVTv2-B2 (Wang et al., 2022a)	51.1	69.8	55.3	83M
Focal-T (Yang et al., 2021)	51.5	70.6	55.9	87M
ViT-S (Li et al., 2021b)	47.9	67.1	51.7	82M
ViT-Adapter-S (ours)	51.5	70.1	55.8	86M
Swin-B (Liu et al., 2021b)	51.9	70.9	57.0	145M
Shuffle-B (Huang et al., 2021b)	52.2	71.3	57.0	145M
ViT-B (Li et al., 2021b)	50.1	69.3	54.3	151M
ViT-Adapter-B (ours)	52.1	70.6	56.5	158M

Method	AP^b	AP^b_{50}	AP^b_{75}	#P
ATSS 3×+MS schedule				
Swin-T (Liu et al., 2021b)	47.2	66.5	51.3	36M
Focal-T (Yang et al., 2021)	49.5	68.8	53.9	37M
PVTv2-B2 (Wang et al., 2022a)	49.9	69.1	54.1	33M
ViT-S (Li et al., 2021b)	45.2	64.8	49.0	32M
ViT-Adapter-S (ours)	49.6	68.5	54.0	36M
GFL 3×+MS schedule				
Swin-T (Liu et al., 2021b)	47.6	66.8	51.7	36M
PVTv2-B2 (Wang et al., 2022a)	50.2	69.4	54.7	33M
ViT-S (Li et al., 2021b)	46.0	65.5	49.7	32M
ViT-Adapter-S (ours)	50.0	69.1	54.3	36M

Table 2: **Object detection with different frameworks on COCO val2017.** For fair comparison, we initialize all ViT-S/B models with the regular ImageNet-1K pre-training (Touvron et al., 2021). "#P" denotes the number of parameters. "MS" means multi-scale training.

Mask R-CNN (He et al., 2017), Cascade Mask R-CNN (Cai & Vasconcelos, 2019), ATSS (Zhang et al., 2020), and GFL (Li et al., 2020). To save time and memory, we refer to (Li et al., 2021b) and modify the L-layer ViT to use 14×14 window attention except for layers spaced at an interval of $L/4$. Following common practices (Wang et al., 2021), we adopt 1× or 3× training schedule (*i.e.*, 12 or 36 epochs) with a batch size of 16, and AdamW (Loshchilov & Hutter, 2017) optimizer with an initial learning rate of 1×10^{-4} and a weight decay of 0.05.

Results with ImageNet-1K Pre-training. In Table 1 and Table 2, we apply the DeiT (Touvron et al., 2021) released ImageNet-1K weights (without distillation) as the initialization for all ViT-T/S/B models. We compare our ViT-Adapter with two related approaches (Li et al., 2021b; 2022b) and multiple representative vision-specific backbones (Wang et al., 2021; 2022a; Huang et al., 2021b; Liu et al., 2021b; Yang et al., 2021). As we can see, when using regular training settings for fair comparison, the detection performance of ViT (Li et al., 2021b) and ViTDet (Li et al., 2022b) is inferior to recent vision-specific models. For example, with Mask R-CNN and 3×+MS schedule,

Method	Pre-train	Crop Size	Semantic FPN 80k			UperNet 160k		
			#Param	mIoU	+MS	#Param	mIoU	+MS
PVT-Tiny (Wang et al., 2021)	IN-1K	512×512	17.0M	36.6	37.3	43.2M	38.5	39.0
ViT-T (Li et al., 2021b)	IN-1K	512×512	10.2M	39.4	40.5	34.1M	41.7	42.6
ViT-Adapter-T (ours)	IN-1K	512×512	12.2M	41.7	42.1	36.1M	42.6	43.6
PVT-Small (Wang et al., 2021)	IN-1K	512×512	28.2M	41.9	42.3	54.5M	43.7	44.0
PVTv2-B2 (Wang et al., 2022a)	IN-1K	512×512	29.1M	45.2	45.7	-	-	-
Swin-T (Liu et al., 2021b)	IN-1K	512×512	31.9M	41.5	-	59.9M	44.5	45.8
Twins-SVT-S (Chu et al., 2021a)	IN-1K	512×512	28.3M	43.2	-	54.4M	46.2	47.1
ViT-S (Li et al., 2021b)	IN-1K	512×512	27.8M	44.6	45.8	53.6M	44.6	45.7
ViT-Adapter-S (ours)	IN-1K	512×512	31.9M	46.1	46.6	57.6M	46.2	47.1
Swin-B (Liu et al., 2021b)	IN-1K	512×512	91.2M	46.0	-	121.0M	48.1	49.7
Twins-SVT-L (Chu et al., 2021a)	IN-1K	512×512	103.7M	46.7	-	133.0M	48.8	50.2
ViT-B (Li et al., 2021b)	IN-1K	512×512	98.0M	46.4	47.6	127.3M	46.1	47.1
ViT-Adapter-B (ours)	IN-1K	512×512	104.6M	47.9	48.9	133.9M	48.8	49.7
Swin-B† (Liu et al., 2021b)	IN-22K	640×640	-	-	-	121.0M	50.0	51.7
Swin-L† (Liu et al., 2021b)	IN-22K	640×640	-	-	-	234.0M	52.1	53.5
ViT-Adapter-B† (ours)	IN-22K	512×512	104.6M	50.7	51.9	133.9M	51.9	52.5
ViT-Adapter-L† (ours)	IN-22K	512×512	332.0M	52.9	53.7	363.8M	53.4	54.4
ViT-Adapter-L* (ours)	MM	512×512	332.0M	54.2	54.7	363.8M	55.0	55.4

Table 4: **Semantic segmentation on the ADE20K val set.** Semantic FPN (Kirillov et al., 2019) and UperNet (Xiao et al., 2018) are used as segmentation frameworks. "IN-1K/22K" and "MM" represent ImageNet-1K/22K and multi-modal pre-training. "MS" denotes multi-scale testing.

ViT-S and ViTDet-S are 3.8 AP^b and 3.3 AP^b lower than PVTv2-B2 (Wang et al., 2022a) respectively. Differently, our ViT-Adapter-S outperforms these two approaches by clear margins and even 0.4 AP^b higher than PVTv2-B2. This observation can also be seen in the experiments of three other detectors, including Cascade Mask R-CNN, ATSS, and GFL. These results indicate that, *with only the regular ImageNet-1K pre-training*, ViT-Adapter can promote the plain ViT to attain similar or even superior performance than these vision-specific transformers.

Results with ImageNet-22K Pre-training. In Table 1, we employ the ImageNet-22K pre-trained weights from AugReg (Steiner et al., 2021) to initialize all ViT-L models, including ViT (Li et al., 2021b), ViTDet (Li et al., 2022b), and our ViT-Adapter. It can be seen that, when training Mask R-CNN with 3×+MS schedule, our ViT-Adapter-L† brings 3.8 AP^b and 3.0 AP^b improvements over ViT-L† (Li et al., 2021b) and ViTDet-L† (Li et al., 2022b), respectively.

Results with Multi-Modal Pre-training. In this experiment, we study the effect of multi-modal pre-training. Specifically, we fine-tune the ViT-Adapter-B with Mask R-CNN for the 3×+MS schedule using different pre-trained weights. As shown in Table 3, simply replacing the ImageNet-22K pre-training (Steiner et al., 2021) with the multi-modal pre-training (Zhu et al., 2021) gives us a significant gain of 0.7 AP^b and AP^m. These results indicate that our method can easily derive considerable benefits from advanced multi-modal pre-training, which is difficult for vision-specific models like Swin.

Method	Pre-train	AP^b	AP^m
Swin-B (Mask R-CNN 3×+MS)	ImageNet-1K	48.6	43.3
	ImageNet-22K	49.6	44.3
	Multi-Modal	N/A	N/A
ViT-Adapter-B (Mask R-CNN 3×+MS)	ImageNet-1K	49.6	43.6
	ImageNet-22K	50.5	44.6
	Multi-Modal	**51.2**	**45.3**

Table 3: **Comparison of different pre-trained weights.** Our method retains the flexibility of ViT and thus could benefit from advanced multi-modal pre-training (Zhu et al., 2021).

4.2 Semantic Segmentation

Settings. We evaluate our ViT-Adapter on semantic segmentation with the ADE20K (Zhou et al., 2017) dataset and MMSegmentation (Contributors, 2020) codebase. Both Semantic FPN (Kirillov et al., 2019) and UperNet (Xiao et al., 2018) are employed as the basic frameworks. For Semantic FPN, we apply the settings of PVT (Wang et al., 2021) and train the models for 80k iterations. For UperNet, we follow the settings of Swin (Liu et al., 2021b) to train it for 160k iterations.

Results with ImageNet-1K Pre-training. In Table 4, we report the semantic segmentation results in terms of single-scale and multi-scale (MS) mIoU. As same as Section 4.1, we initialize all ViT-T/S/B models with the DeiT (Touvron et al., 2021) released ImageNet-1K weights. It shows that, under comparable model sizes, our method surpasses the ViT (Li et al., 2021b) and many representative vision-specific transformers (Wang et al., 2021; 2022a; Liu et al., 2021b; Chu et al., 2021a). For instance, our ViT-Adapter-S achieves 47.1 MS mIoU with UperNet, outperforming many strong counterparts such as Swin-T. Similarly, ViT-Adapter-B reports a competitive performance of 49.7 MS mIoU, which is 2.6 points higher than ViT-B and on-par with Swin-B and Twins-SVT-L. These fair comparisons using only regular ImageNet-1K pre-training (Touvron et al., 2021) demonstrate the effectiveness and universality of our ViT-Adapter.

Results with ImageNet-22K Pre-training. When using the ImageNet-22K pre-trained weights (Steiner et al., 2021), our ViT-Adapter-B† attains 51.9 mIoU and 52.5 MS mIoU with UperNet, exceeding Swin-B† by at least 0.8 mIoU. Similarly, ViT-Adapter-L† yields the results of 53.4 mIoU and 54.4 MS mIoU, which is outstanding from the counterparts like Swin-L†. These significant and consistent improvements over different model sizes suggest that our method can cover the shortage of plain ViT, making it more suitable for semantic segmentation.

Results with Multi-Modal Pre-training. Here, we apply the multi-modal pre-trained weights from Uni-Perceiver (Zhu et al., 2021) for semantic segmentation. As shown in Table 4, for Semantic FPN and UperNet, replacing the ImageNet-22K pre-training with multi-modal pre-training benefits our ViT-Adapter-L* with impressive gains of 1.3 mIoU and 1.6 mIoU, respectively.

4.3 Comparisons with State-of-the-Arts

Settings. We conduct experiments to combine our ViT-Adapter with state-of-the-art detection/segmentation frameworks, including HTC++ (Liu et al., 2021b) (without extra detection dataset) and Mask2Former (Cheng et al., 2021), and recent multi-modal pre-training BEiTv2 (Peng et al., 2022). The experimental settings are listed in Appendix A.1.1 and A.1.2.

Results. As shown in Table 5, our method reaches state-of-the-art performance. While these results may be partly due to the effectiveness of advanced pre-training, our study demonstrates that plain backbone detectors/segmenters can challenge the entrenched position of hierarchical backbones.

COCO test-dev	AP^b	AP^m
CB-Swin-L	60.1	52.3
SwinV2-L	60.8	52.7
ViT-Adapter-L	**60.9**	**53.0**

ADE20K val	mIoU
FD-SwinV2-G	61.4
ViT-Adapter-L	**61.5**
BEiT3(w/ ViT-Adapter)	**62.8**

Table 5: **Comparison with previous SOTA.**

4.4 Ablation Study

ViT *vs.* ViT-Adapter Feature. Recent works (Park & Kim, 2022; Si et al., 2022) show that ViT presents the characteristics of learning low-frequency global signals, while CNN tends to extract high-frequency information (*e.g.*, local edges and textures). To show the difference between the features of ViT and ViT-Adapter, we first use Fourier analysis as a toolkit for visualization. As shown in Figure 5 (a), the Fourier spectrum and relative log amplitudes of the Fourier transformed feature maps (average over 100 images) indicate that ViT-Adapter captures more high-frequency signals than the ViT (Li et al., 2021b) baseline. In addition, we also visualize the stride-8 feature map in Figure 5 (b)(c), which shows that the features of ViT are blurry and coarse. In contrast, our features are more fine-grained and have more local edges and textures. This observation demonstrates that our method grafts the merit of CNN for capturing high-frequency information to ViT.

Ablation for Components. To investigate the contribution of each key design, we gradually extend the ViT-S baseline (Li et al., 2021b) to our ViT-Adapter-S. All models are trained with Mask R-CNN for 1$\times$ schedule. As shown in the left side of Table 6, by directly resizing and adding the spatial features from SPM, our variant 1 improves 1.4 AP^b and 0.9 AP^m over the baseline, showing that local spatial information is essential for dense prediction. From variant 2, we find that the spatial feature injector further boosts the performance by 1.0 AP^b and 0.8 AP^m. This observation illustrates that cross-attention is a more flexible way to inject spatial features. Moreover, we employ the multi-scale feature extractor to reconstruct hierarchical features, which brings 2.1 AP^b and 1.1 AP^m gains, alleviating ViT's drawback of single-scale features. In summary, our proposed components are each necessary and collectively create 4.5 AP^b and 2.8 AP^m improvements.

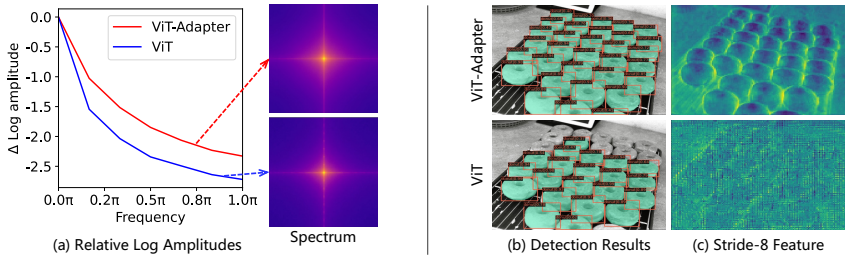


Figure 5: **ViT *vs.* ViT-Adapter Feature.** (a) Relative log amplitudes of Fourier transformed feature maps. (b) Detection results. (c) Stride-8 feature map. Compared to the ViT (Li et al., 2021b), our ViT-Adapter captures more high-frequency signals, and produces more fine-grained features with rich edges and textures, which is of great help for dense prediction.

Method	Components			Interaction Mode	Mask R-CNN 1×		
	SPM	Injector	Extractor		AP^b	AP^m	#Param
ViT-S (Li et al., 2021b)				-	40.2	37.1	43.8M
Variant 1	✓			Add	41.6	38.0	45.1M
Variant 2	✓	✓		Attention	42.6	38.8	46.6M
ViT-Adapter-S (ours)	✓	✓	✓	Attention	**44.7**	**39.9**	47.8M

N	AP^b	AP^m	#Param
0	40.2	37.1	43.8M
1	43.2	38.9	45.5M
2	43.9	39.4	46.2M
4	**44.7**	**39.9**	47.8M
6	44.7	39.8	49.4M

Table 6: **Ablation studies of ViT-Adapter.** (Left) ablation of key components. Our proposed components collectively bring 4.5 AP^b and 2.8 AP^m gains. (Right) ablation of the number of interactions N. The model gives the best performance when $N = 4$. SPM is short for the spatial prior module.

Attention Mechanism	Complexity	AP^b	AP^m	FLOPs	#Param	Train Time	Memory
Global Attention (Vaswani et al., 2017)	Quadratic	43.7	39.3	1080G	50.3M	1.61s	*19.0G
CSwin Attention (Dong et al., 2021)	Linear	43.5	39.2	456G	50.3M	0.56s	15.6G
Pale Attention (Wu et al., 2022a)	Linear	44.2	39.8	458G	50.3M	0.75s	17.4G
Deformable Attention (Zhu et al., 2020)	Linear	**44.7**	**39.9**	**403G**	**47.8M**	**0.36s**	**13.7G**

Table 7: **Ablation of using different attention mechanisms in our adapter.** The per-iteration training time and GPU training memory are measured by A100 GPUs with per-GPU batch size 2 and FP16 training. "*" indicates using activation checkpointing to save training memory.

Number of Interactions. In the right side of Table 6, we study the effect of number of interactions N. Specifically, we build several ViT-Adapter-S variants with different numbers of interactions. We observe that the model accuracy saturates when N goes larger, and applying more interactions cannot monotonically promote the performance. Therefore, we empirically set N to 4 by default.

Attention Type. Our method is a general framework in which the attention mechanism is optional. To verify this, we adopt ViT-Adapter-S as the basic model and study 4 different attention mechanisms. As shown in Table 7, sparse attention with linear complexity is more suitable for our adapter than global attention with quadratic complexity. We ended up using deformable attention (Zhu et al., 2020) as the default configuration. Notably, it can be replaced by other more advanced attention mechanisms in the future to further boost the performance.

5 Conclusion

This work explores a new paradigm, namely ViT-Adapter, to bridge the performance gap between the plain ViT and vision-specific transformers on dense prediction tasks. Without modifying the inherent architecture, we flexibly inject image-related inductive biases into the ViT and reconstruct fine-grained multi-scale features required by dense predictions. Extensive experiments on object detection, instance segmentation, and semantic segmentation show that our method can achieve comparable or even better performance than well-designed vision-specific transformers, and further derive considerable benefits from advanced multi-modal pre-training.

REFERENCES

Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. Layer normalization. *arXiv preprint arXiv:1607.06450*, 2016. 5

Hyojin Bahng, Ali Jahanian, Swami Sankaranarayanan, and Phillip Isola. Exploring visual prompts for adapting large-scale models. *arXiv preprint arXiv:2203.17274*, 1(3):4, 2022. 18, 19

Hangbo Bao, Li Dong, and Furu Wei. Beit: Bert pre-training of image transformers. In *ICLR*, 2022. 3, 15, 16

Navaneeth Bodla, Bharat Singh, Rama Chellappa, and Larry S Davis. Soft-nms–improving object detection with one line of code. In *ICCV*, pp. 5561–5569, 2017. 15

Daniel Bolya, Sean Foley, James Hays, and Judy Hoffman. Tide: A general toolbox for identifying object detection errors. In *ECCV*, pp. 558–573, 2020. 17

Holger Caesar, Jasper Uijlings, and Vittorio Ferrari. Coco-stuff: Thing and stuff classes in context. In *CVPR*, pp. 1209–1218, 2018. 17

Zhaowei Cai and Nuno Vasconcelos. Cascade r-cnn: high quality object detection and instance segmentation. *TPAMI*, 43(5):1483–1498, 2019. 6, 15

Kai Chen, Jiangmiao Pang, Jiaqi Wang, Yu Xiong, Xiaoxiao Li, Shuyang Sun, Wansen Feng, Ziwei Liu, Jianping Shi, Wanli Ouyang, et al. Hybrid task cascade for instance segmentation. In *CVPR*, pp. 4974–4983, 2019a. 15

Kai Chen, Jiaqi Wang, Jiangmiao Pang, Yuhang Cao, Yu Xiong, Xiaoxiao Li, Shuyang Sun, Wansen Feng, Ziwei Liu, Jiarui Xu, Zheng Zhang, Dazhi Cheng, Chenchen Zhu, Tianheng Cheng, Qijie Zhao, Buyu Li, Xin Lu, Rui Zhu, Yue Wu, Jifeng Dai, Jingdong Wang, Jianping Shi, Wanli Ouyang, Chen Change Loy, and Dahua Lin. MMDetection: Open mmlab detection toolbox and benchmark. *arXiv preprint arXiv:1906.07155*, 2019b. 5

Shoufa Chen, Chongjian Ge, Zhan Tong, Jiangliu Wang, Yibing Song, Jue Wang, and Ping Luo. Adaptformer: Adapting vision transformers for scalable visual recognition. *arXiv preprint arXiv:2205.13535*, 2022. 18, 19

Wuyang Chen, Xianzhi Du, Fan Yang, Lucas Beyer, Xiaohua Zhai, Tsung-Yi Lin, Huizhong Chen, Jing Li, Xiaodan Song, Zhangyang Wang, et al. A simple single-scale vision transformer for object localization and instance segmentation. *arXiv preprint arXiv:2112.09747*, 2021. 1

Bowen Cheng, Ishan Misra, Alexander G Schwing, Alexander Kirillov, and Rohit Girdhar. Masked-attention mask transformer for universal image segmentation. *arXiv preprint arXiv:2112.01527*, 2021. 8, 16

Xiangxiang Chu, Zhi Tian, Yuqing Wang, Bo Zhang, Haibing Ren, Xiaolin Wei, Huaxia Xia, and Chunhua Shen. Twins: Revisiting the design of spatial attention in vision transformers. *NeurIPS*, 34, 2021a. 5, 7, 8

Xiangxiang Chu, Zhi Tian, Bo Zhang, Xinlong Wang, Xiaolin Wei, Huaxia Xia, and Chunhua Shen. Conditional positional encodings for vision transformers. *arXiv preprint arXiv:2102.10882*, 2021b. 3

MMSegmentation Contributors. MMSegmentation: Openmmlab semantic segmentation toolbox and benchmark. https://github.com/open-mmlab/mmsegmentation, 2020. 7

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *CVPR*, pp. 248–255, 2009. 2

Xiaoyi Dong, Jianmin Bao, Dongdong Chen, Weiming Zhang, Nenghai Yu, Lu Yuan, Dong Chen, and Baining Guo. Cswin transformer: A general vision transformer backbone with cross-shaped windows. *arXiv preprint arXiv:2107.00652*, 2021. 1, 3, 5, 9

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. In *ICLR*, 2020. 1, 2, 3, 4

Hao-Shu Fang, Jianhua Sun, Runzhong Wang, Minghao Gou, Yong-Lu Li, and Cewu Lu. Instaboost: Boosting instance segmentation via probability map guided copy-pasting. In *ICCV*, pp. 682–691, 2019. 15

Yuxin Fang, Wen Wang, Binhui Xie, Quan Sun, Ledell Wu, Xinggang Wang, Tiejun Huang, Xinlong Wang, and Yue Cao. Eva: Exploring the limits of masked visual representation learning at scale. *arXiv preprint arXiv:2211.07636*, 2022a. 18

Yuxin Fang, Shusheng Yang, Shijie Wang, Yixiao Ge, Ying Shan, and Xinggang Wang. Unleashing vanilla vision transformer with masked image modeling for object detection. *arXiv preprint arXiv:2204.02964*, 2022b. 4

Peng Gao, Shijie Geng, Renrui Zhang, Teli Ma, Rongyao Fang, Yongfeng Zhang, Hongsheng Li, and Yu Qiao. Clip-adapter: Better vision-language models with feature adapters. *arXiv preprint arXiv:2110.04544*, 2021. 3

Golnaz Ghiasi, Yin Cui, Aravind Srinivas, Rui Qian, Tsung-Yi Lin, Ekin D Cubuk, Quoc V Le, and Barret Zoph. Simple copy-paste is a strong data augmentation method for instance segmentation. In *CVPR*, pp. 2918–2928, 2021. 19

Kai Han, An Xiao, Enhua Wu, Jianyuan Guo, Chunjing Xu, and Yunhe Wang. Transformer in transformer. *NeurIPS*, 34, 2021. 1

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, pp. 770–778, 2016. 4

Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *ICCV*, pp. 2961–2969, 2017. 6, 19

Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. *arXiv preprint arXiv:2111.06377*, 2021. 3, 15, 19

Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin De Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. Parameter-efficient transfer learning for nlp. In *ICML*, pp. 2790–2799, 2019. 2, 3

Shihua Huang, Zhichao Lu, Ran Cheng, and Cheng He. Fapn: Feature-aligned pyramid network for dense image prediction. In *ICCV*, pp. 864–873, 2021a. 16

Zilong Huang, Youcheng Ben, Guozhong Luo, Pei Cheng, Gang Yu, and Bin Fu. Shuffle transformer: Rethinking spatial shuffle for vision transformer. *arXiv preprint arXiv:2106.03650*, 2021b. 2, 6

Jitesh Jain, Anukriti Singh, Nikita Orlov, Zilong Huang, Jiachen Li, Steven Walton, and Humphrey Shi. Semask: Semantically masked transformers for semantic segmentation. *arXiv preprint arXiv:2112.12782*, 2021. 16

Menglin Jia, Luming Tang, Bor-Chun Chen, Claire Cardie, Serge Belongie, Bharath Hariharan, and Ser-Nam Lim. Visual prompt tuning. *arXiv preprint arXiv:2203.12119*, 2022. 18, 19

Shibo Jie and Zhi-Hong Deng. Convolutional bypasses are better vision transformer adapters. *arXiv preprint arXiv:2207.07039*, 2022. 18, 19

Alexander Kirillov, Ross Girshick, Kaiming He, and Piotr Dollár. Panoptic feature pyramid networks. In *CVPR*, pp. 6399–6408, 2019. 7, 18

Feng Li, Hao Zhang, Shilong Liu, Lei Zhang, Lionel M Ni, Heung-Yeung Shum, et al. Mask dino: Towards a unified transformer-based framework for object detection and segmentation. *arXiv preprint arXiv:2206.02777*, 2022a. 16

Xiang Li, Wenhai Wang, Lijun Wu, Shuo Chen, Xiaolin Hu, Jun Li, Jinhui Tang, and Jian Yang. Generalized focal loss: Learning qualified and distributed bounding boxes for dense object detection. *NeurIPS*, 33:21002–21012, 2020. 6

Yanghao Li, Chao-Yuan Wu, Haoqi Fan, Karttikeya Mangalam, Bo Xiong, Jitendra Malik, and Christoph Feichtenhofer. Improved multiscale vision transformers for classification and detection. *arXiv preprint arXiv:2112.01526*, 2021a. 15

Yanghao Li, Saining Xie, Xinlei Chen, Piotr Dollar, Kaiming He, and Ross Girshick. Benchmarking detection transfer learning with vision transformers. *arXiv preprint arXiv:2111.11429*, 2021b. 3, 6, 7, 8, 9, 15, 17, 18, 20, 21

Yanghao Li, Hanzi Mao, Ross Girshick, and Kaiming He. Exploring plain vision transformer backbones for object detection. *arXiv preprint arXiv:2203.16527*, 2022b. 3, 6, 7, 16, 18, 19

Yawei Li, Kai Zhang, Jiezhang Cao, Radu Timofte, and Luc Van Gool. Localvit: Bringing locality to vision transformers. *arXiv preprint arXiv:2104.05707*, 2021c. 1

Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *ECCV*, pp. 740–755, 2014. 2, 5

Ze Liu, Han Hu, Yutong Lin, Zhuliang Yao, Zhenda Xie, Yixuan Wei, Jia Ning, Yue Cao, Zheng Zhang, Li Dong, et al. Swin transformer v2: Scaling up capacity and resolution. *arXiv preprint arXiv:2111.09883*, 2021a. 15, 16

Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *ICCV*, pp. 10012–10022, 2021b. 1, 2, 3, 5, 6, 7, 8, 15, 16

Ze Liu, Jia Ning, Yue Cao, Yixuan Wei, Zheng Zhang, Stephen Lin, and Han Hu. Video swin transformer. *arXiv preprint arXiv:2106.13230*, 2021c. 2

Zhuang Liu, Hanzi Mao, Chao-Yuan Wu, Christoph Feichtenhofer, Trevor Darrell, and Saining Xie. A convnet for the 2020s. *arXiv preprint arXiv:2201.03545*, 2022. 6

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017. 6, 15

Namuk Park and Songkuk Kim. How do vision transformers work? *arXiv preprint arXiv:2202.06709*, 2022. 4, 8

Zhiliang Peng, Wei Huang, Shanzhi Gu, Lingxi Xie, Yaowei Wang, Jianbin Jiao, and Qixiang Ye. Conformer: Local features coupling global representations for visual recognition. In *ICCV*, pp. 367–376, 2021. 3

Zhiliang Peng, Li Dong, Hangbo Bao, Qixiang Ye, and Furu Wei. Beit v2: Masked image modeling with vector-quantized visual tokenizers. *arXiv preprint arXiv:2208.06366*, 2022. 2, 8, 15, 16

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International Conference on Machine Learning*, pp. 8748–8763. PMLR, 2021. 3, 15, 16

René Ranftl, Alexey Bochkovskiy, and Vladlen Koltun. Vision transformers for dense prediction. In *ICCV*, pp. 12179–12188, 2021. 3

Yongming Rao, Wenliang Zhao, Yansong Tang, Jie Zhou, Ser-Nam Lim, and Jiwen Lu. Hornet: Efficient high-order spatial interactions with recursive gated convolutions. *arXiv preprint arXiv:2207.14284*, 2022. 16

Sylvestre-Alvise Rebuffi, Hakan Bilen, and Andrea Vedaldi. Learning multiple visual domains with residual adapters. *NeurIPS*, 30, 2017. 3

Sylvestre-Alvise Rebuffi, Hakan Bilen, and Andrea Vedaldi. Efficient parametrization of multi-domain deep neural networks. In *CVPR*, pp. 8119–8127, 2018. 3

Amir Rosenfeld and John K Tsotsos. Incremental learning through deep adaptation. *TPAMI*, 42(3): 651–663, 2018. 3

Shuai Shao, Zeming Li, Tianyuan Zhang, Chao Peng, Gang Yu, Xiangyu Zhang, Jing Li, and Jian Sun. Objects365: A large-scale, high-quality dataset for object detection. In *ICCV*, pp. 8430–8439, 2019. 2, 15, 16

Chenyang Si, Weihao Yu, Pan Zhou, Yichen Zhou, Xinchao Wang, and Shuicheng Yan. Inception transformer. *arXiv preprint arXiv:2205.12956*, 2022. 8

Andreas Steiner, Alexander Kolesnikov, Xiaohua Zhai, Ross Wightman, Jakob Uszkoreit, and Lucas Beyer. How to train your vit? data, augmentation, and regularization in vision transformers. *arXiv preprint arXiv:2106.10270*, 2021. 6, 7, 8, 15

Asa Cooper Stickland and Iain Murray. Bert and pals: Projected attention layers for efficient adaptation in multi-task learning. In *ICML*, pp. 5986–5995, 2019. 2, 3

Robin Strudel, Ricardo Garcia, Ivan Laptev, and Cordelia Schmid. Segmenter: Transformer for semantic segmentation. In *ICCV*, pp. 7262–7272, 2021. 3

Yi-Lin Sung, Jaemin Cho, and Mohit Bansal. Vl-adapter: Parameter-efficient transfer learning for vision-and-language tasks. *arXiv preprint arXiv:2112.06825*, 2021. 3

Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Hervé Jégou. Training data-efficient image transformers & distillation through attention. In *ICML*, pp. 10347–10357, 2021. 1, 6, 8, 19

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *NeurIPS*, 30, 2017. 2, 3, 9

Wenhai Wang, Enze Xie, Xiang Li, Deng-Ping Fan, Kaitao Song, Ding Liang, Tong Lu, Ping Luo, and Ling Shao. Pyramid vision transformer: A versatile backbone for dense prediction without convolutions. In *ICCV*, pp. 568–578, 2021. 1, 3, 5, 6, 7, 8

Wenhai Wang, Enze Xie, Xiang Li, Deng-Ping Fan, Kaitao Song, Ding Liang, Tong Lu, Ping Luo, and Ling Shao. Pvtv2: Improved baselines with pyramid vision transformer. *CVMJ*, pp. 1–10, 2022a. 1, 2, 3, 4, 5, 6, 7, 8

Wenhui Wang, Hangbo Bao, Li Dong, Johan Bjorck, Zhiliang Peng, Qiang Liu, Kriti Aggarwal, Owais Khan Mohammed, Saksham Singhal, Subhojit Som, et al. Image as a foreign language: Beit pretraining for all vision and vision-language tasks. *arXiv preprint arXiv:2208.10442*, 2022b. 2, 3, 15, 16, 17, 18

Yixuan Wei, Han Hu, Zhenda Xie, Zheng Zhang, Yue Cao, Jianmin Bao, Dong Chen, and Baining Guo. Contrastive learning rivals masked image modeling in fine-tuning via feature distillation. *arXiv preprint arXiv:2205.14141*, 2022. 17

Haiping Wu, Bin Xiao, Noel Codella, Mengchen Liu, Xiyang Dai, Lu Yuan, and Lei Zhang. Cvt: Introducing convolutions to vision transformers. In *ICCV*, pp. 22–31, 2021. 4

Sitong Wu, Tianyi Wu, Haoru Tan, and Guodong Guo. Pale transformer: A general vision transformer backbone with pale-shaped attention. In *AAAI*, volume 36, pp. 2731–2739, 2022a. 9

Yu-Huan Wu, Yun Liu, Xin Zhan, and Ming-Ming Cheng. P2t: Pyramid pooling transformer for scene understanding. *TPAMI*, 2022b. 1, 3

Tete Xiao, Yingcheng Liu, Bolei Zhou, Yuning Jiang, and Jian Sun. Unified perceptual parsing for scene understanding. In *ECCV*, pp. 418–434, 2018. 7

Enze Xie, Wenhai Wang, Zhiding Yu, Anima Anandkumar, Jose M Alvarez, and Ping Luo. Segformer: Simple and efficient design for semantic segmentation with transformers. *NeurIPS*, 34, 2021. 2

Jianwei Yang, Chunyuan Li, Pengchuan Zhang, Xiyang Dai, Bin Xiao, Lu Yuan, and Jianfeng Gao. Focal self-attention for local-global interactions in vision transformers. *arXiv preprint arXiv:2107.00641*, 2021. 6, 15

Renrui Zhang, Rongyao Fang, Peng Gao, Wei Zhang, Kunchang Li, Jifeng Dai, Yu Qiao, and Hongsheng Li. Tip-adapter: Training-free clip-adapter for better vision-language modeling. *arXiv preprint arXiv:2111.03930*, 2021. 3

Shifeng Zhang, Cheng Chi, Yongqiang Yao, Zhen Lei, and Stan Z Li. Bridging the gap between anchor-based and anchor-free detection via adaptive training sample selection. In *CVPR*, pp. 9759–9768, 2020. 6

Yuanhan Zhang, Kaiyang Zhou, and Ziwei Liu. Neural prompt search. *arXiv preprint arXiv:2206.04673*, 2022. 18, 19

Sixiao Zheng, Jiachen Lu, Hengshuang Zhao, Xiatian Zhu, Zekun Luo, Yabiao Wang, Yanwei Fu, Jianfeng Feng, Tao Xiang, Philip HS Torr, et al. Rethinking semantic segmentation from a sequence-to-sequence perspective with transformers. In *CVPR*, pp. 6881–6890, 2021. 3, 18, 19

Bolei Zhou, Hang Zhao, Xavier Puig, Sanja Fidler, Adela Barriuso, and Antonio Torralba. Scene parsing through ade20k dataset. In *CVPR*, pp. 633–641, 2017. 2, 7

Jinguo Zhu, Xizhou Zhu, Wenhai Wang, Xiaohua Wang, Hongsheng Li, Xiaogang Wang, and Jifeng Dai. Uni-perceiver-moe: Learning sparse generalist models with conditional moes. *arXiv preprint arXiv:2206.04674*, 2022. 2, 3, 15

Xizhou Zhu, Weijie Su, Lewei Lu, Bin Li, Xiaogang Wang, and Jifeng Dai. Deformable detr: Deformable transformers for end-to-end object detection. In *ICLR*, 2020. 5, 9

Xizhou Zhu, Jinguo Zhu, Hao Li, Xiaoshi Wu, Xiaogang Wang, Hongsheng Li, Xiaohua Wang, and Jifeng Dai. Uni-perceiver: Pre-training unified architecture for generic perception for zero-shot and few-shot tasks. *arXiv preprint arXiv:2112.01522*, 2021. 2, 3, 7, 8, 15

Method	Framework	Epoch	Backbone Pre-train	val		val (+MS)		test-dev		test-dev (+MS)	
				AP^b	AP^m	AP^b	AP^m	AP^b	AP^m	AP^b	AP^m
Swin-L	HTC++	72	IN-22K, sup	57.1	49.5	58.0	50.4	57.7	50.2	58.7	51.1
Focal-L	HTC++	36	IN-22K, sup	57.0	49.9	58.1	50.9	-	-	58.4	51.3
MViTv2-L	Cascade	50	IN-22K, sup	56.9	48.6	58.7	50.5	-	-	-	-
MViTv2-H	Cascade	50	IN-22K, sup	57.1	48.8	58.4	50.1	-	-	-	-
CBV2-Swin-L	HTC	36	IN-22K, sup	59.1	51.0	59.6	51.8	59.4	51.6	60.1	52.3
ViT-Adapter-L	HTC++	36	IN-22K, sup	56.6	49.0	57.7	49.9	57.4	50.0	58.4	50.7
Swin-L	HTC++	36	IN-1K, UM-MAE	57.4	49.8	58.7	50.9	-	-	-	-
ViTDet-L	Cascade	100	IN-1K, MAE	**59.6**	51.1	60.4	52.2	-	-	-	-
ViT-Adapter-L	HTC++	36	IN-22K, BEiT	58.4	50.8	60.2	52.2	58.9	51.3	60.4	52.5
ViT-Adapter-L	HTC++	36	MM†, BEiTv2	58.8	**51.1**	**60.5**	**52.5**	**59.5**	**51.8**	**60.9**	**53.0**

Table 8: **Comparisons with the leading results on the COCO val2017 and test-dev sets.** There are three detection frameworks used, including Cascade Mask R-CNN (Cai & Vasconcelos, 2019), HTC (Chen et al., 2019a), and its extension HTC++ (Liu et al., 2021b). All of these models are trained *without* extra detection datasets, such as Object365 (Shao et al., 2019). "IN-22K, sup" is short for ImageNet-22K supervised pre-training. "MS" indicates multi-scale testing. "†": Since the pre-trained CLIP (Radford et al., 2021) model is used in the training process of BEiTv2 (Peng et al., 2022), we regard it as a multi-modal pre-training method.

A Appendix

A.1 Comparison with Previous State-of-the-Arts

In recent years, the state-of-the-art models on dense prediction benchmarks are primarily vision-specific transformers, such as Swin (Liu et al., 2021b), Focal (Yang et al., 2021), MViTv2 (Li et al., 2021a), and SwinV2 (Liu et al., 2021a), while the plain ViT is rarely found. Nevertheless, we argue that the plain ViT still has the potential to reach the leading performance by leveraging our ViT-Adapter. To verify this, we conduct extensive additional experiments as follows.

A.1.1 Object Detection and Instance Segmentation

Settings. Following prior art (Li et al., 2021b), we modify the 24-layer ViT-L to use 14×14 window attention except for layers spaced at an interval of 6, to save training time and memory. The state-of-the-art detector HTC++ (Liu et al., 2021b) is employed for our experiments. Specifically, we rescale the shorter side of images between 400 and 1400, while the longer side is at most 1600. Instaboost (Fang et al., 2019), Soft-NMS (Bodla et al., 2017), AdamW (Loshchilov & Hutter, 2017) optimizer (batch size of 16, initial learning rate of 1×10^{-4}, and weight decay of 0.05), and 3× schedule are adopted during training. We use a layer-wise learning rate decay rate of 0.9, and drop path rate of 0.4. For a fairer comparison, here we take two initialization strategies, *i.e.* regular ImageNet-22K pre-training, and more advanced self-supervised or multi-modal pre-training.

Results with ImageNet-22K Pre-training. As shown in Table 8, with the ImageNet-22K supervised pre-training from AugReg (Steiner et al., 2021), our ViT-Adapter-L reports 58.4 AP^b and 50.7 AP^m on the COCO test-dev, which is comparable to many vision-specific transformers such as Swin-L (58.4 AP^b *vs.* 58.7 AP^b) and Focal-L (58.4 AP^b *vs.* 58.4 AP^b). This fair comparison illustrates that, our ViT-Adapter significantly narrows the performance gap between the plain ViT and well-designed vision-specific models.

Results with More Advanced Pre-training. Since our paradigm retains the flexibility of the plain ViT, it can *easily derive significant benefits from advanced pre-training techniques*, such as multi-modal pre-training (Zhu et al., 2021; 2022) or self-supervised pre-training (Bao et al., 2022; Peng et al., 2022; He et al., 2021), or a combination of the both (Wang et al., 2022b). Here, we take the readily available weights from BEiT (Bao et al., 2022) and BEiTv2 (Bao et al., 2022) as examples. Due to BEiT using learnable relative position biases instead of the absolute position embeddings, we replace the remaining global attention (see settings part) with 56×56 window attention as an approximation. For these layers, the relative position biases need to be interpolated to adapt to the new window size.

Method	Framework	Backbone Pre-train	Extra Pre-train	Crop Size	Iters	ADE20K val mIoU	+MS	#Param
Swin-L	Mask2Former	IN-22K, sup	-	640	160k	56.1	57.3	215M
Swin-L-FaPN	Mask2Former	IN-22K, sup	-	640	160k	56.4	57.7	217M
SeMask-Swin-L	Mask2Former	IN-22K, sup	-	640	160k	57.0	58.2	-
HorNet-L	Mask2Former	IN-22K, sup	-	640	160k	57.5	57.9	-
ViT-Adapter-L	Mask2Former	IN-22K, sup	-	640	160k	56.8	57.7	438M
BEiT-L	UperNet	IN-22K, BEiT	-	640	160k	56.7	57.0	441M
ViT-Adapter-L	UperNet	IN-22K, BEiT	-	640	160k	58.0	58.4	451M
BEiTv2-L	UperNet	IN-22K, BEiTv2	-	512	160k	57.5	58.0	441M
ViT-Adapter-L	UperNet	IN-22K, BEiTv2	-	512	160k	58.0	58.5	451M
ConvNeXt-XL*	Mask2Former	IN-22K, sup	COCO-Stuff, sup	896	80k	57.1	58.4	588M
Swin-L*	Mask2Former	IN-22K, sup	COCO-Stuff, sup	896	80k	57.3	58.3	434M
SwinV2-G	UperNet	IN-22K, sup	Ext-70M, sup	896	160k	59.3	59.9	3.0B
FD-SwinV2-G	UperNet	IN-22K, sup	Ext-70M, sup	896	160k	-	61.4	3.0B
Swin-L	Mask DINO	IN-22K, sup	Object365, sup	1024	160k	59.5	60.8	223M
ViT-Adapter-L	Mask2Former	IN-22K, BEiT	COCO-Stuff, sup	896	80k	59.4	60.5	571M
ViT-Adapter-L	Mask2Former	MM†, BEiTv2	COCO-Stuff, sup	896	80k	**61.2**	**61.5**	571M
BEiT-3 (w/ ViT-Adapter)	Mask2Former	MM, BEiTv3	COCO-Stuff, sup	896	80k	**62.0**	**62.8**	1.3B

Table 9: **Comparison with previous state-of-the-art results on the ADE20K validation set.** The results of BEiT-3 are collected from (Wang et al., 2022b). "*": We follow BEiT (Bao et al., 2022) to use a wider segmentation head for ConvNeXt-XL and Swin-L to match the number of parameters, and apply the same training strategy to them. "IN-22K, sup": ImageNet-22K supervised pre-training. "MM": Multi-modal pre-training. "MS": Multi-scale testing. "†": Since the pre-trained CLIP (Radford et al., 2021) model is used in the training process of BEiTv2 (Peng et al., 2022), we regard it as a multi-modal pre-training method.

As reported in Table 8, our ViT-Adapter-L (w/ BEiT) creates 60.4 AP^b and 52.5 AP^m on the COCO test-dev, and ViT-Adapter-L (w/ BEiTv2) further sets this record to 60.9 AP^b and 53.0 AP^m. Notably, although it's not a perfectly controlled comparison, our method attains similar performance taking fewer training epochs (36 *vs.* 100) than ViTDet (Li et al., 2022b). We argue that a longer training schedule such as 100 epochs may bring an added bonus, but it is expensive to afford due to limited computing resources. In summary, from a system-level perspective, our ViT-Adapter can enjoy the dividends of various advanced pre-training techniques and help plain ViT achieve leading performance on the object detection and instance segmentation tasks.

A.1.2 Semantic Segmentation

Settings. For semantic segmentation, we employ the AdamW optimizer with an initial learning rate of 2×10^{-5}, a batch size of 16, and a weight decay of 0.05. Layer-wise learning rate decay of 0.9 and drop path rate of 0.4 are used to train the models. Other training settings, such as pre-training techniques, crop size, and the number of iterations, are listed in Table 9.

Results with ImageNet-22K Pre-training. As shown in Table 9, using Mask2Former (Cheng et al., 2021) as the segmenter, our ViT-Adapter-L achieves 56.8 mIoU and 57.7 MS mIoU on the ADE20K val, which is comparable to recent vision-specific models, such as Swin-L (Liu et al., 2021b), Swin-L-FaPN (Huang et al., 2021a), SeMask-Swin-L (Jain et al., 2021), and HorNet-L (Rao et al., 2022).

Results with More Advanced Pre-training. It can be seen from Table 9, when training with UperNet for 160k iterations, our ViT-Adapter-L (w/ BEiT) yields 58.4 MS mIoU, outperforming BEiT-L by 1.4 points with only 10M additional parameters. It shows that our adapter can deliver significant benefits even for a powerful self-supervised pre-trained ViT.

Furthermore, we compare the performance of our method with vision-specific models that also use additional datasets. For example, SwinV2-G (Liu et al., 2021a) uses a privately collected ImageNet-22K-ext-70M dataset that contains 70 million images. Mask DINO (Li et al., 2022a) takes the detection pre-training on large-scale Object365 (Shao et al., 2019) dataset as the initialization for segmentation. Due to limited computing resources, we explore a *simple and affordable* transfer

Variants	Settings of ViT					Settings of Adapter				Total
	Layers	Width	FFN	Heads	#Param	N	FFN	Heads	#Param	Param
Tiny (T)	12	192	768	3	5.5M	4	48	6	2.5M	8.0M
Small (S)	12	384	1536	6	21.7M	4	96	6	5.8M	27.5M
Base (B)	12	768	3072	12	85.8M	4	192	12	14.0M	99.8M
Large (L)	24	1024	4096	16	303.3M	4	256	16	23.7M	327.0M

Table 10: **Configurations of the ViT-Adapter.** We apply our adapters on four different settings of ViT, including ViT-T, ViT-S, ViT-B, and ViT-L, covering a wide range of different model sizes.

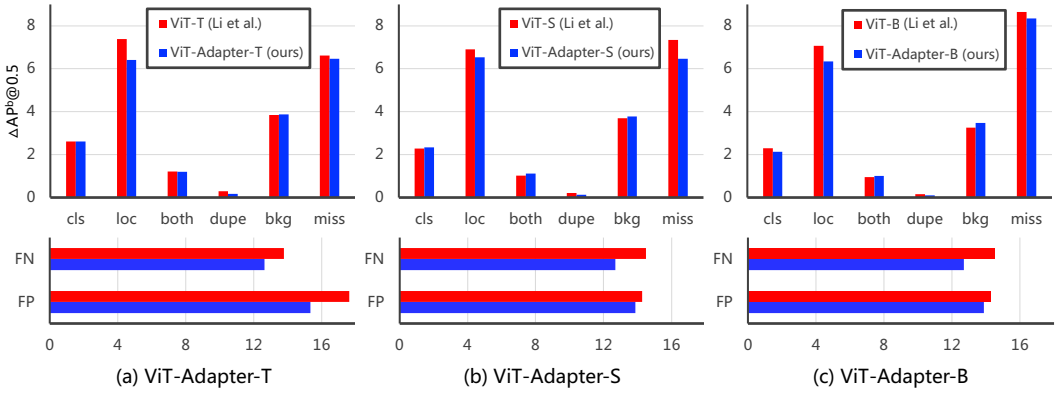


Figure 6: **TIDE error type analysis (the lower the better).** We use the models listed in Table 1 for analysis. As defined in (Bolya et al., 2020), we plot the AP^b metric at an IoU threshold of 0.5. These bars show the effect of each error type on overall detection performance). The error types include: **cls**: localized correctly but classified incorrectly; **loc**: classified correctly but localized incorrectly; **both**: classified incorrectly and localized incorrectly; **dupe**: detection would be correct if not for a higher scoring detection; **bkg**: detected background as foreground; **miss**: all undetected ground-truth not covered by other error types; **FN**: false negatives; **FP**: false positives. We observe that our ViT-Adapter makes fewer localization and miss errors than the ViT baseline (Li et al., 2021b), and occurs fewer false positive and negative errors.

learning pipeline for semantic segmentation. Specifically, we use the COCO-Stuff (Caesar et al., 2018) dataset for 80k iterations of pre-training, and then ADE20K for 80k iterations of fine-tuning. The total number of iterations is still 160k, and no additional training overhead is added.

Under this setting, our ViT-Adapter-L (w/ BEiT) produces an exciting score of 60.5 MS mIoU. Further, ViT-Adapter-L (w/ BEiTv2) creates a new record of 61.5 MS mIoU, which is slightly better than FD-SwinV2-G (Wei et al., 2022), while the parameter number is much smaller (571M *vs.* 3.0B). It's worth noting that, our ViT-Adapter is also *adopted by the recently proposed BEiTv3* (Wang et al., 2022b), which is a ViT-style foundation model that can be pre-trained with multi-modal data. As described in their paper, using ViT-Adapter for the transfer learning of semantic segmentation, BEiTv3 establishes a new state-of-the-art of 62.8 MS mIoU on ADE20K val, which is a convincing verification of our new paradigm proposed in Figure 1.

A.2 Additional Ablation and Discussion

Architecture Configurations. The more detailed configurations are listed in Table 10.

TIDE Error Type Analysis. TIDE (Bolya et al., 2020) is a toolbox for analyzing the sources of error in object detection algorithms. Following (Li et al., 2021b), we show the error type analysis in Figure 6. For fair comparison, the models listed in Table 1 are adopted for analysis. These results reveal where our ViT-Adapter improves overall AP^b relative to the ViT baseline (Li et al., 2021b). For instance, we observe that our adapter helps reduce missed and localization errors, and has a substantial effect on fixing false negative and positive errors.

Method	Segmentor	Pre-train	#Param	FLOPs	Train Time	Train Mem.	FPS	mIoU_{ss}	mIoU_{ms}
ViT-B	SETR-PUP	IN-1K	98M	170G	0.16s/iter	9.5G	30.3	46.3	47.3
ViT-B	Semantic FPN	IN-1K	98M	147G	0.15s/iter	5.6G	29.7	46.4	47.6
ViT-Adapter-B	Semantic FPN	IN-1K	105M	183G	0.16s/iter	7.5G	26.7	**47.9**	**48.9**
ViT-L	SETR-PUP	IN-22K	318M	425G	0.25s/iter	16.8G	14.0	48.6	50.1
ViT-L	Semantic FPN	IN-22K	321M	414G	0.21s/iter	14.1G	15.5	51.5	52.0
ViT-Adapter-L_{light}	Semantic FPN	IN-22K	324M	445G	0.23s/iter	15.2G	13.5	52.7	53.5
ViT-Adapter-L	Semantic FPN	IN-22K	332M	473G	0.25s/iter	16.0G	12.9	**52.9**	**53.7**

Table 11: **Efficiency analysis of ViT-Adapter for semantic segmentation.** We equip our ViT-Adapter models with Semantic FPN (Kirillov et al., 2019) for segmentation and compared them with the representative ViT-based method SETR (Zheng et al., 2021). We use a single A100 GPU and FP16 mode to benchmark these models. The training time and training memory are measured with a batch size of 2, and the FPS is tested with a batch size of 1. The FLOPs is calculated with 512×512 inputs. light: We build a lightweight model by reducing the width of the adapter.

Method	Detector	Pre-train	#Param	FLOPs	Train Time	Train Mem.	FPS	AP^{b}	AP^{m}
ViT-B	Mask R-CNN	IN-1K	114M	692G	0.25s/iter	18.3G	17.4	45.8	41.3
ViTDet-B	Mask R-CNN	IN-1K	121M	747G	0.25s/iter	18.2G	17.1	46.3	41.6
ViT-Adapter-B	Mask R-CNN	IN-1K	120M	832G	0.32s/iter	21.7G	13.5	**49.6**	**43.6**
ViT-L	Mask R-CNN	IN-22K	337M	1.7T	0.44s/iter	33.9G	9.2	48.3	43.4
ViTDet-L	Mask R-CNN	IN-22K	350M	1.9T	0.45s/iter	33.8G	9.0	49.1	44.0
ViT-Adapter-L	Mask R-CNN	IN-22K	347M	1.9T	0.52s/iter	37.6G	7.8	**52.1**	**46.0**

Table 12: **Efficiency analysis of ViT-Adapter for object detection.** We benchmark all ViT-based methods with Mask R-CNN 3× schedule. The training time/memory are measured with a batch size of 2, and the FPS is tested with a batch size of 1. The FLOPs is calculated with 1280×800 inputs.

Feature Visualization. We plot more visualization of feature maps produced by ViT-B (Li et al., 2021b) and our ViT-Adapter-B in Figure 7 and Figure 8, which are trained based on Mask R-CNN for detection and UperNet for segmentation, respectively. As can be seen, the features of ViT-B are blurry and coarse, while our features are more refined and have more local edges and textures. This observation also accords with the Fourier analysis in Section 4.4, which demonstrates that ViT has the characteristics of capturing low-frequency information, and our ViT-Adapter can supplement the missing high-frequency signals.

Efficiency Analysis. Here, we analyze the efficiency of our ViT-Adapter based on MMDetection and MMSegmentation. The comparisons with SETR (Zheng et al., 2021) and ViTDet (Li et al., 2022b) are shown in Table 11 and Table 12, respectively. We see that our parameters, training/inference speed, and computational/memory overhead are similar to mainstream ViT-based approaches. Although due to the speed bottleneck of the ViT backbone and deformable attention, the latency of ViT-based methods (including our ViT-Adapter) has no advantage over CNN-based methods, our method can still benefit the computer vision community with its leading accuracy, and it has already been adopted as one of a default framework for dense prediction in recent state-of-the-art vision foundation models (Wang et al., 2022b; Fang et al., 2022a).

Comparison with SETR. Like ViTDet (Li et al., 2022b), SETR (Zheng et al., 2021) also changes the shape of features of ViT according to the task prior (see Fig. 3 (a)), thus allowing ViT to achieve better segmentation performance. Although this paradigm shares some similarities with our approach, *e.g.* combining ViT and convolutions, they have three main differences: **(1)** In addition to the task prior, our method also takes the information of the input image (the input prior) into consideration when adapting ViT to dense prediction tasks; **(2)** The input prior will constantly interact with ViT's features, making the output features more suitable for dense prediction tasks; **(3)** Our method is an adapter that is general in both detection and segmentation tasks, and moreover achieves better results than segmentation-specific head SETR (Zheng et al., 2021).

Comparison with other Adapters. We would like to clarify the differences between ViT-Adapter and other adapters (Jia et al., 2022; Bahng et al., 2022; Chen et al., 2022; Zhang et al., 2022; Jie & Deng, 2022) for ViTs, from two aspects as follows:

(1) *Different tasks*. Our method is designed for dense prediction tasks, while VPT (Jia et al., 2022), Visual Prompt (Bahng et al., 2022), AdaptFormer (Chen et al., 2022), NOAH (Zhang et al., 2022), and Convpass (Jie & Deng, 2022) are mainly proposed for classification tasks. By training the parameters only in input spaces, or some modules attached to the backbone, or their combination, these models perform well on classification and even obtain better results than full tuning models.

However, when applying these methods (Jia et al., 2022; Bahng et al., 2022; Chen et al., 2022; Zhang et al., 2022; Jie & Deng, 2022) to dense prediction tasks, they perform below expectations. For example, we see from the Table 13 that the performance of VPT (Jia et al., 2022) has a large gap with the baseline ViT-L (Zheng et al., 2021) on ADE20K.

Method	Trainable Param.	mIoU (ss/ms)
ViT-L (full tuning)	318M	48.3 / 50.1
ViT-L (w/ VPT)	16M	44.0 / 45.6
ViT-Adapter-L (full tuning)	332M	52.9 / 53.7
ViT-Adapter-L (freeze ViT)	30M	49.0 / 50.6

Table 13: **Comparison with VPT.**

(2) *Different targets*. These mentioned adapters (Jia et al., 2022; Bahng et al., 2022; Chen et al., 2022; Zhang et al., 2022; Jie & Deng, 2022) aim to explore parameter-efficient transfer learning, while the goal of our ViT-Adapter is to push the performance boundaries of plain ViT downstream applications, make ViT more general for downstream tasks, and efficiently utilize large-scale ViT weights pre-trained in different ways. We argue that these two technical lines are orthogonal, as shown in the last column in Table 13. Combining ViT-Adapter with these adapters to achieve efficient and accurate transfer learning of dense prediction is a research topic worth exploring.

Comment of ViTDet's Performance. Due to the different training settings, the performance of ViTDet (Li et al., 2022b) reported in Table 1 is lower than the original paper. To avoid misunderstanding, we would like to make further clarification.

(1) In the ViTDet, the authors adopt stronger training strategies than us, *i.e.*, loading the more advanced MAE (He et al., 2021) pre-trained weights, and using the Large Scale Jitter (Ghiasi et al., 2021) augmentation to train the model for **100 epochs**. These settings lead to 3× training cost compared to the commonly used 3× schedule (*i.e.*, 36 epochs). To some extent, it confirms the lack of image-related inductive bias in ViT will lead to slow convergence of the plain ViT detector.

(2) For fairer comparison, we benchmark all plain ViT methods using Mask R-CNN (He et al., 2017), including ViTDet and our ViT-Adapter, under the widely used 3× training schedule, and keep the pre-trained weights as same (*i.e.*, ImageNet-1K weights from DeiT (Touvron et al., 2021)). It makes sense that our ViT-Adapter achieves better performance than ViTDet under this setting, because our adapter introduces image-related prior into the plain ViT, which is helpful for dense prediction tasks such as detection and segmentation.

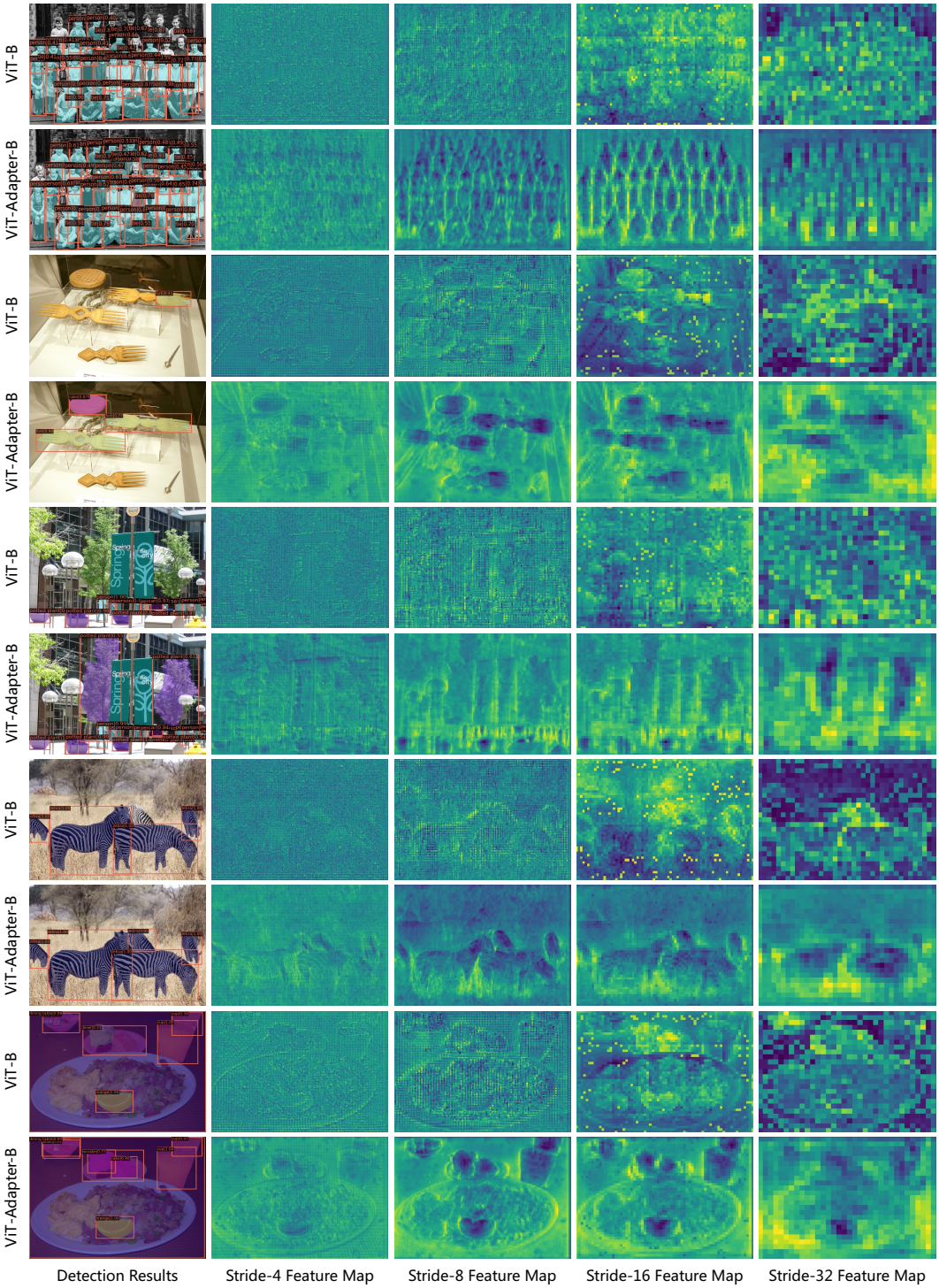


Figure 7: **Visualization of feature maps for object detection and instance segmentation.** Compared to the ViT baseline (Li et al., 2021b), our ViT-Adapter yields more fine-grained multi-scale feature maps, thus improving localization quality and reducing missed detection.

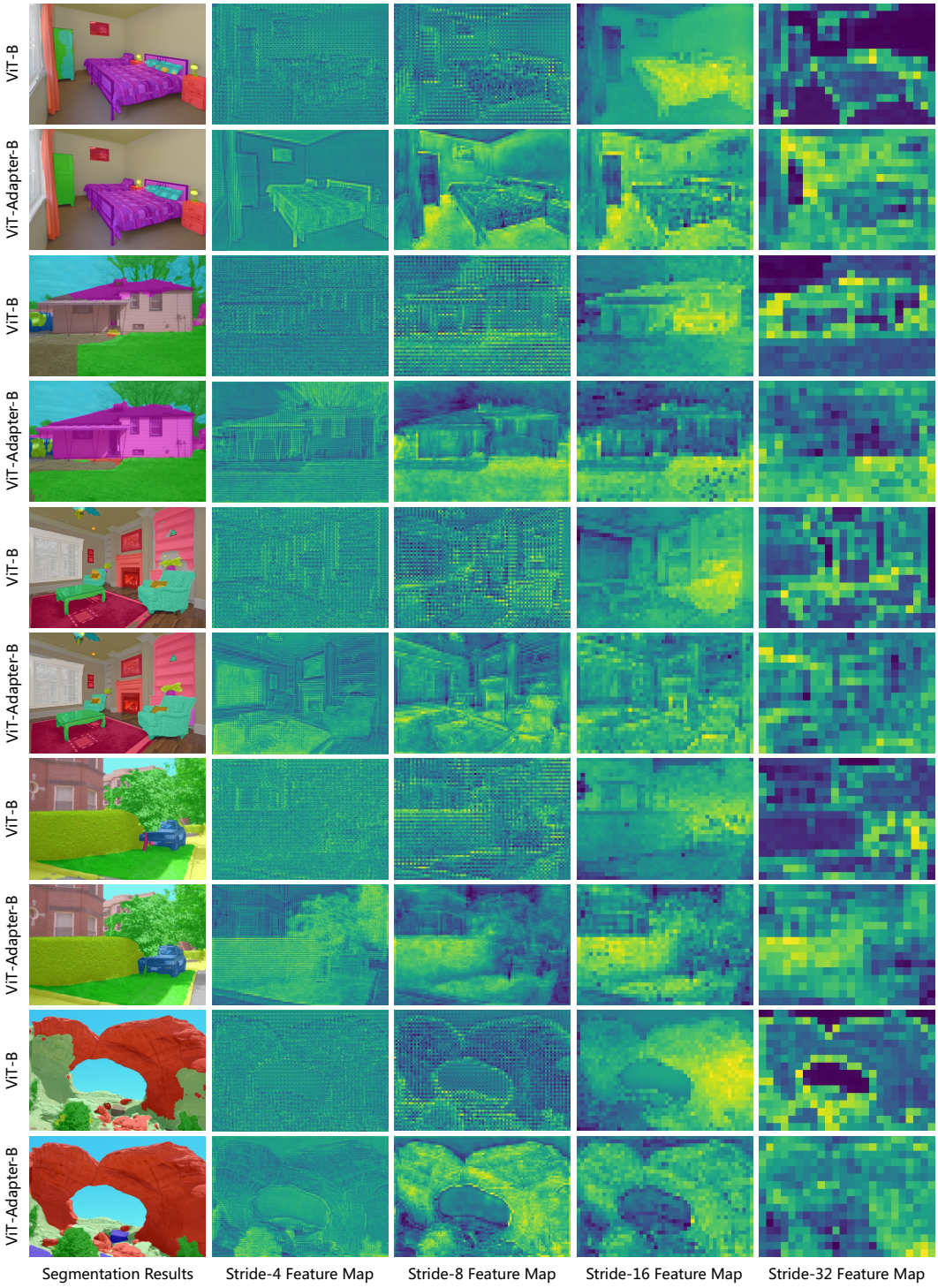


Figure 8: **Visualization of feature maps for semantic segmentation.** Compared to the ViT baseline (Li et al., 2021b), our ViT-Adapter yields more fine-grained multi-scale feature maps with rich local edges and textures, thus improving the performance of semantic segmentation.