

# **Vision Transformer Adapter for Dense Predictions**

Zhe Chen\*, Yuchen Duan\*, Wenhai Wang\*, Junjun He, Tong Lu,  
Jifeng Dai, and Yu Qiao

Nanjing University, Shanghai AI Laboratory, and Tsinghua  
University


Presented by Minho Park

<https://arxiv.org/abs/2205.08534>


# State-of-the-art

---


- Dense prediction: Semantic segmentation, Object detection, etc.

 README.md


## ViT-Adapter

 Ranked #4


 Semantic Segmentation on ADE20K (using additional training data)

 State of the Art


 Semantic Segmentation on Cityscapes test (using additional training data)

 State of the Art


 Semantic Segmentation on COCO-Stuff test

 State of the Art

 Semantic Segmentation on PASCAL Context

 Ranked #10

 Object Detection on COCO test-dev (using additional training data)

 Ranked #7

 Instance Segmentation on COCO test-dev

The official implementation of the paper "[Vision Transformer Adapter for Dense Predictions](#)".

## News

(2022/06/09) ViT-Adapter-L yields 60.4 box AP and 52.5 mask AP on COCO test-dev.

(2022/06/04) Code and models are released.

(2022/05/17) ViT-Adapter-L yields 60.1 box AP and 52.1 mask AP on COCO test-dev.

(2022/05/12) ViT-Adapter-L reaches 85.2 mIoU on Cityscapes test set without coarse data.

(2022/05/05) ViT-Adapter-L achieves the SOTA on ADE20K val set with 60.5 mIoU!

# Contributions

---

- Capture good properties of Vision Transformer (ViT) for dense prediction tasks.
  - ViT can be trained by multimodal input (image, text, etc.) since its global transformer architecture.
  - ViT has a full receptive field with only leveraging single layer.
- Reasonable architecture choice for Adapter.
- However, the adapter significant increases the computational cost.

# Various Transformer Architectures for Computer Vision

---

- Convolutional Neural Networks (e.g., ConvNeXt, HRNet)
- Global self-attention (e.g., ViT, BEiT)
- Local self-attention (e.g., Swin)
- Mixed architecture

# Recent Trend

- Pretraining (SSL) + Finetuning
- Multi-modal learning or Foundation models (leveraging large-scale dataset)
  - e.g., CLIP, NUWA, Florence, BEiT-3, CoCa, etc.

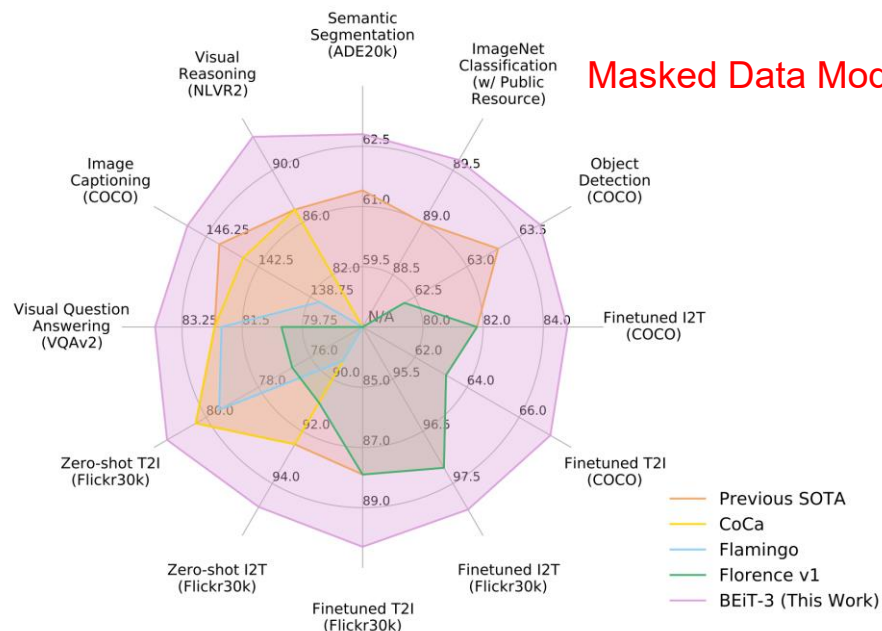
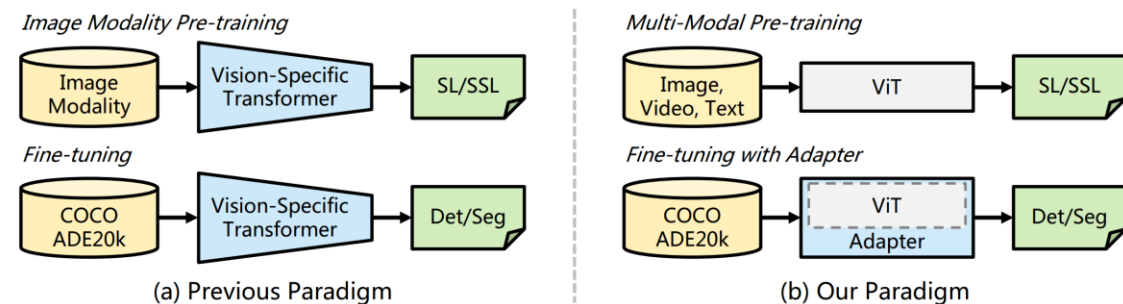
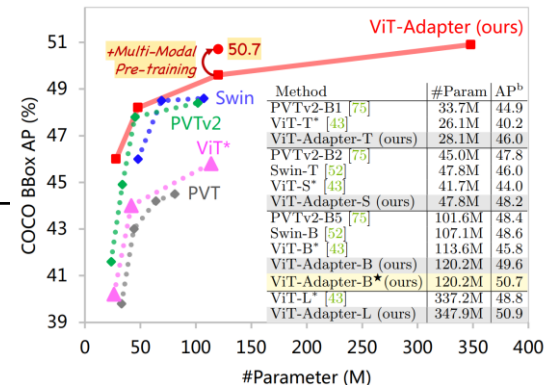


Figure 1: BEiT-3 achieves state-of-the-art performance on a broad range of tasks compared with other customized or foundation models. I2T/T2I is short for image-to-text/text-to-image retrieval.



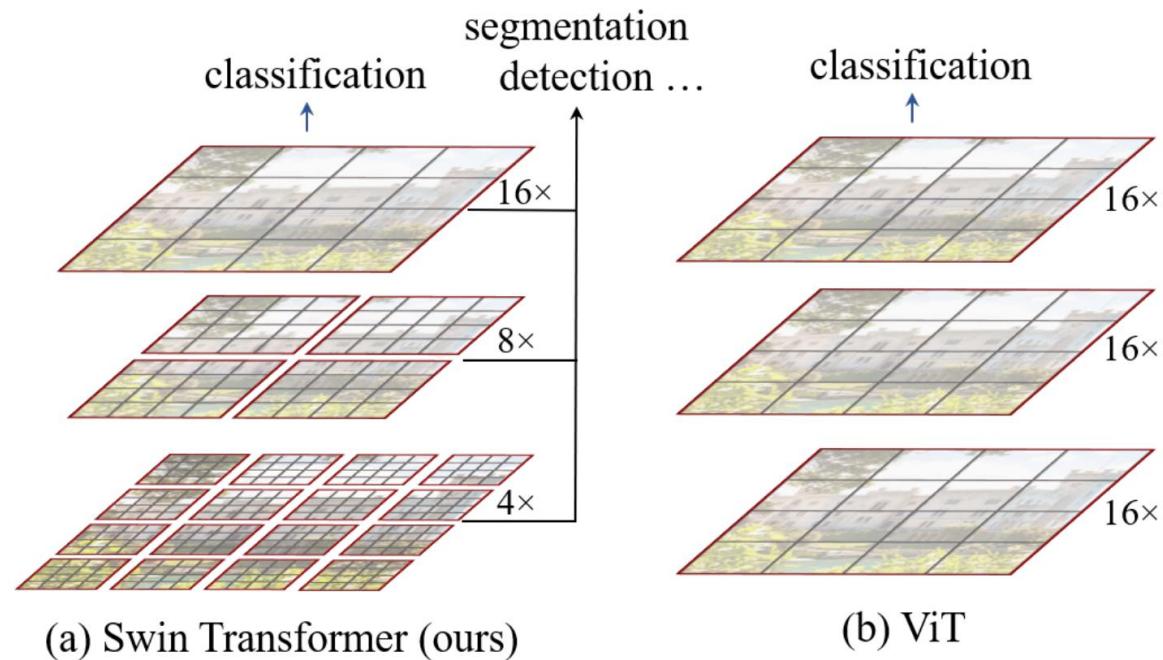
**Fig. 1. Previous paradigm vs. our paradigm.** (a) Previous paradigm designs vision-specific models and pre-trains on large-scale image datasets via supervised (SL) or self-supervised learning (SSL) and then fine-tunes them on downstream tasks. (b) We propose an adapter to close the gap between ViT [22] and vision-specific models on 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.



**Fig. 2. Object detection performance on COCO val2017 using Mask R-CNN.** We see that our method achieves significant improvements on object detection. \* indicates using multi-modal pre-trained weights from [86].

# Weakness of ViT

- Single-scale representation
- Low-resolution representation



Swin Transformer

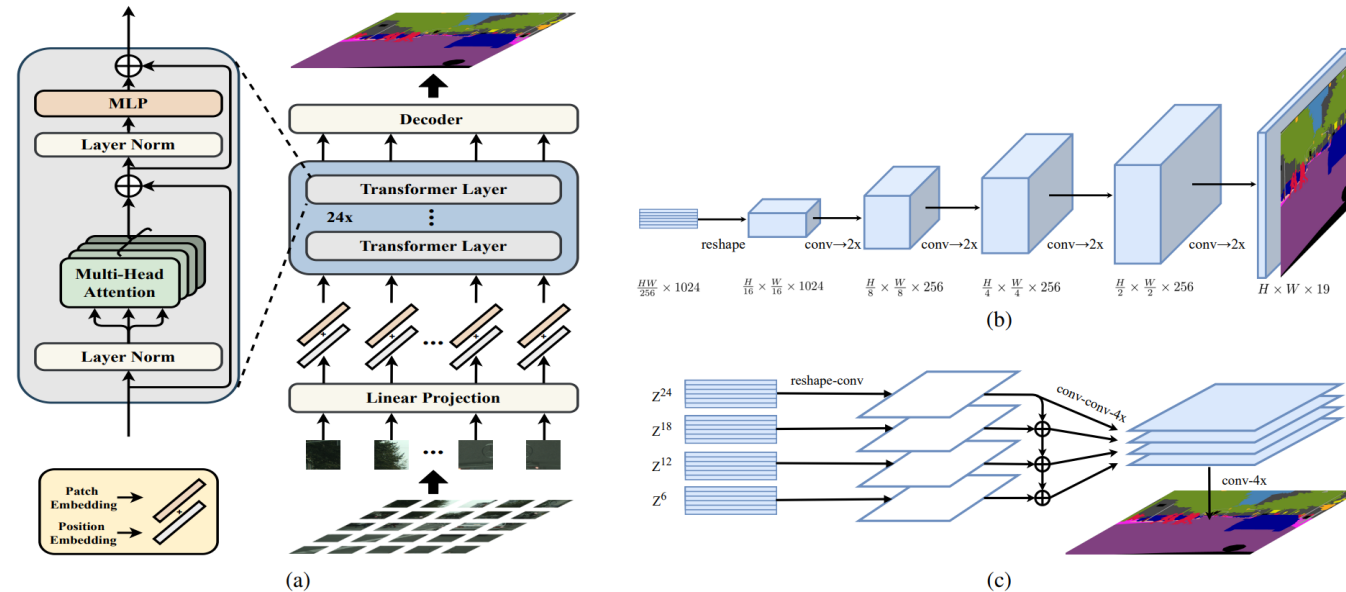
# Adapter (In NLP Field)

---

- In the NLP field this trend is already started since BERT and GPT-3.
- This work aims to develop an adapter to close the gap between vanilla transformers such as ViT and the dedicated models for downstream vision tasks.

# Previous Work

- Decoders for ViT.
  - SETR, Segmenter, and DPT



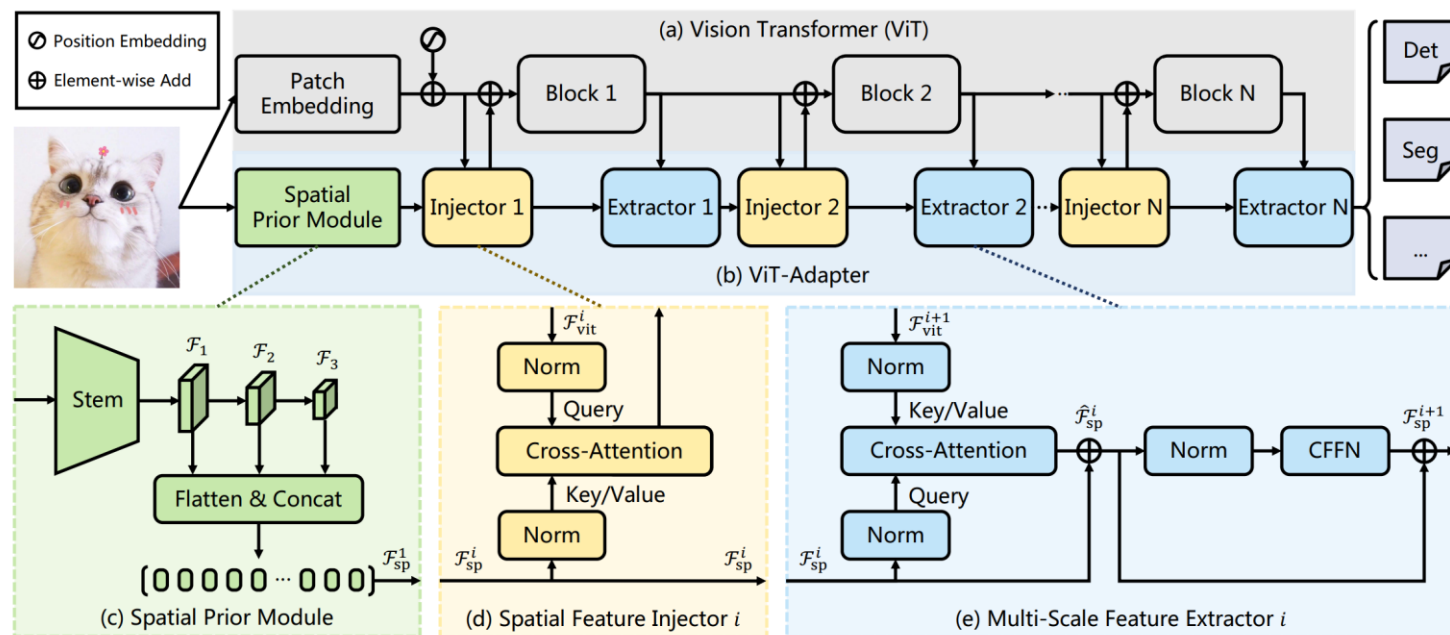
Zheng, Sixiao, et al. "Rethinking semantic segmentation from a sequence-to-sequence perspective with transformers." *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2021.

Strudel, Robin, et al. "Segmenter: Transformer for semantic segmentation." *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2021.

Ranftl, René, Alexey Bochkovskiy, and Vladlen Koltun. "Vision transformers for dense prediction." *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2021.



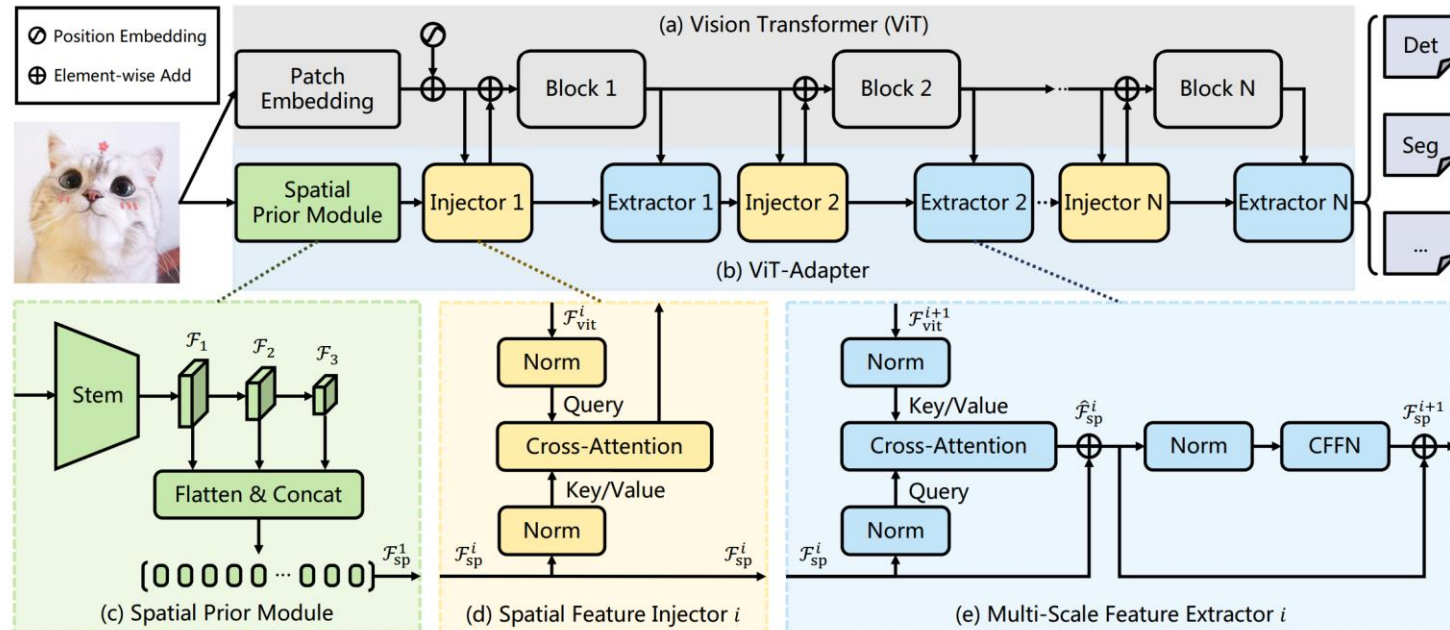
# ViT-Adapter



**Fig. 3. Overall architecture of ViT-Adapter.** (a) The Vision Transformer (ViT), whose encoder layers are divided into  $N$  equal blocks for feature interaction; (b) Our ViT-Adapter, which contains three key components; (c) The spatial prior module, which is used to model local spatial contexts from the input image; (d) The spatial feature injector for incorporating image prior into the ViT; (e) The multi-scale feature extractor for reconstructing fine-grained multi-scale features from the single-scale features of ViT.

# Spatial Prior Module

- Stem = [Conv(s=2) BN ReLU] + [Conv BN ReLU] x 2 + MaxPool(s=2)
- [Conv(s=2) BN ReLU] for each spatial feature.
  - Feature pyramid:  $\{\mathcal{F}_1, \mathcal{F}_2, \mathcal{F}_3\}$
  - Feature tokens:  $\mathcal{F}_{sp}^1 \in \mathbb{R}^{\left(\frac{HW}{8^2} + \frac{HW}{16^2} + \frac{HW}{32^2}\right) \times D}$



# Spatial Feature Injector

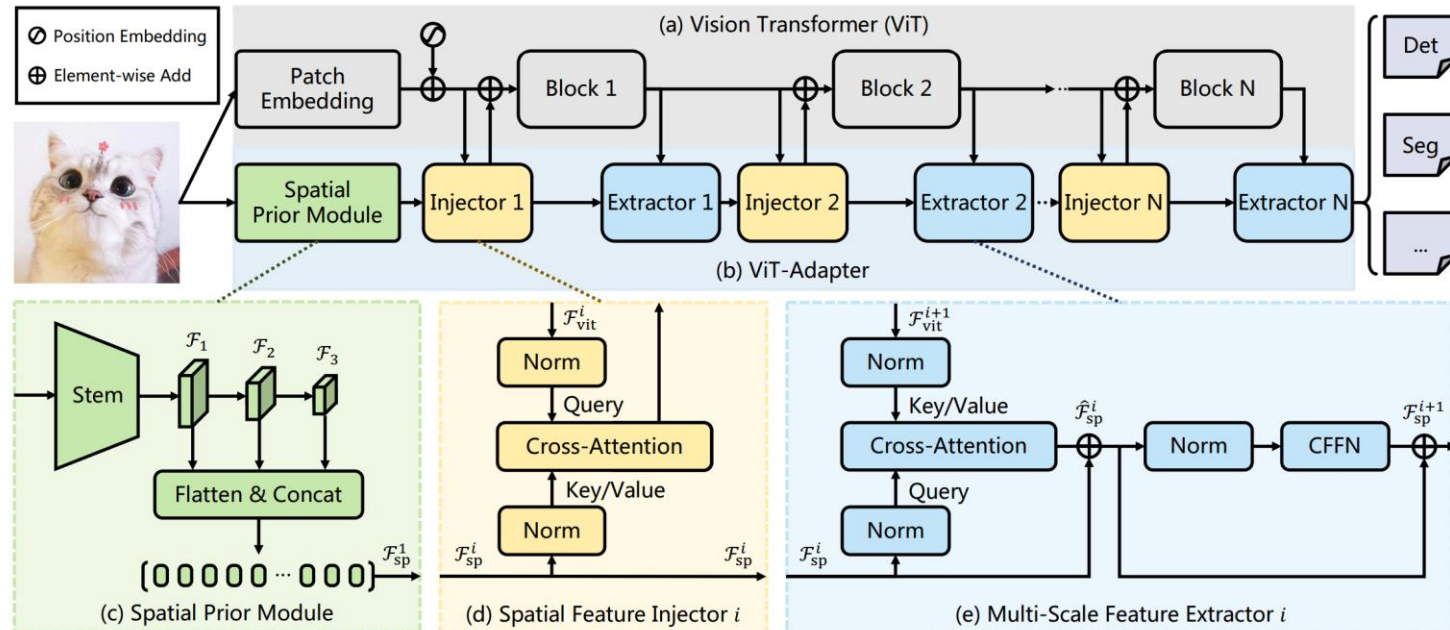
- Spatial Feature Injector is used to inject the spatial priors into ViT.

- $\mathcal{F}_{vit}^i \in \mathbb{R}^{\frac{HW}{16^2} \times D}$  as query and  $\mathcal{F}_{sp}^i \in \mathbb{R}^{(\frac{HW}{8^2} + \frac{HW}{16^2} + \frac{HW}{32^2}) \times D}$  as the key and value.

Balance the attention layer's output and the input feature (initialize with 0)

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

Deformable attention to reduce computational cost (linear complexity)



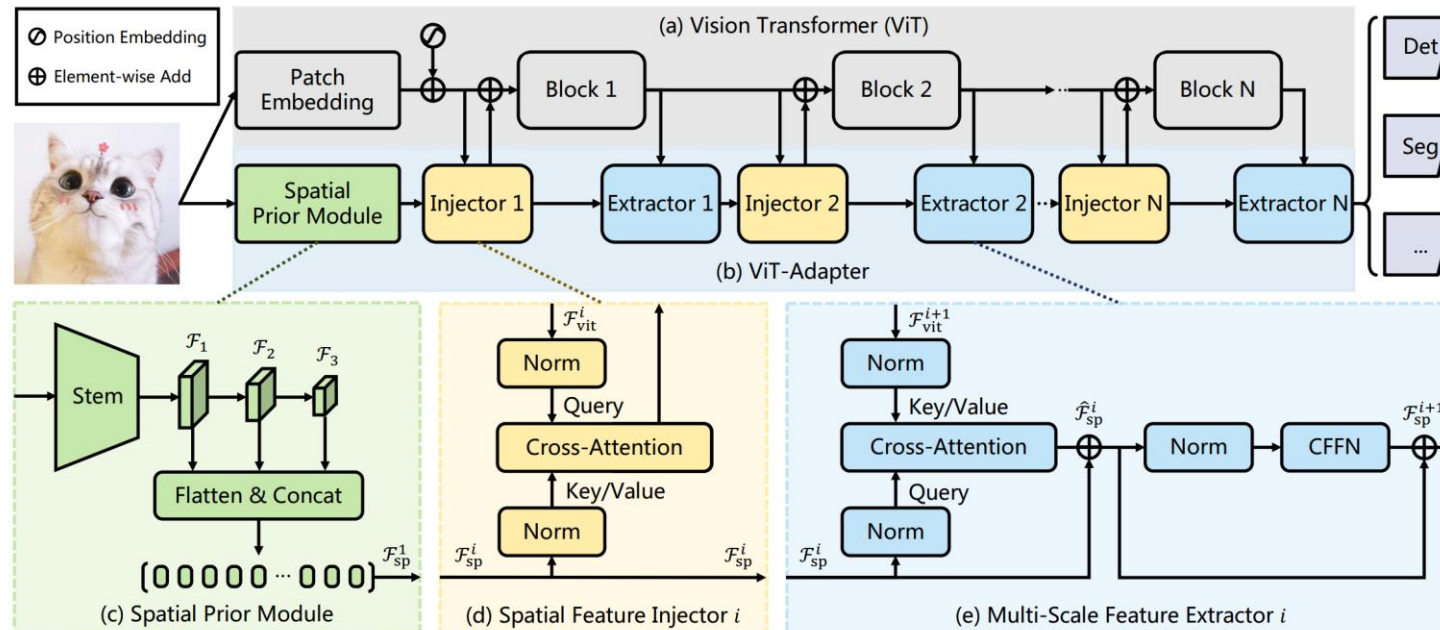
# Multi-Scale Feature Extractor

- Swap the roles of ViT's feature and the spatial feature.

- $\mathcal{F}_{sp}^i \in \mathbb{R}^{\left(\frac{HW}{8^2} + \frac{HW}{16^2} + \frac{HW}{32^2}\right) \times D}$  as query and  $\mathcal{F}_{vit}^i \in \mathbb{R}^{\frac{HW}{16^2} \times D}$  as the key and value.

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

$$\mathcal{F}_{sp}^{i+1} = \hat{\mathcal{F}}_{sp}^i + \text{CFFN}\left(\text{norm}(\hat{\mathcal{F}}_{sp}^i)\right) \quad \text{CFFN: reshape + depthwise conv + FFN}$$





# Experiments

- Object detection and instance segmentation with various frameworks.

**Table 2. Object detection and instance segmentation with Mask R-CNN on COCO val2017.** “\*” indicates following [43] to produce multi-scale feature maps with transposed convolutions. We initialize ViT-T/S/B with the DeiT released weights [71], and ViT-L with the weights from [65]. AP<sup>b</sup> and AP<sup>m</sup> represent box AP and mask AP, respectively. “MS” means multi-scale training.

Method	#Param (M)	Mask R-CNN 1×						Mask R-CNN 3× + MS					
		AP <sup>b</sup>	AP <sup>b</sup> <sub>50</sub>	AP <sup>b</sup> <sub>75</sub>	AP <sup>m</sup>	AP <sup>m</sup> <sub>50</sub>	AP <sup>m</sup> <sub>75</sub>	AP <sup>b</sup>	AP <sup>b</sup> <sub>50</sub>	AP <sup>b</sup> <sub>75</sub>	AP <sup>m</sup>	AP <sup>m</sup> <sub>50</sub>	AP <sup>m</sup> <sub>75</sub>
PVT-Tiny [74]	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-B0 [75]	23.5	38.2	60.5	40.7	36.2	57.8	38.6	41.6	63.9	45.1	38.2	60.8	40.7
PVTv2-B1 [75]	33.7	41.2	61.9	43.9	25.4	44.5	54.3	44.9	67.3	49.4	40.8	64.0	43.8
ViT-T* [43]	26.1	36.4	58.9	38.8	34.2	55.8	36.1	40.2	62.9	43.5	37.0	59.6	39.0
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 [74]	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 [75]	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 [52]	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
Conformer-S [59]	58.1	43.6	65.6	47.7	39.7	62.6	42.5	-	-	-	-	-	-
ConvNeXt-T [54]	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 [79]	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
UniFormer-S [40]	41.0	45.6	68.1	49.7	41.6	64.8	45.0	48.2	70.4	52.5	43.4	67.1	47.0
ViT-S* [43]	41.7	40.2	63.1	43.4	37.1	59.9	39.3	44.0	66.9	47.8	39.9	63.4	42.2
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 [75]	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 [52]	107.1	46.9	-	-	42.3	-	-	48.6	70.0	53.4	43.3	67.1	46.7
ViT-B* [43]	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
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* [43]	337.2	45.7	68.9	49.4	41.5	65.6	44.6	48.8	70.9	53.3	43.6	67.8	46.8
ViT-Adapter-L (ours)	347.9	48.7	70.1	53.2	43.3	67.0	46.9	50.9	72.1	55.8	44.8	69.3	48.2

**Table 3. Object detection with different frameworks on COCO val2017.** We initialize ViT-S with the DeiT released weights [71]. “\*” indicates following [43] to produce multi-scale feature maps with transposed convolutions. “global” denotes using global attention in all layers of ViT, otherwise following the settings of [43]. AP<sup>b</sup> represents box AP. “MS” means multi-scale training.

Method	AP <sup>b</sup>	AP <sup>b</sup> <sub>50</sub>	AP <sup>b</sup> <sub>75</sub>	#Param
Cascade Mask R-CNN 3× + MS				
Swin-T [52]	50.5	69.3	54.9	86M
Shuffle-T [35]	50.8	69.6	55.1	86M
PVTv2-B2 [75]	51.1	69.8	55.3	83M
Focal-T [79]	51.5	70.6	55.9	87M
ViT-S* <sub>global</sub> [43]	48.4	67.8	52.1	82M
ViT-S* [43]	47.9	67.1	51.7	82M
ViT-Adapter-S (ours)	51.5	70.1	55.8	86M
GFL 3× + MS				
Swin-T [52]	47.6	66.8	51.7	36M
PVTv2-B2 [75]	50.2	69.4	54.7	33M
ViT-S* <sub>global</sub> [43]	47.0	66.4	50.7	32M
ViT-S* [43]	46.0	65.5	49.7	32M
ViT-Adapter-S (ours)	50.0	69.1	54.3	36M
Method	AP <sup>b</sup>	AP <sup>b</sup> <sub>50</sub>	AP <sup>b</sup> <sub>75</sub>	#Param
ATSS 3× + MS				
Swin-T [52]	47.2	66.5	51.3	36M
Focal-T [79]	49.5	68.8	53.9	37M
PVTv2-B2 [75]	49.9	69.1	54.1	33M
ViT-S* <sub>global</sub> [43]	46.2	66.0	50.0	32M
ViT-S* [43]	45.2	64.8	49.0	32M
ViT-Adapter-S (ours)	49.6	68.5	54.0	36M
Sparse R-CNN 3× + MS				
Swin-T [52]	47.9	67.3	52.3	110M
Focal-T [79]	49.0	69.1	53.2	111M
PVTv2-B2 [75]	50.1	69.5	54.9	107M
ViT-S* <sub>global</sub> [43]	45.3	64.8	48.8	106M
ViT-S* [43]	44.5	64.3	48.2	106M
ViT-Adapter-S (ours)	48.1	67.0	52.4	110M

# Experiments

- Semantic segmentation

**Table 5. Performance comparisons with different methods on the ADE20K validation set.** Two different frameworks Semantic FPN [39] and UperNet [77] are used. “\*” indicates following [43] to produce multi-scale feature maps with transposed convolutions. “IN-1K” and “IN-22K” represent ImageNet-1K and 22K, respectively. “MS” denotes multi-scale testing.

Method	Pre-train	Semantic FPN 80k			UperNet 160k		
		#Param	mIoU	+MS	#Param	mIoU	+MS
PVT-Tiny [74]	IN-1K	17.0M	36.6	37.3	43.2M	38.5	39.0
XCiT-T12/16 [1]	IN-1K	8.4M	38.1	39.6	33.7M	41.5	42.2
ViT-T* [43]	IN-1K	10.2M	39.4	40.5	34.1M	41.7	42.6
ViT-Adapter-T (ours)	IN-1K	12.2M	41.7	42.1	36.1M	42.6	43.6
PVT-Small [74]	IN-1K	28.2M	41.9	42.3	54.5M	43.7	44.0
PVTv2-B2 [75]	IN-1K	29.1M	45.2	45.7	-	-	-
Swin-T [52]	IN-1K	31.9M	41.5	-	59.9M	44.5	45.8
XCiT-S12/16 [1]	IN-1K	30.4M	43.9	45.2	52.4M	45.9	46.7
Twins-PCPVT-S [14]	IN-1K	28.4M	44.3	-	54.6M	46.2	47.5
Twins-SVT-S [14]	IN-1K	28.3M	43.2	-	54.4M	46.2	47.1
ViT-S* [43]	IN-1K	27.8M	44.6	45.8	53.6M	44.6	45.7
ViT-Adapter-S (ours)	IN-1K	31.9M	46.1	46.6	57.6M	46.6	47.4
Swin-B [52]	IN-1K	91.2M	46.0	-	121.0M	48.1	49.7
XCiT-M24/16 [1]	IN-1K	90.8M	45.9	47.4	109.0M	47.6	48.6
Twins-SVT-L [14]	IN-1K	103.7M	46.7	-	133.0M	48.8	50.2
ViT-B* [43]	IN-1K	98.0M	46.4	47.6	127.3M	46.1	47.1
ViT-Adapter-B (ours)	IN-1K	104.6M	47.9	48.9	133.9M	48.1	49.2
Swin-B [52]	IN-22K	-	-	-	121.0M	50.0	51.7
Swin-L [52]	IN-22K	-	-	-	234.0M	52.1	53.5
ViT-Adapter-B (ours)	IN-22K	104.6M	50.7	51.9	133.9M	51.9	52.5
ViT-Adapter-L (ours)	IN-22K	332.0M	52.9	53.7	363.8M	53.4	54.4

# Pre-trained Weights

- DeiT, MAE, and Uni-Perceiver
  - BEiT-3 ... and so on.

**Table 8. Comparison of different pre-trained weights.** It’s worth noting that our method could enjoy diverse advanced pre-training for free, such as the mask image modeling in MAE [27] and the multi-modal pre-training in Uni-Perceiver [86].  $AP^b$  and  $AP^m$  represent box AP and mask AP, respectively. “IN-1K” denotes ImageNet-1K.

Method	Pre-train	Data	$AP^b$	$AP_{50}^b$	$AP_{75}^b$	$AP^m$	$AP_{50}^m$	$AP_{75}^m$
ViT-Adapter-B	DeiT [71]	IN-1K	47.0	68.2	51.4	41.8	65.1	44.9
	MAE [27]	IN-1K	47.8	68.2	52.4	42.3	65.5	45.6
	Uni-Perceiver [86]	Multi-Modal	48.4	70.6	53.2	43.1	67.4	46.7

**Pre-training Sources**

Name	Type	Year	Data	Repo	Paper
DeiT	Supervised	2021	ImageNet-1K	<a href="#">repo</a>	<a href="#">paper</a>
AugReg	Supervised	2021	ImageNet-22K	<a href="#">repo</a>	<a href="#">paper</a>
BEiT	MIM	2021	ImageNet-22K	<a href="#">repo</a>	<a href="#">paper</a>
MAE	MIM	2021	ImageNet-1K	<a href="#">repo</a>	<a href="#">paper</a>
Uni-Perceiver	Supervised	2022	Multi-Modal	<a href="#">repo</a>	<a href="#">paper</a>
BEiTv2	MIM	2022	ImageNet-22K	<a href="#">repo</a>	<a href="#">paper</a>