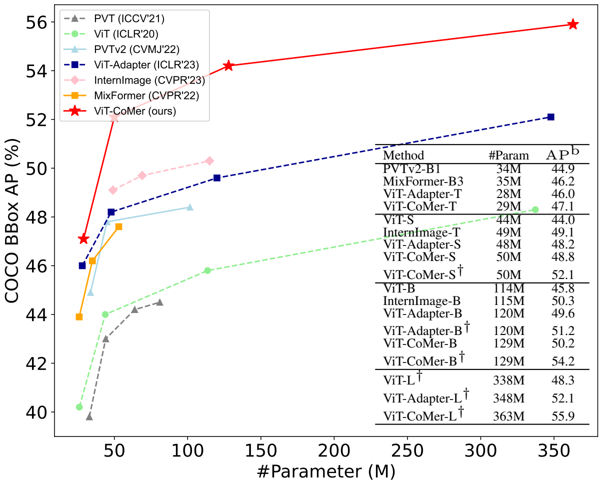
ViT-CoMer: Vision Transformer with Convolutional Multi-scale Feature Interaction for Dense Predictions

Chunlong Xia* Xinliang Wang* Feng Lv* Xin Hao* Yifeng Shi†
Baidu Inc.
{xiachunlong, wangxinliang02, lvfeng02, haoxin04, shiyifeng}@baidu.com

Abstract

*Although Vision Transformer (ViT) has achieved significant success in computer vision, it does not perform well in dense prediction tasks due to the lack of inner-patch information interaction and the limited diversity of feature scale. Most existing studies are devoted to designing vision-specific transformers to solve the above problems, which introduce additional pre-training costs. Therefore, we present a plain, pre-training-free, and feature-enhanced **ViT** backbone with **Co**nvolutional **M**ulti-scale feature int**er**action, named **ViT-CoMer**, which facilitates bidirectional interaction between CNN and transformer. Compared to the state-of-the-art, ViT-CoMer has the following advantages: (1) We inject spatial pyramid multi-receptive field convolutional features into the ViT architecture, which effectively alleviates the problems of limited local information interaction and single-feature representation in ViT. (2) We propose a simple and efficient CNN-Transformer bidirectional fusion interaction module that performs multi-scale fusion across hierarchical features, which is beneficial for handling dense prediction tasks. (3) We evaluate the performance of ViT-CoMer across various dense prediction tasks, different frameworks, and multiple advanced pre-training. Notably, our ViT-CoMer-L achieves **64.3% AP** on COCO val2017 **without extra training data**, and **62.1% mIoU** on ADE20K val, both of which are comparable to state-of-the-art methods. We hope ViT-CoMer can serve as a new backbone for dense prediction tasks to facilitate future research. The code will be released at https://github.com/Traffic-X/ViT-CoMer.*



Method	#Param	AP^b
PVTv2-B1	34M	44.9
MixFormer-B3	35M	46.2
ViT-Adapter-T	28M	46.0
ViT-CoMer-T	29M	47.1
ViT-S	44M	44.0
InternImage-T	49M	49.1
ViT-Adapter-S	48M	48.2
ViT-CoMer-S	50M	48.8
ViT-CoMer-S†	50M	52.1
ViT-B	114M	45.8
InternImage-B	115M	50.3
ViT-Adapter-B	120M	49.6
ViT-Adapter-B†	120M	51.2
ViT-CoMer-B	129M	50.2
ViT-CoMer-B†	129M	54.2
ViT-L†	338M	48.3
ViT-Adapter-L†	348M	52.1
ViT-CoMer-L†	363M	55.9

Figure 1. **Object detection performance on COCO val2017 using Mask R-CNN.** Our ViT-CoMer, with advanced pre-trained weights of ViT, outperforms other methods. "†" denotes the utilization of advanced pre-trained weights, otherwise ImageNet-1K.

1. Introduction

In recent years, owing to the release of large-scale datasets [46, 48, 49] and the development of deep learning [35], significant progress has been made in dense prediction tasks such as object detection, instance segmentation, and semantic segmentation (e.g.,YOLO series [3, 32–34], RCNN series [5, 19], DETR [6]). This progress has led to the emergence of numerous classic convolutional neural networks (CNNs), including ResNet [18], ConvNeXt [29], etc. These models leverage the local continuity and multi-scale capabilities of convolutional neural networks, enabling them to be effectively applied to dense prediction tasks. Meanwhile, inspired by the success of transformers in NLP, the Vision Transformer (ViT) [13] garners significant attention as the pioneering approach to applying transformers to visual tasks. Currently, Transformer-based network architecture designed for dense prediction tasks is mainly divided into three paradigms: plain backbone, vision-specific backbone, and adapted backbone, as shown in Figure 2. The plain backbone optimizes the use of ViT features without changing the framework of ViT, such as ViTDet [26], as shown in Figure 2(a). The vision-specific backbone (e.g., Swin [28], CMT [15], MPViT [22], PVT series [40, 41]), combines the advantages of CNN and

[*] Equal Contribution.
[†] Corresponding Author.

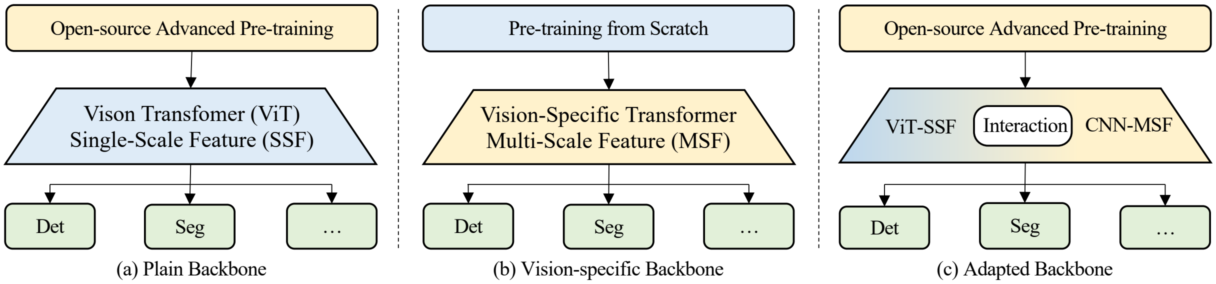

Open-source Advanced Pre-training
Vison Transfomer (ViT)
Single-Scale Feature (SSF)
Det
Seg
…
(a) Plain Backbone
Pre-training from Scratch
Vision-Specific Transformer
Multi-Scale Feature (MSF)
Det
Seg
…
(b) Vision-specific Backbone
Open-source Advanced Pre-training
ViT-SSF
Interaction
CNN-MSF
Det
Seg
…
(c) Adapted Backbone

Figure 2. **Different backbone paradigms for dense predictions.** (a) Plain backbone paradigm can leverage open-source advanced pre-trained weights (e.g., BEiT series [2, 31, 42], DINOv2 [30]). However, its drawback lies in the limited scale diversity of feature representation, which is insufficient to meet the requirements of dense predictions. (b) Vision-specific backbone paradigm designs a multi-scale feature framework that effectively addresses dense predictions. However, each structural modification requires retraining the pre-trained weights from scratch on large-scale image datasets. (c) Adapted backbone paradigm integrates the advantages of both CNN and transformer. It can directly load advanced pre-training and achieve fusion interaction between multi-scale convolutional features and transformer features, which is beneficial for dense predictions.

Transformer to redesign the network structure, which helps them achieve better performance in dense prediction tasks, as shown in Figure 2(b). The adapted backbone, shown in Figure 2(c), is based on the plain ViT, which only introduces CNN features by adding additional branches and can directly load various open-source and powerful ViT pre-trained weights to improve ViT performance on dense prediction tasks. In this work, we present a plain, pre-training-free, and feature-enhanced ViT backbone named ViT-CoMer, which can directly load various open-source and advanced pre-trained weights. Specifically, we design two core modules: the **M**ulti-**R**eceptive Field **F**eature **P**yramid module (MRFP) and the **C**NN-**T**ransformer Bidirectional Funsion **I**nteraction module (CTI). MRFP can supplement ViT with more abundant multi-scale spatial information; CTI can fuse multi-scale features from CNN and Transformer, facilitating the model with a more powerful feature representation ability. In the weight initialization process of ViT-CoMer, the ViT module directly uses the open-source pre-training, and the rest use random initialization. As shown in Figure 1, our model performs better when using advanced pre-trained weights of ViT. Our main contributions are as follows:

- We propose a novel dense prediction backbone by combining the plain ViT with CNN features. It effectively leverages various open-source pre-trained ViT weights and incorporates spatial pyramid convolutional features that address the lack of interaction among local ViT features and the challenge of single-scale representation.
- We design a multi-receptive field feature pyramid module and a CNN-Transformer bidirectional fusion interaction module. The former can capture various spatial features, the latter performs multi-scale fusion across hierarchical features to obtain richer semantic information, which is beneficial for handling dense prediction tasks.
- We evaluate our proposed ViT-CoMer on several challenging dense prediction benchmarks, including object detection, instance segmentation and semantic segmentation. The experimental results demonstrate that our method significantly enhances the capabilities of the plain ViT. Especially, when utilizing advanced open-source pre-training such as DINOv2 [30], ViT-CoMer can consistently outperform SOTA methods under fair comparison conditions. Notably, our ViT-CoMer-L, with advanced pre-training, achieves **64.3% AP**, which is the best record on COCO val2017 **without training on extra detection data (e.g., Objects365)**. Moreover, ViT-CoMer-L attains **62.1% mIoU** on the ADE20K val, which is comparable with SOTA methods.

2. Related Work

2.1. Plain backbones

ViT [13] is the first work introducing the transformer to the image classification task and achieving impressive results. ViTDet [26] is a plain, non-hierarchical detector based on ViT by incorporating a simple feature pyramid module. However, the performance of ViTDet exhibits a gap compared to state-of-the-art methods. One potential reason is that the feature representation of ViT might not be sufficiently rich. Nonetheless, dense prediction models require a strong ability for multi-scale perception. Our work combines multi-scale enhanced convolutional features with ViT features, enabling the model to extract rich multi-scale features when dealing with dense prediction tasks.

2.2. Vision-specific backbones

Vision-specific backbones are primarily designed to alleviate challenges in ViT, such as the non-hierarchical feature, and the lack of interaction among local features. Swin-

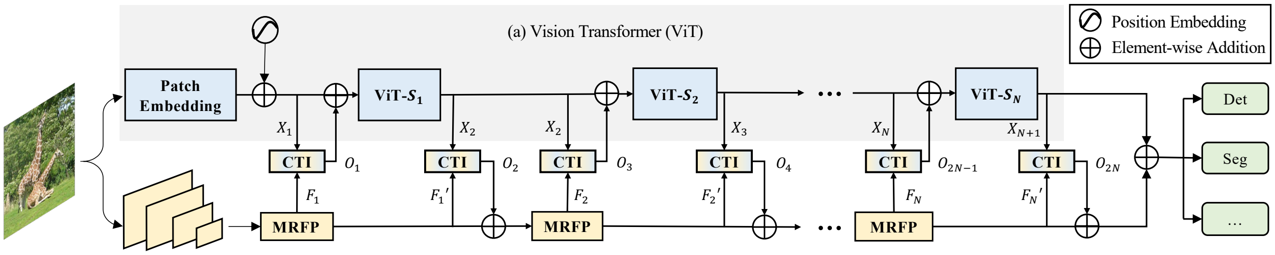


Figure 3. **The overall architecture of ViT-CoMer.** ViT-CoMer is a two-branch architecture consisting of three components: (a) a plain ViT with L layers, which is evenly divided into N stages for feature interaction. (b) a CNN branch that employs the proposed **M**ulti-**R**eceptive Field **F**eature **P**yramid (MRFP) module to provide multi-scale spatial features, and (c) a simple and efficient **C**NN-**T**ransformer Bidirectional Fusion **I**nteraction (CTI) module to integrate the features of the two branches at different stages, enhancing semantic information.

Transformer [28] employs shifted windows to alleviate the lack of interaction among local information in ViT. Simultaneously, it constructs multi-scale features to adapt the dense predictions. PVT [40] constructs a feature pyramid structure to address the limitations of single-scale features in ViT, which simplifies the structure of the transformer and effectively reduces the computational complexity. MixFormer [8] utilizes a bidirectional feature interaction operator with convolution and self-attention to enhance feature representation. iFormer [36] analyzes the advantages of CNN and Transformer architectures at high and low frequencies. MetaFormer [50] introduces a general hierarchical network architecture that utilizes pooling instead of attention, which achieves favorable results in various vision tasks. UniFormer [24] cascades CNN and attention within a block, which integrates the advantages of both CNN and Transformer. Vision-specific backbones alter the ViT structure, which prevents them from directly using existing powerful pre-trained weights, e.g., BEiT series [2, 31, 42]. Our work preserves the original ViT, allowing it to load open-source pre-trained weights based on ViT directly. This enables our model to rapidly acquire enhanced generalization performance.

2.3. Adapted backbones

ViT-Adapter [9] presents a ViT framework that integrates spatial prior information. It leverages the advantages of ViT's pre-trained weights. ViT-adapter needs to full-finetune during training, resulting in an impressive performance in dense prediction tasks. Meanwhile, it lacks feature interaction among spatial prior information. VPT [21] introduces a method that freezes the pre-trained weights of ViT and updates only the parameters of the adapter module during training. While this approach can yield results comparable to the full-finetuning method in some tasks, it doesn't perform as well as full-finetuning in semantic segmentation. LoRand [47] is also an algorithm that preserves the weights of ViT and trains only the adapter module. which only need to train 1%–3% of the overall training parameters. However, its performance is not as effective as the full-finetuning approach. Our work enhances spatial hierarchical features through feature fusion and employs the full-finetuning approach to optimize the model during training, which effectively boosts the performance of the model.

3. The ViT-CoMer Method

3.1. Overall Architecture

The overall architecture of ViT-CoMer is illustrated in Figure 3, which includes three parts: (a) Plain ViT. (b) Multi-receptive field feature pyramid module (MRFP). (c) CNN-Transformer bidirectional fusion interaction module (CTI). Firstly, for the ViT branch (see Figure 3(a)), an input image with the shape of $H \times W \times 3$ is fed into the patch embedding to obtain features with a resolution reduction of $1/16$ of the original image. Meanwhile, for the other branch, this image passes through a stack of convolutions to obtain feature pyramid C_3, C_4, and C_5 with resolutions of $1/8$, $1/16$, and $1/32$, and each of them contains D-dimensional feature maps. Secondly, both of the two branch features pass through N stage feature interactions. At each stage, the feature pyramid will first be enhanced through the MRFP module, and then bidirectionally interact with the feature of ViT through the CTI module, which can obtain multi-scale features with rich semantic information. CTI operates at the beginning and end of each stage. After N stage feature interactions, the features from two branches are added at each scale for dense prediction tasks.

3.2. Multi-Receptive Field Feature Pyramid

The multi-receptive field feature pyramid module consists of a feature pyramid and multi-receptive field convolutional layers. The feature pyramid can provide rich multi-scale information, while the latter expands the receptive field through different convolution kernels, enhancing the long-range modeling ability of CNN features. The mod-

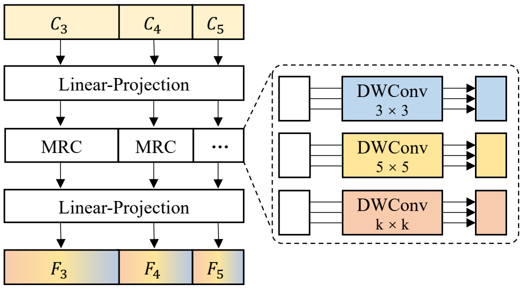


Figure 4. **Multi-Receptive Field Feature Pyramid module.** The C_3, C_4, and C_5 features are first dimensionally reduced through a linear projection layer. Subsequently, these features are divided into multiple groups along the channel dimension. Different groups employ varied kernel sizes of DWConv to enrich receptive field representation, MRC represents a multi-receptive field convolution operation. Finally, the features are restored to their original dimensions through dimensional expansion.

ule is shown in Figure 4. MRFP is composed of two linear projection layers and a set of depth-wise separable convolutions with multi-receptive fields. Specifically, the input of the module is a set of multi-scale features $\{C_3, C_4, C_5\}$, we flatten and concatenate these feature maps into feature tokens $C \in \mathbb{R}^{(\frac{HW}{8^2}+\frac{HW}{16^2}+\frac{HW}{32^2})\times D}$, which first passes through a linear projection layer to obtain dimensionally reduced features, and then the features are divided into M groups on the channel dimension. Different groups of features correspond to convolutional layers with different receptive fields $(e.g., k = 3 \times 3, 5 \times 5)$. Finally, the processed features are concatenated and dimensionally increased through the linear projection layer. The process can be represented as:

$$F = FC(DWConv(FC(C))), \tag{1}$$

where $FC(\cdot)$ is linear projection, $DWConv(\cdot)$ is a set of depth-wise convolutions with different kernel sizes.

3.3. CNN-Transformer Bidirectional Fusion Interaction

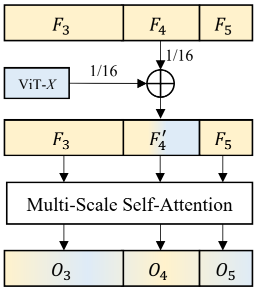


Figure 5. **CNN-Transformer Bidirectional Fusion Interaction module.** $\{F_3,\ F_4,\ F_5\}$ are multi-scale CNN features obtained through the MRFP module. We add F_4 and X from the ViT branch and use a multi-scale self-attention module to unify the two modal features, ultimately achieving information interaction and obtaining updated features.

We propose a cross-architecture feature fusion method named CTI, as shown in Figure 5. It introduces CNN's multi-scale features without altering the ViT structure. Simultaneously, through bidirectional interaction, we alleviate the problems of the lack of inner-patch information interaction and the non-hierarchical feature in ViT, while further enhancing the CNN's long-range modeling ability and semantic representation. In order to fuse the ViT feature $X \in \mathbb{R}^{\frac{H}{16}\times\frac{W}{16}\times D}$ and the multi-scale feature $\{F_3, F_4, F_5\}$ obtained through the MRFP module, it can be represented as $F \in \mathbb{R}^{(\frac{HW}{8^2}+\frac{HW}{16^2}+\frac{HW}{32^2})\times D}$. We directly add features X and F_4, yielding the set F', expressed as $F' = \{F_3, F_4', F_5\}$, which aggregates multi-scale features from different architectures. However, due to architectural differences, they exhibit bias in modality representation (e.g., high-low frequency semantics, and global-local information). To address this, we employ self-attention to unify CNN and Transformer features, reinforcing the representation invariance against modality discrepancy. The process can be described as:

$$O = FFN(Attention(norm(F'))), \tag{2}$$

where the $norm(\cdot)$ is LayerNorm [1], $Attention(\cdot)$ is multi-scale deformable attention [54] and $FFN(\cdot)$ is feed-forward network. Finally, we align the feature map sizes of O_3 and O_5 to O_4 through bilinear interpolation and add X as the input of the next ViT layer. Moreover, since F' contains multi-scale features with resolutions of $1/8, 1/16$, and $1/32$, self-attention can facilitate interaction among multi-scale features, and enable the model to better capture multi-scale information in images. This diverges from the traditional transformer architecture, which employs self-attention solely on a single-scale feature. By effectively fusing multi-scale CNN and Transformer features, the model gains enhanced modeling capability. Regarding features fused across architectures, bidirectional interaction is employed to update features of the ViT and CNN branches. Specifically, for the i-th stage, at the beginning of stage i, the two branch features are fused, and then the fused features are injected into the ViT branch. The process can be formulated as:

$$\hat{X}_i = \alpha * O_{2i-1} + X_i, \tag{3}$$

where $\hat{X}_i$ is the updated feature of the ViT branch, α is a learnable variable initialized to zero, it minimizes the influence of randomly initialized CNN architecture on ViT during early training. At the end of stage i, the process is repeated to inject features into the CNN branch, represented

Method	#Param	Mask R-CNN 1× schedule						Mask R-CNN 3× schedule					
		$\mathrm{AP^b}$	$\mathrm{AP^b_{50}}$	$\mathrm{AP^b_{75}}$	$\mathrm{AP^m}$	$\mathrm{AP^m_{50}}$	$\mathrm{AP^b_{75}}$	$\mathrm{AP^b}$	$\mathrm{AP^b_{50}}$	$\mathrm{AP^b_{75}}$	$\mathrm{AP^m}$	$\mathrm{AP^m_{50}}$	$\mathrm{AP^b_{75}}$
PVT-T [40]	33M	36.7	59.2	39.3	35.1	56.7	37.3	39.8	62.2	43.0	37.4	59.3	39.9
ViT-Adapter-T [9]	28M	41.1	62.5	44.3	37.5	59.7	39.9	46.0	67.6	50.4	41.0	64.4	44.1
ViT-T [25]	26M	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 [26]	27M	35.7	57.7	38.4	33.5	54.7	35.2	40.4	63.3	43.9	37.1	60.1	39.3
PVT-S [40]	44M	40.4	**62.9**	43.8	37.8	**60.1**	40.3	43.0	65.3	46.9	39.9	62.5	42.8
ViT-CoMer-T (ours)	29M	**42.1**	62.7	**45.3**	**38.0**	**60.1**	**40.5**	**47.1**	**67.8**	**51.5**	**41.5**	**64.8**	**44.3**
ConvNeXt-T [29]	48M	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 [45]	49M	44.8	67.7	49.2	41.0	64.7	44.2	47.2	69.4	51.9	42.7	66.5	45.9
SPANet-S [51]	48M	44.7	65.7	48.8	40.6	62.9	43.8	-	-	-	-	-	-
MixFormer-B4 [8]	53M	45.1	67.1	49.2	41.2	64.3	44.1	47.6	69.5	52.2	43.0	66.7	46.4
ViT-Adapter-S [9]	48M	44.7	65.8	48.3	39.9	62.5	42.8	48.2	69.7	52.5	42.8	66.4	45.9
Twins-B [11]	76M	45.2	67.6	49.3	41.5	64.5	44.8	48.0	69.5	52.7	43.0	66.8	46.6
Swin-S [28]	69M	44.8	66.6	48.9	40.9	63.4	44.2	48.5	70.2	53.5	43.3	67.3	46.6
Flatten-PVT-T [16]	49M	44.2	67.3	48.5	40.2	63.8	43.0	46.5	68.5	50.8	42.1	65.4	45.1
NAT-S [17]	70M	-	-	-	-	-	-	48.4	69.8	53.2	43.2	66.9	46.5
ViT-S [25]	44M	40.2	63.1	43.4	37.1	60.0	38.8	44.0	66.9	47.8	39.9	63.4	42.2
ViTDet-S [26]	46M	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-CoMer-S (ours)	50M	45.8	67.0	49.8	40.5	63.8	43.3	48.8	69.4	53.5	43.0	66.9	46.3
ViT-CoMer-S‡ (ours)	50M	**48.6**	**70.5**	**53.1**	**42.9**	**67.0**	**45.8**	**52.1**	**73.1**	**57.1**	**45.8**	**70.2**	**49.4**
PVTv2-B5 [41]	102M	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 [28]	107M	46.9	-	-	42.3	-	-	48.6	70.0	53.4	43.3	67.1	46.7
InternImage-B [43]	115M	48.8	70.9	54.0	44.0	67.8	47.4	50.3	71.4	55.3	44.8	68.7	48.0
ViT-Adapter-B [9]	120M	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-B [25]	114M	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 [26]	121M	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-CoMer-B (ours)	129M	47.6	68.9	51.9	41.8	65.9	44.9	50.2	70.7	54.9	44.0	67.9	47.4
ViT-CoMer-B‡ (ours)	129M	**52.0**	**73.6**	**57.2**	**45.5**	**70.6**	**49.0**	**54.2**	**75.2**	**59.4**	**47.6**	**72.7**	**51.6**
ViT-L† [25]	337M	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† [26]	351M	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† [9]	348M	48.7	70.1	53.2	43.3	67.0	46.9	52.1	73.8	56.5	46.0	70.5	49.7
ViT-CoMer-L† (ours)	363M	51.4	73.5	55.7	45.2	70.3	48.5	52.9	73.8	57.5	46.4	71.1	50.4
ViT-CoMer-L‡ (ours)	363M	**53.4**	**75.3**	**58.9**	**46.8**	**72.0**	**50.9**	**55.9**	**77.3**	**61.5**	**49.1**	**74.5**	**53.5**

Table 1. **Object detection and instance segmentation with Mask R-CNN on COCO val2017.** "†" denotes the use of ImageNet-22K pre-training, "‡" denotes the use of DINOv2 [30], while the default is to use ImageNet-1K pre-training.

as:

$$\hat{F}_i = O_{2i} + F_i{}', \tag{4}$$

where $\hat{F}_i$ is the updated feature of the CNN branch, the number of stages i is determined based on the depth of the ViT. The cross-architecture feature fusion and bidirectional interaction enable the utilization of features from multi-scales and multi-levels, enhancing the model's expressive and generalization abilities. Simultaneously, the proposed components might be easily integrated into other advanced models and perform better in dense prediction tasks.

4. Experiment

We select typical tasks in dense prediction: object detection, instance segmentation, and semantic segmentation and conduct extensive experiments (with different model sizes, algorithm frameworks, and configurations) on COCO [27] and ADE20K [53] datasets, to verify the effectiveness of ViT-CoMer. Meanwhile, we use various pre-training of ViT, including weights pre-trained on ImageNet-1K, ImageNet-22K, and multi-modal data. ViT-CoMer achieves results that are superior to existing SOTA ViT-based methods (e.g., ViTDet [26], ViT-Adapter [9]) and comparable to vision-specific advanced methods. In addition, we perform ablation experiments on the designed modules and qualitative experiments for dense prediction tasks. These results indicate that ViT-CoMer can promote plain ViT to attain superior performance, and can be migrated as a robust backbone to various dense prediction task frameworks.

4.1. Object Detection and Instance Segmentation

Settings. We utilize the MMDetection [7] framework to implement our method and perform object detection and instance segmentation experiments on the COCO dataset.

Method	AP^{b}	$\mathrm{AP}^{\mathrm{b}}_{50}$	$\mathrm{AP}^{\mathrm{b}}_{75}$	#Param
Cascade Mask R-CNN 3× +MS schedule				
Swin-T [28]	50.5	69.3	54.9	86M
Shuffle-T [20]	50.8	69.6	55.1	86M
PVTv2-B2 [41]	51.1	69.8	55.3	83M
Focal-T [45]	51.5	70.6	55.9	87M
Swin-S [45]	**51.9**	**70.7**	56.3	107M
ViT-S [25]	47.9	67.1	51.7	82M
ViT-Adapter-S [9]	51.5	70.1	55.8	86M
ViT-CoMer-S (ours)	**51.9**	70.6	**56.4**	89M
ATSS 1× schedule				
ViT-T [25]	34.8	52.9	36.9	14M
ViT-Adapter-T [9]	39.3	57.0	42.4	16M
ViT-CoMer-T (ours)	**40.4**	**58.4**	**43.6**	17M
GFL 1× schedule				
ViT-T [25]	35.7	53.6	38.1	14M
ViT-Adapter-T [9]	40.3	58.2	43.4	16M
ViT-CoMer-T (ours)	**40.7**	**58.9**	**43.7**	17M

Table 2. **Object detection with different frameworks on COCO val2017.** "+MS" means multi-scale training.

The object detection and instance segmentation frameworks encompass Mask R-CNN, Cascade Mask R-CNN, ATSS, and GFL. Referring to PVT, we conduct experiments with a training schedule of 1× (12 epochs) or 3× (36 epochs). We use a total batch size of 16, utilize the AdamW optimizer with a learning rate of 1×10^{-4} and a weight decay of 0.05.

Comparisons with different backbones and frameworks. Table 1 shows the comparisons between ViT-CoMer and various scales of plain ViT, vision-specific and adapted backbones on Mask R-CNN 1× and 3× schedules. It can be seen that under similar model sizes, ViT-CoMer outperforms other backbones in the two typical dense prediction tasks of COCO object detection and instance segmentation. For instance, ViT-CoMer-S demonstrates a notable increase of +5.6% (+4.8%) in box mAP and +3.4% (+3.1%) in mask mAP compared to plain ViT-S under the 1× (3×) schedule. ViT-CoMer-S achieves superior detection results compared to ViT-L while utilizing only $1/6$ of the parameters. Furthermore, our approach still shows notable improvements over vision-specific and adapted backbones, such InternImage [43] and ViT-Adapter [9].

We further evaluate ViT-CoMer with different detection frameworks, the results are shown in Table 2. It can be seen that our approach consistently outperforms other backbones across various frameworks, model sizes, and configurations.

Results on different pre-trained weights. We conduct experiments on Mask R-CNN (3× schedule) using different pre-trained weights, and the results are shown in Table 3. Specifically, ViT-CoMer-B with multi-modal pre-training [31], can achieve +1.7% AP^{b} and +1.7% AP^{m} gain compared to ImageNet-1K [39]. Furthermore, we compared more pre-training on ViT-CoMer-L, among which self-supervised pre-training [30] achieved significant results. Compared with ImageNet-22K [37] pre-training, it achieves +3.0% AP^{b} and +2.7% AP^{m} gains. These results demonstrate that our ViT-CoMer can easily leverage diverse, open-source, large-scale pre-training to improve performance on downstream tasks.

Pre-training	Method	AP^{b}	AP^{m}
IN-1K	ViT-CoMer-B	50.2	44.0
IN-22K	ViT-CoMer-L	52.9	46.4
MM	ViT-CoMer-B	51.9	45.7
	ViT-CoMer-L	54.9	48.3
SSL	ViT-CoMer-L	55.9	49.1

Table 3. **Comparisons of different pre-training for object detection and instance segmentation tasks with Mask R-CNN on COCO val2017.** IN-1K, IN-22K, MM and SSL respectively represent the use of ImageNet-1K [39], ImageNet-22K [37], multi-modal, self-supervised learning pre-training.

Method	Backbone	Pre-training	#P	AP^{b}
DINO-D-DETR [52]	Swin-L [28]	IN-22K	284M	58.5
HTC++ [28]	ViT-Adapter-L [9]	BEiTv2 [31]	401M	60.5
HTC++ [28]	ViT-Adapter-L [9]	BEiTv2+O365	401M	62.6
CMask R-CNN [5]	ViTDet-H [26]	IN-1K	692M	61.3
Co-DETR [56]	Swin-L [28]	IN-22K	218M	60.7
Co-DETR [56]	ViT-CoMer-L (ours)	BEiTv2 [31]	363M	62.1
Co-DETR [56]	ViT-CoMer-L (ours)	BEiTv2* [9]	363M	64.3

Table 4. **Comparisons with previous SOTA on COCO val 2017.** O365 indicates the Objects365 dataset is used during training. * indicates a variant version of BEiTv2 used in ViT-Adapter.

Comparisons with state-of-the-arts. In order to further improve the performance, we conduct experiments based on Co-DETR [56], using ViT-CoMer as the backbone, and initializing the model with multi-modal pre-training BEiTv2. As shown in Table 4, our approach outperforms the existing SOTA algorithms without extra training data on COCO val2017, which strongly demonstrates the effectiveness of ViT-CoMer.

4.2. Semantic Segmentation

Settings. Our semantic segmentation experiments are based on the ADE20K dataset with MMSegmentation [12]. We select UperNet [44] as the basic framework. The training configuration remains consistent with Swin [28], encompassing training for 160,000 iterations. The batch size is set to 16, and the AdamW optimizer is used. The learning rate and weight decay parameters are tuned to 2×10^{-5} and 0.05, respectively.

Comparisons with different backbones. Table 5 presents the comparisons of both single-scale and multi-scale mIoU between ViT-CoMer and various backbones, including plain ViT, vision-specific backbones, and adapted backbones in semantic segmentation tasks. It shows that,

Method	UperNet 160k		
	#Param	mIoU	+MS
PVT-T [41]	43.2M	38.5	39.0
ViT-T [25]	34.1M	41.7	42.6
ViT-Adapter-T [9]	36.1M	42.6	43.6
ViT-CoMer-T (ours)	38.7M	**43.0**	**44.3**
PVT-S [41]	54.5M	43.7	44.0
Swin-T [28]	59.9M	44.5	45.8
Twins-SVT-S [11]	54.4M	46.2	47.1
ViT-S [25]	53.6M	44.6	45.7
ViT-Adapter-S [9]	57.6M	46.2	47.1
ViT-CoMer-S (ours)	61.4M	**46.5**	**47.7**
Swin-B [28]	121.0M	48.1	49.7
Twins-SVT-L [11]	133.0M	**48.8**	**50.2**
ViT-B [25]	127.3M	46.1	47.1
ViT-Adapter-B [9]	133.9M	**48.8**	49.7
ViT-CoMer-B (ours)	144.7M	**48.8**	49.4
Swin-L† [28]	234.0M	52.1	53.5
ViT-Adapter-L† [9]	363.8M	53.4	54.4
ViT-CoMer-L (ours)†	383.4M	**54.3**	**55.6**

Table 5. **Semantic segmentation results on the ADE20K val. "+MS" means multi-scale testing.** "†" denotes the use of ImageNet-22K pre-trained weight, while the default is to use ImageNet-1K pre-training.

Pre-training	Method	mIoU	+MS
IN-22K	Swin-L [28]	52.1	53.5
	ViT-Adapter-L [9]	53.4	54.4
	ViT-CoMer-L (ours)	**54.3**	**55.6**
MM	ViT-Adapter-L [9]	55.0	55.4
	ViT-CoMer-L (ours)	**56.3**	**56.8**

Table 6. **Comparisons of different pre-trained weights for semantic segmentation with UperNet on ADE20K val.** IN-22K and MM respectively represent the use of ImageNet-22K and multi-modal pre-trained weights.

under comparable model sizes, our method surpasses the ViT and many vision-specific backbones. For instance, our ViT-CoMer-S achieves 47.7% MS mIoU, outperforming many strong counterparts such as Swin-T (+1.9%) and ViT-Adapter-S (+0.6%). Similarly, ViT-CoMer-L reports a competitive performance of 55.6% MS mIoU, which is 2.1% higher than Swin-L and 1.2% higher than ViT-Adapter-L. These equitable comparisons demonstrate the effectiveness and universality of our ViT-CoMer in the semantic segmentation task.

Method	Backbone	Pre-train	#P	mIoU	+MS
Mask DINO [23]	Swin-L [28]	IN-22K	223M	59.5	60.8
Mask2Former [10]	ViT-Adapter-G [9]	BEiTv3 [42]	1.9B	62.0	62.8
Mask2Former [10]	ViT-Adapter-G [9]	EVA [14]	1.0B	61.5	62.3
Mask2Former [10]	RevCol-H [4]	-	2.4B	60.4	61.0
Mask2Former [10]	ViT-Adapter-L [9]	BEiTv2 [31]	571M	61.2	61.5
Mask2Former [10]	ViT-CoMer-L (ours)	BEiTv2 [31]	604M	61.7	62.1

Table 7. **Comparisons with previous SOTA on ADE20K dataset for semantic segmentation.**

Method	Components			AP^b	AP^m
	MRFP	CTI (to V)	CTI (to C)		
ViT-S	×	×	×	40.2	37.1
	✓	×	×	41.5	38.2
	✓	✓	×	43.3	39.3
	✓	✓	✓	**45.8**	**40.5**

Table 8. **Ablation studies of key components.** Our proposed components collectively bring 5.6 AP^b and 3.4 AP^m gains. CTI (to V) indicates that the fused features are injected into the ViT branch, whereas CTI (to C) means that the fused features are injected into the CNN branch.

Comparisons with different pre-trained weights. Table 6 is the result of using different pre-trained weights on UperNet. When using the ImageNet-22K pre-trained weights [38], our ViT-CoMer-L attains 55.6% MS mIoU, exceeding ViT-Adapter-L by 1.2% mIoU. Then, we initialize ViT-CoMer-L with the multi-modal pre-training [55], which benefits our model with impressive gains of 2.0% mIoU, exceeding ViT-Adapter-L by 1.4%. These significant and consistent improvements suggest that our method can effectively improve plain ViT and fully utilize various open-source ViT-based pre-trained weights, enabling the model to perform better in semantic segmentation.

Comparisons with state-of-the-arts. To enhance the performance even more, we conduct experiments based on Mask2Former [10] using ViT-CoMer as the backbone, and initializing the model with multi-modal pre-training BEiTv2. As shown in Table 7, our method achieves comparable performance to SOTA methods on ADE20K with fewer parameters.

4.3. Ablation Study

Settings. We conduct ablation experiments on the ViT-CoMer-S, using Mask R-CNN (1× schedule) for object detection and instance segmentation tasks. The total batch size used during the training process is 16, the optimizer employed is AdamW, and the learning rate and weight decay parameters are set to 1×10^{-4} and 0.05, respectively.

Ablation for components. We gradually add the proposed submodules to the ViT-S, ultimately evolving the model into the ViT-CoMer. The results of the ablation experiment are shown in Table 8. When MRFP is used to provide multi-scale and multi-receptive-field features of CNN to plain ViT (features are directly added), it results in improvements of 1.3% AP^b and 1.1% AP^m. Furthermore, we replace the "directly added" operation with CTI proposed in this work. When only CTI (to V) is used, the model improves by 1.8% AP^b and 1.1% AP^m; when CTI (to V) and CTI (to C) are used simultaneously, the performance further

N	AP^{b}	AP^{m}	#Param
0	40.2	37.1	43.8M
2	45.1	40.1	48.2M
4	**45.8**	**40.5**	50.3M
6	45.6	**40.5**	52.5M

Table 9. **Ablation of the number of bidirectional fusion interaction modules.** The model performs best when N=4.

k	AP^{b}	AP^{m}	#Param
3	45.7	40.4	50.29M
3, 5	**45.8**	**40.5**	50.31M
3, 5, 7	45.5	40.2	50.33M
3, 5, 7, 9	45.4	40.0	50.36M

Table 10. **Ablation of the setting of kernel size in MRFP .** The model performs best when k=3 and 5.

significantly improves by 2.5% AP^{b} and 1.2% AP^{m}. Overall, compared to plain ViT, our ViT-CoMer achieved significant improvements of 5.6% AP^{b} and 3.4% AP^{m}. The experimental results demonstrate that our proposed MRFP and CTI modules can significantly enhance the ability of plain ViT, making it well adapted to dense prediction tasks.

Number of bidirectional fusion interaction. In Table 9, we analyze the impact of the number of bidirectional fusion interaction modules. We observe that as N increases, the model accuracy reaches a plateau, and introducing more interactions does not consistently enhance performance. Therefore, we set N to 4 by default.

Different kernel size in MRFP. Table 10 illustrates the influence of varying kernel sizes on the MRFP. The results show that the number of parameters increases as the kernel size increases. Simultaneously, we observe AP^{b} and AP^{m} peaks when using kernel sizes 3 and 5, therefore we adopt these as the default settings.

4.4. Scalability

Our method also can be employed with hierarchical vision transformers such as Swin. We apply our approach to Swin-T with Mask R-CNN (1x schedule) for object detection. As illustrated in Table 11, our method improves the performance of Swin-T by +2.1% box AP and +1.2% mask AP. Since the Swin architecture already introduces inductive biases, the improvements are somewhat lower compared to a plain ViT. Nonetheless, these results still substantiate the scalability of our approach.

4.5. Qualitative Results

According to iFormer [36], plain ViT tends to capture global and low-frequency features in images due to self-attention operations, while CNN tends to capture local and high-frequency features in the image due to convolution operations. However, in dense prediction tasks, various objects will appear in the image with different sizes and densities, which requires the model to have the ability to simultaneously extract and capture local and global, high-frequency and low-frequency features. We qualitatively evaluate the difference between plain ViT and our proposed ViT-CoMer by visualizing feature maps on different layers (downsampling $1/4$, $1/8$, $1/16$, and $1/32$) for instance segmentation and object detection tasks. The qualitative visualization results are shown in Figure 6. It can be seen that compared to the plain ViT, our ViT-CoMer yields more fine-grained multi-scale features, thereby enhancing the model's object localization capability.

Method	AP^{b}	AP^{m}	#Param
Swin-T	42.7	39.3	48M
Swin-CoMer-T (ours)	**44.8**	**40.5**	54M

Table 11. **Scalability of the Swin Transformer.**

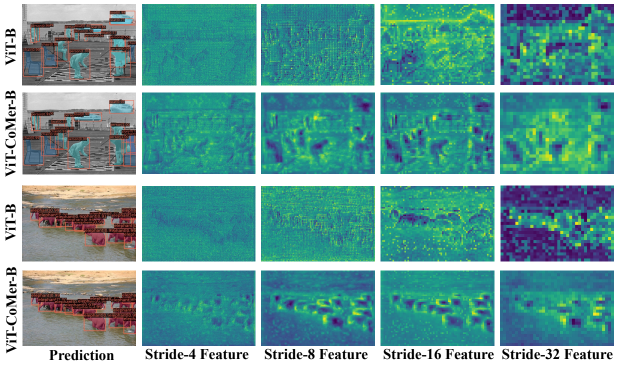


Figure 6. **Visualization of feature maps for object detection and instance segmentation.** Prediction results and feature maps with different resolutions are generated from ViT-B and ViT-CoMer-B.

5. Conclusion

In this work, we propose ViT-CoMer, a plain, non-hierarchical, and feature-enhanced ViT backbone that effectively leverages the strengths of both CNN and Transformer. Without altering the ViT architecture, we integrate a multi-scale convolutional feature interaction module to reconstruct fine-grained hierarchical semantic features. We validate ViT-CoMer on dense prediction tasks including object detection, instance segmentation, and semantic segmentation. Extensive experiments demonstrate that our approach can achieve superior performance compared to both plain and adapted backbones. Additionally, our approach can easily obtain advanced ViT pre-trained weights and attain comparable, even surpassing performance compared to state-of-the-art backbones.

References

[1] Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. Layer normalization. *arXiv preprint arXiv:1607.06450*, 2016. 4

[2] Hangbo Bao, Li Dong, Songhao Piao, and Furu Wei. Beit: Bert pre-training of image transformers. *arXiv preprint arXiv:2106.08254*, 2021. 2, 3

[3] Alexey Bochkovskiy, Chien Yao Wang, and Hong Yuan Mark Liao. Yolov4: Optimal speed and accuracy of object detection. 2020. 1

[4] Yuxuan Cai, Yizhuang Zhou, Qi Han, Jianjian Sun, Xiangwen Kong, Jun Li, and Xiangyu Zhang. Reversible column networks. In *ICLR*, 2023. 7

[5] Zhaowei Cai and Nuno Vasconcelos. Cascade r-cnn: High quality object detection and instance segmentation. *TPAMI*, 43(5):1483–1498, 2019. 1, 6

[6] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *ECCV*, pages 213–229. Springer, 2020. 1

[7] Kai Chen, Jiaqi Wang, Jiangmiao Pang, Yuhang Cao, Yu Xiong, Xiaoxiao Li, Shuyang Sun, Wansen Feng, Ziwei Liu, Jiarui Xu, et al. Mmdetection: Open mmlab detection toolbox and benchmark. *arXiv preprint arXiv:1906.07155*, 2019. 5

[8] Qiang Chen, Qiman Wu, Jian Wang, Qinghao Hu, Tao Hu, Errui Ding, Jian Cheng, and Jingdong Wang. Mixformer: Mixing features across windows and dimensions. In *CVPR*, pages 5249–5259, 2022. 3, 5

[9] Zhe Chen, Yuchen Duan, Wenhai Wang, Junjun He, Tong Lu, Jifeng Dai, and Yu Qiao. Vision transformer adapter for dense predictions. *ICLR*, 2023. 3, 5, 6, 7

[10] Bowen Cheng, Ishan Misra, Alexander G. Schwing, Alexander Kirillov, and Rohit Girdhar. Masked-attention mask transformer for universal image segmentation. 2022. 7

[11] 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*, 34:9355–9366, 2021. 5, 7

[12] MMSegmentation Contributors. Mmsegmentation: Openmmlab semantic segmentation toolbox and benchmark, 2020. 6

[13] 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. *arXiv preprint arXiv:2010.11929*, 2020. 1, 2

[14] 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*, 2022. 7

[15] Jianyuan Guo, Kai Han, Han Wu, Yehui Tang, Xinghao Chen, Yunhe Wang, and Chang Xu. Cmt: Convolutional neural networks meet vision transformers. In *CVPR*, pages 12175–12185, 2022. 1

[16] Dongchen Han, Xuran Pan, Yizeng Han, Shiji Song, and Gao Huang. Flatten transformer: Vision transformer using focused linear attention. In *ICCV*, 2023. 5

[17] Ali Hassani, Steven Walton, Jiachen Li, Shen Li, and Humphrey Shi. Neighborhood attention transformer. In *CVPR*, pages 6185–6194, 2023. 5

[18] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, pages 770–778, 2016. 1

[19] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. *TPAMI*, 2017. 1

[20] 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*, 2021. 6

[21] Menglin Jia, Luming Tang, Bor-Chun Chen, Claire Cardie, Serge Belongie, Bharath Hariharan, and Ser-Nam Lim. Visual prompt tuning. In *ECCV*, pages 709–727. Springer, 2022. 3

[22] Youngwan Lee, Jonghee Kim, Jeffrey Willette, and Sung Ju Hwang. Mpvit: Multi-path vision transformer for dense prediction. In *CVPR*, pages 7287–7296, 2022. 1

[23] Feng Li, Hao Zhang, Huaizhe xu, Shilong Liu, Lei Zhang, Lionel M. Ni, and Heung-Yeung Shum. Mask dino: Towards a unified transformer-based framework for object detection and segmentation, 2022. 7

[24] Kunchang Li, Yali Wang, Peng Gao, Guanglu Song, Yu Liu, Hongsheng Li, and Yu Qiao. Uniformer: Unified transformer for efficient spatiotemporal representation learning. *arXiv preprint arXiv:2201.04676*, 2022. 3

[25] 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*, 2021. 5, 6, 7

[26] Yanghao Li, Hanzi Mao, Ross Girshick, and Kaiming He. Exploring plain vision transformer backbones for object detection. In *ECCV*, pages 280–296. Springer, 2022. 1, 2, 5, 6

[27] 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 *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13*, pages 740–755. Springer, 2014. 5

[28] 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*, pages 10012–10022, 2021. 1, 3, 5, 6, 7

[29] Zhuang Liu, Hanzi Mao, Chao-Yuan Wu, Christoph Feichtenhofer, Trevor Darrell, and Saining Xie. A convnet for the 2020s. In *CVPR*, pages 11976–11986, 2022. 1, 5

[30] Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, et al. Dinov2: Learning robust visual features without supervision. *arXiv preprint arXiv:2304.07193*, 2023. 2, 5, 6

[31] 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, 3, 6, 7

[32] Joseph Redmon and Ali Farhadi. Yolo9000: Better, faster, stronger. In *CVPR*, pages 6517–6525, 2017. 1

[33] Joseph Redmon and Ali Farhadi. Yolov3: An incremental improvement. *arXiv*, 2018.

[34] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. In *CVPR*, pages 779–788, 2016. 1

[35] Yifeng Shi, Feng Lv, Xinliang Wang, Chunlong Xia, Shaojie Li, Shujie Yang, Teng Xi, and Gang Zhang. Opentransmind: A new baseline and benchmark for 1st foundation model challenge of intelligent transportation. In *CVPR*, pages 6327–6334, 2023. 1

[36] Chenyang Si, Weihao Yu, Pan Zhou, Yichen Zhou, Xinchao Wang, and Shuicheng Yan. Inception transformer. *NeurIPS*, 35:23495–23509, 2022. 3, 8

[37] 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

[38] 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. 7

[39] Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Hervé Jégou. Training data-efficient image transformers & distillation through attention. In *ICML*, pages 10347–10357. PMLR, 2021. 6

[40] 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*, pages 568–578, 2021. 1, 3, 5

[41] Wenhai Wang, Enze Xie, Xiang Li, Deng-Ping Fan, Kaitao Song, Ding Liang, Tong Lu, Ping Luo, and Ling Shao. Pvt v2: Improved baselines with pyramid vision transformer. *Computational Visual Media*, 8(3):415–424, 2022. 1, 5, 6, 7

[42] Wenhui Wang, Hangbo Bao, Li Dong, Johan Bjorck, Zhiliang Peng, Qiang Liu, Kriti Aggarwal, Owais Khan Mohammed, Saksham Singhal, Subhojit Som, and Furu Wei. Image as a foreign language: BEiT pretraining for vision and vision-language tasks. In *CVPR*, 2023. 2, 3, 7

[43] Wenhai Wang, Jifeng Dai, Zhe Chen, Zhenhang Huang, Zhiqi Li, Xizhou Zhu, Xiaowei Hu, Tong Lu, Lewei Lu, Hongsheng Li, et al. Internimage: Exploring large-scale vision foundation models with deformable convolutions. In *CVPR*, pages 14408–14419, 2023. 5, 6

[44] Tete Xiao, Yingcheng Liu, Bolei Zhou, Yuning Jiang, and Jian Sun. Unified perceptual parsing for scene understanding. In *ECCV*, pages 418–434, 2018. 6

[45] 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. 5, 6

[46] Xiaoqing Ye, Mao Shu, Hanyu Li, Yifeng Shi, Yingying Li, Guangjie Wang, Xiao Tan, and Errui Ding. Rope3d: The roadside perception dataset for autonomous driving and monocular 3d object detection task. In *CVPR*, pages 21341–21350, 2022. 1

[47] Dongshuo Yin, Yiran Yang, Zhechao Wang, Hongfeng Yu, Kaiwen Wei, and Xian Sun. 1% vs 100%: Parameter-efficient low rank adapter for dense predictions. In *CVPR*, pages 20116–20126, 2023. 3

[48] Haibao Yu, Yizhen Luo, Mao Shu, Yiyi Huo, Zebang Yang, Yifeng Shi, Zhenglong Guo, Hanyu Li, Xing Hu, Jirui Yuan, et al. Dair-v2x: A large-scale dataset for vehicle-infrastructure cooperative 3d object detection. In *CVPR*, pages 21361–21370, 2022. 1

[49] Haibao Yu, Wenxian Yang, Hongzhi Ruan, Zhenwei Yang, Yingjuan Tang, Xu Gao, Xin Hao, Yifeng Shi, Yifeng Pan, Ning Sun, et al. V2x-seq: A large-scale sequential dataset for vehicle-infrastructure cooperative perception and forecasting. In *CVPR*, pages 5486–5495, 2023. 1

[50] Weihao Yu, Mi Luo, Pan Zhou, Chenyang Si, Yichen Zhou, Xinchao Wang, Jiashi Feng, and Shuicheng Yan. Metaformer is actually what you need for vision. In *CVPR*, pages 10819–10829, 2022. 3

[51] Guhnoo Yun, Juhan Yoo, Kijung Kim, Jeongho Lee, and Dong Hwan Kim. Spanet: Frequency-balancing token mixer using spectral pooling aggregation modulation. In *ICCV*, pages 6113–6124, 2023. 5

[52] Hao Zhang, Feng Li, Shilong Liu, Lei Zhang, Hang Su, Jun Zhu, Lionel M. Ni, and Heung-Yeung Shum. Dino: Detr with improved denoising anchor boxes for end-to-end object detection, 2022. 6

[53] Bolei Zhou, Hang Zhao, Xavier Puig, Tete Xiao, Sanja Fidler, Adela Barriuso, and Antonio Torralba. Semantic understanding of scenes through the ade20k dataset. *IJCV*, 127: 302–321, 2019. 5

[54] Xizhou Zhu, Weijie Su, Lewei Lu, Bin Li, Xiaogang Wang, and Jifeng Dai. Deformable detr: Deformable transformers for end-to-end object detection. *arXiv preprint arXiv:2010.04159*, 2020. 4

[55] Xizhou Zhu, Jinguo Zhu, Hao Li, Xiaoshi Wu, Hongsheng Li, Xiaohua Wang, and Jifeng Dai. Uni-perceiver: Pre-training unified architecture for generic perception for zero-shot and few-shot tasks. In *CVPR*, pages 16804–16815, 2022. 7

[56] Zhuofan Zong, Guanglu Song, and Yu Liu. Detrs with collaborative hybrid assignments training, 2022. 6