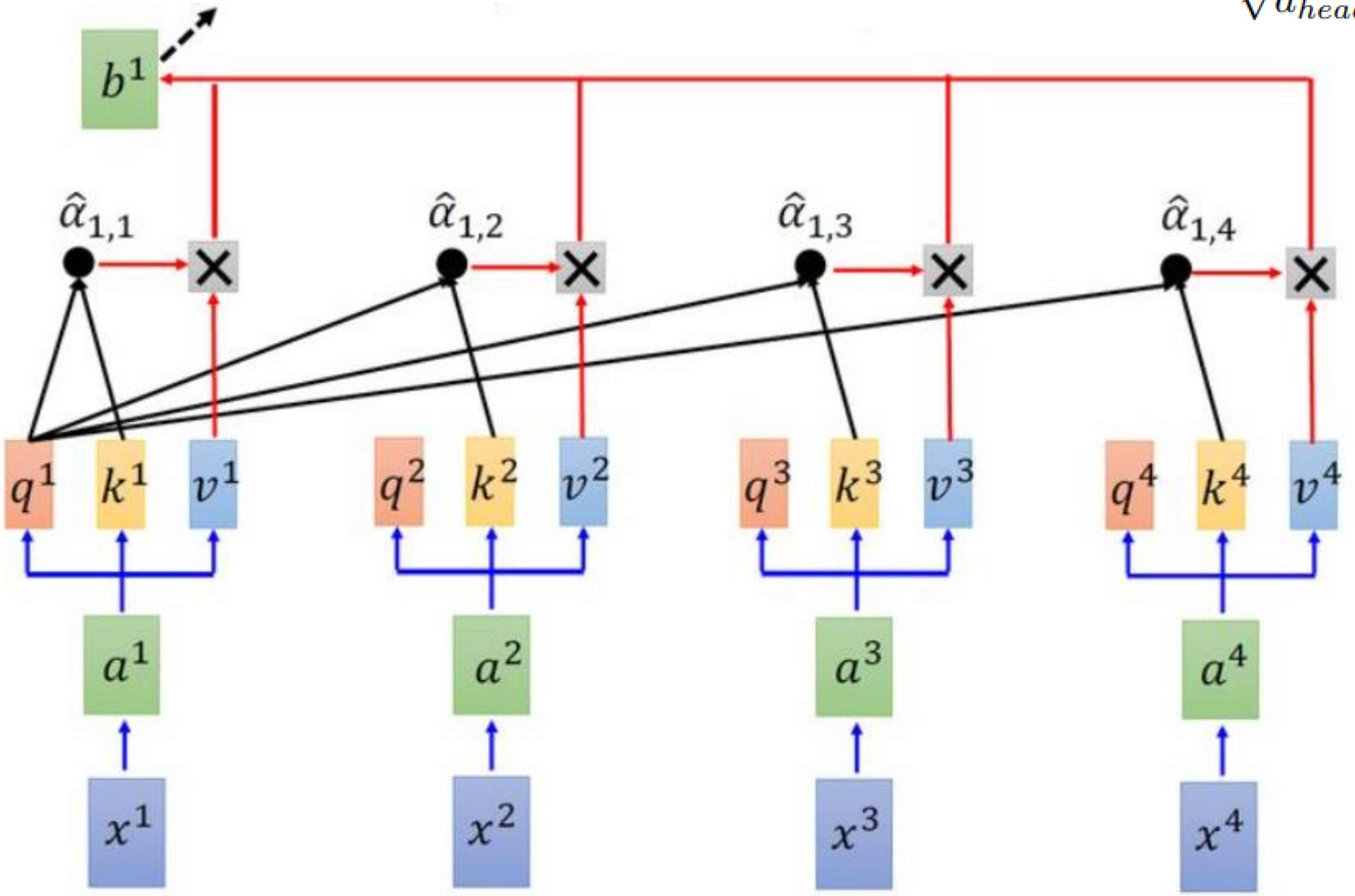
SegFormer: Simple and Efficient Design for Semantic Segmentation with Transformers

Enze Xie[1]* **Wenhai Wang**[2] **Zhiding Yu**[3] **Anima Anandkumar**[3,4] **Jose M. Alvarez**[3] **Ping Luo**[1]

[1]The University of Hong Kong [2]Nanjing University [3]NVIDIA [4]Caltech

Self Attention

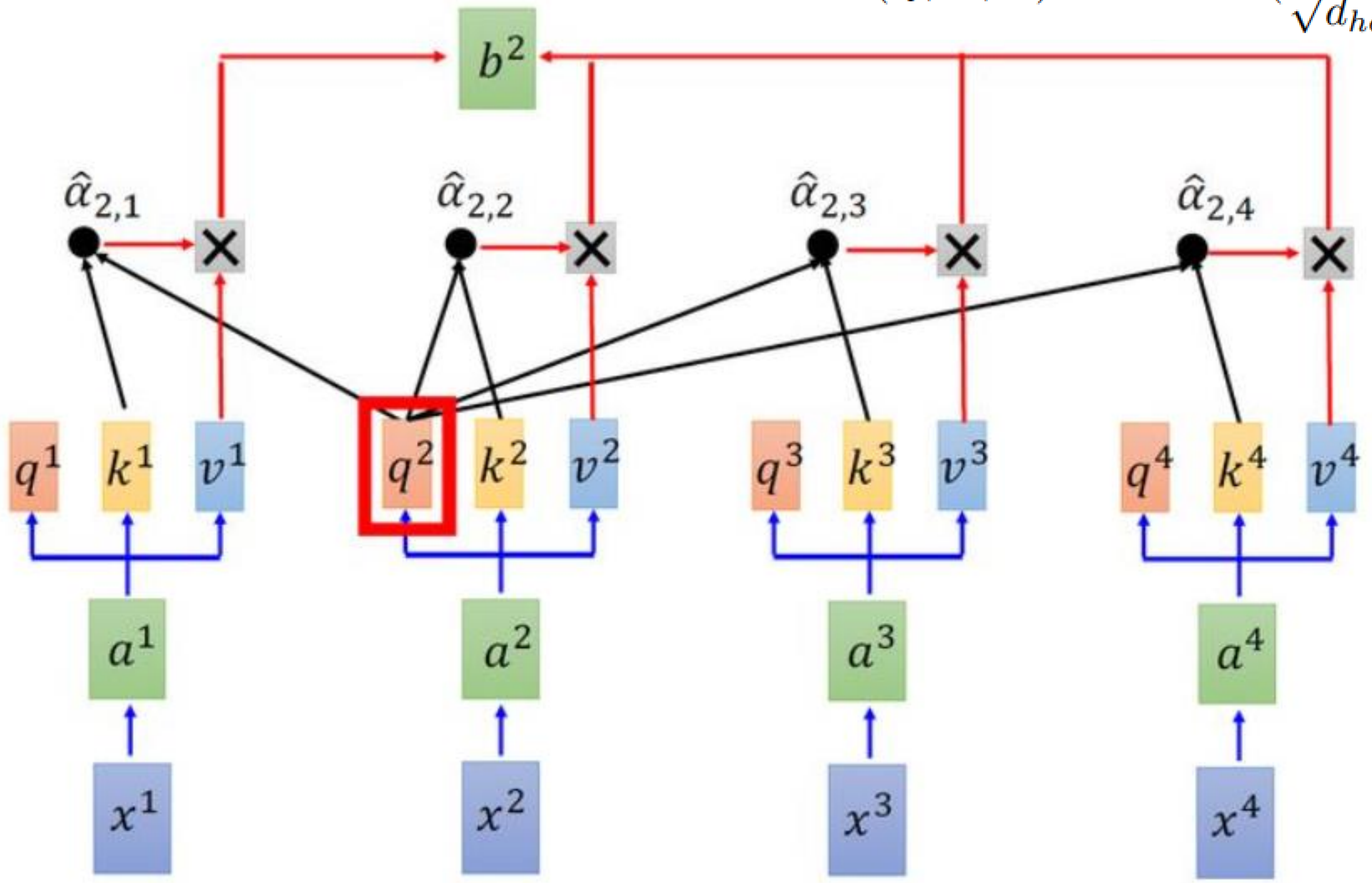
$$\text{Attention}(Q, K, V) = \text{Softmax}(\frac{QK^{\top}}{\sqrt{d_{head}}})V$$



Input:
Sequence($\boldsymbol{x_i}$)
Output:
Sequence($\boldsymbol{b_i}$)
$\boldsymbol{Q}$:Query
$\boldsymbol{K}$:Key
$\boldsymbol{V}$:Value

Self Attention

$$\text{Attention}(Q,K,V) = \text{Softmax}\left(\frac{QK^{\top}}{\sqrt{d_{head}}}\right)V$$



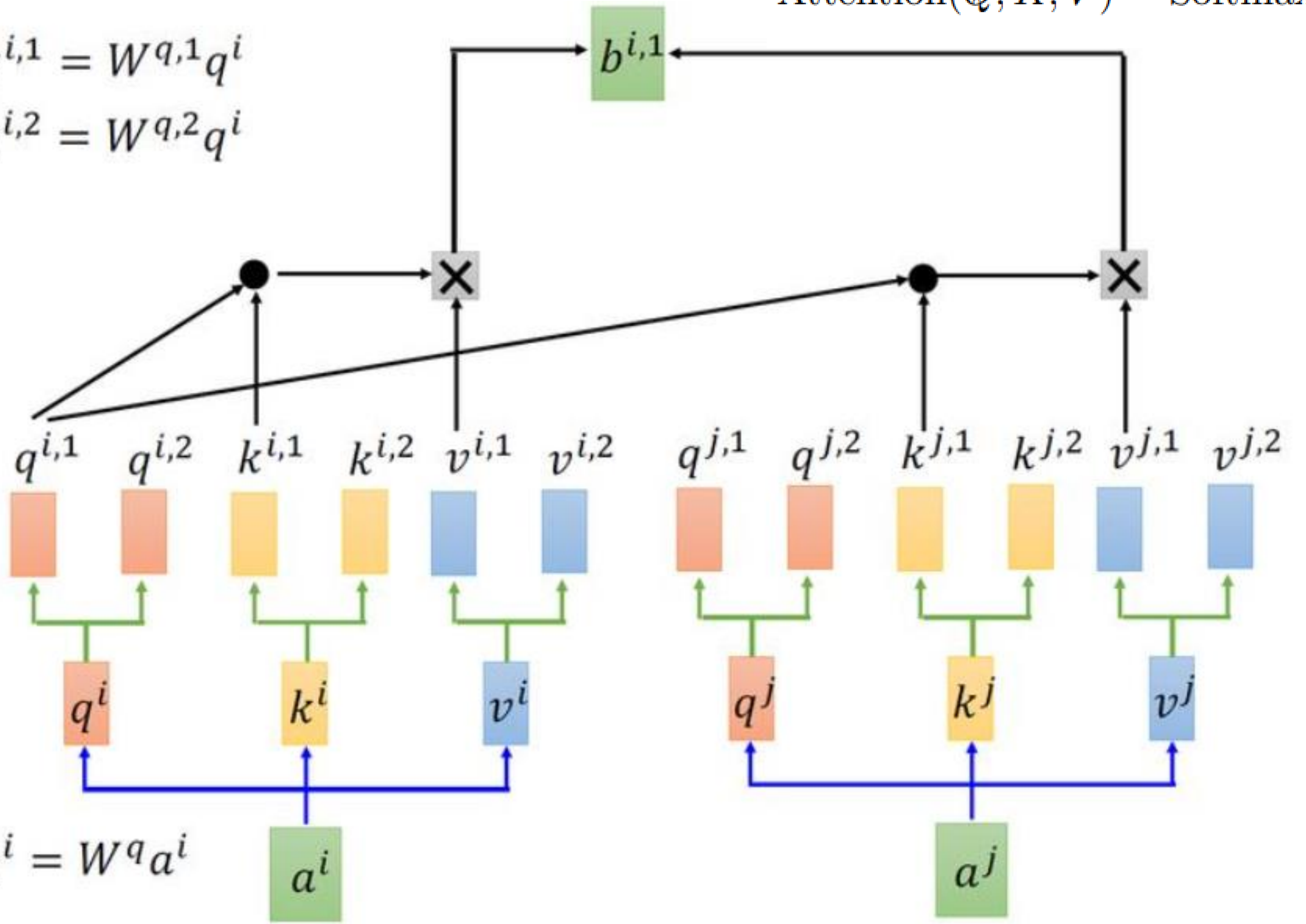
Input:
Sequence($\boldsymbol{a_i}$)
Output:
Sequence($\boldsymbol{b_i}$)
$\boldsymbol{Q}$:Query
$\boldsymbol{K}$:Key
$\boldsymbol{V}$:Value

Multi-head Self Attention

$$\text{Attention}(Q, K, V) = \text{Softmax}(\frac{QK^\top}{\sqrt{d_{head}}})V$$

$q^{i,1} = W^{q,1}q^i$

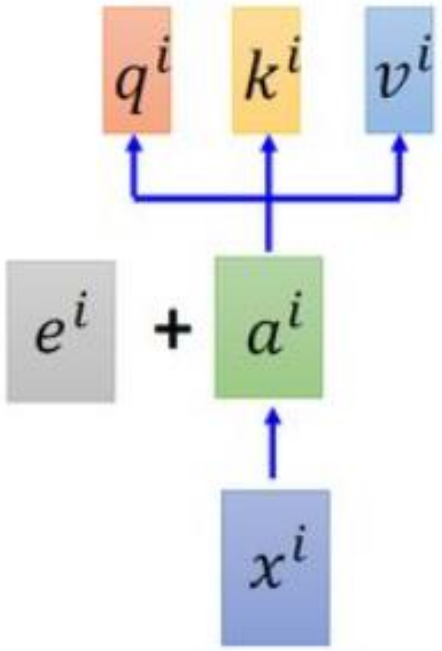
$q^{i,2} = W^{q,2}q^i$



$q^i = W^q a^i$

Input: Sequence($\boldsymbol{a_i}$)
Output: Sequence($\boldsymbol{b_i}$)
$\boldsymbol{Q}$:Query
$\boldsymbol{K}$:Key
$\boldsymbol{V}$:Value

Position Encoding

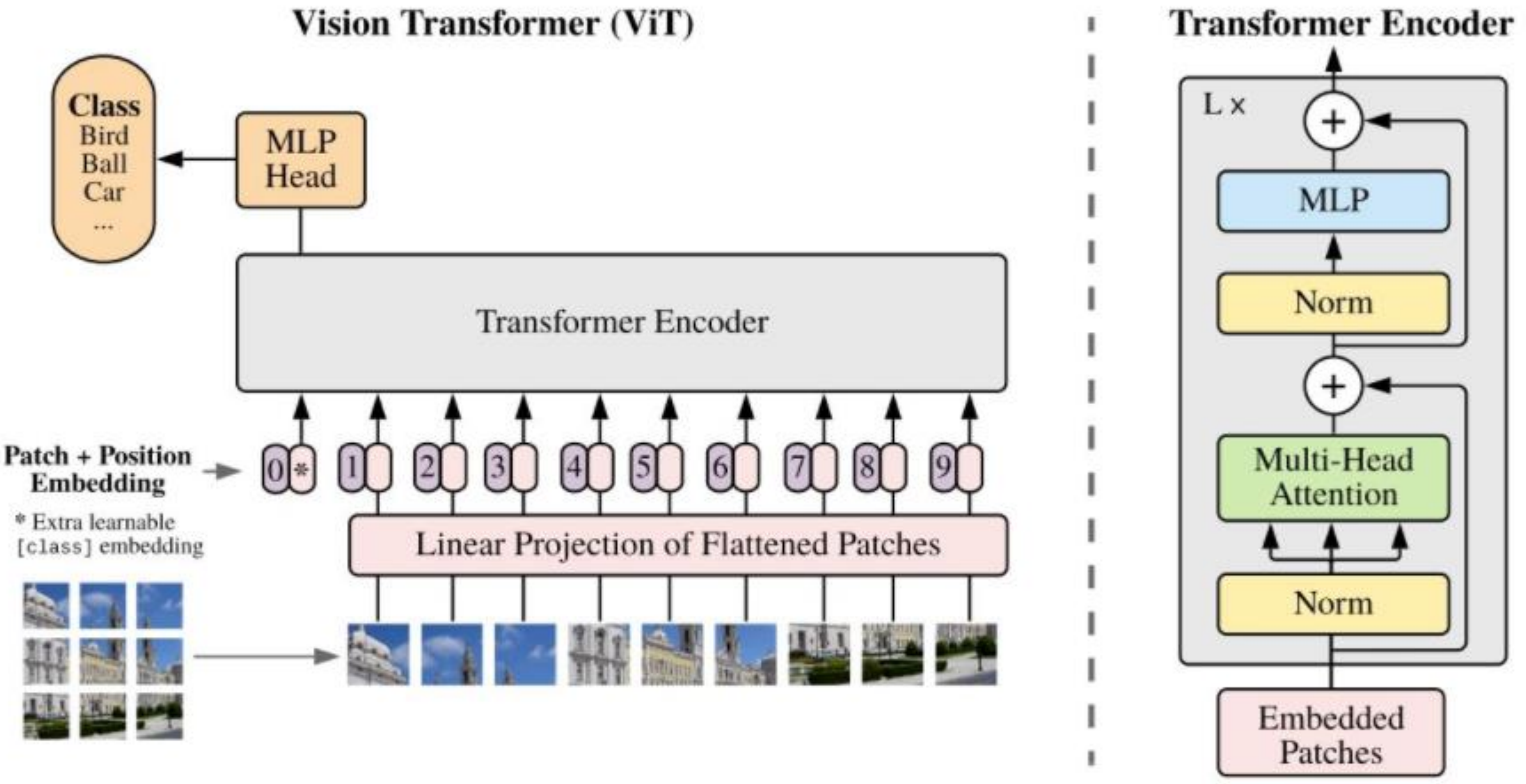


Add **Position information** to Self-attention

Original Paper: an manual-designed position encoding ,added with input embedding

Position Encoding can be learnt from data

ViT & others



Input:
C*H*W -> N * (P^2*C)

Linear Projection
N * (P^2*C) -> N * 512

Position Encoding
Learnable

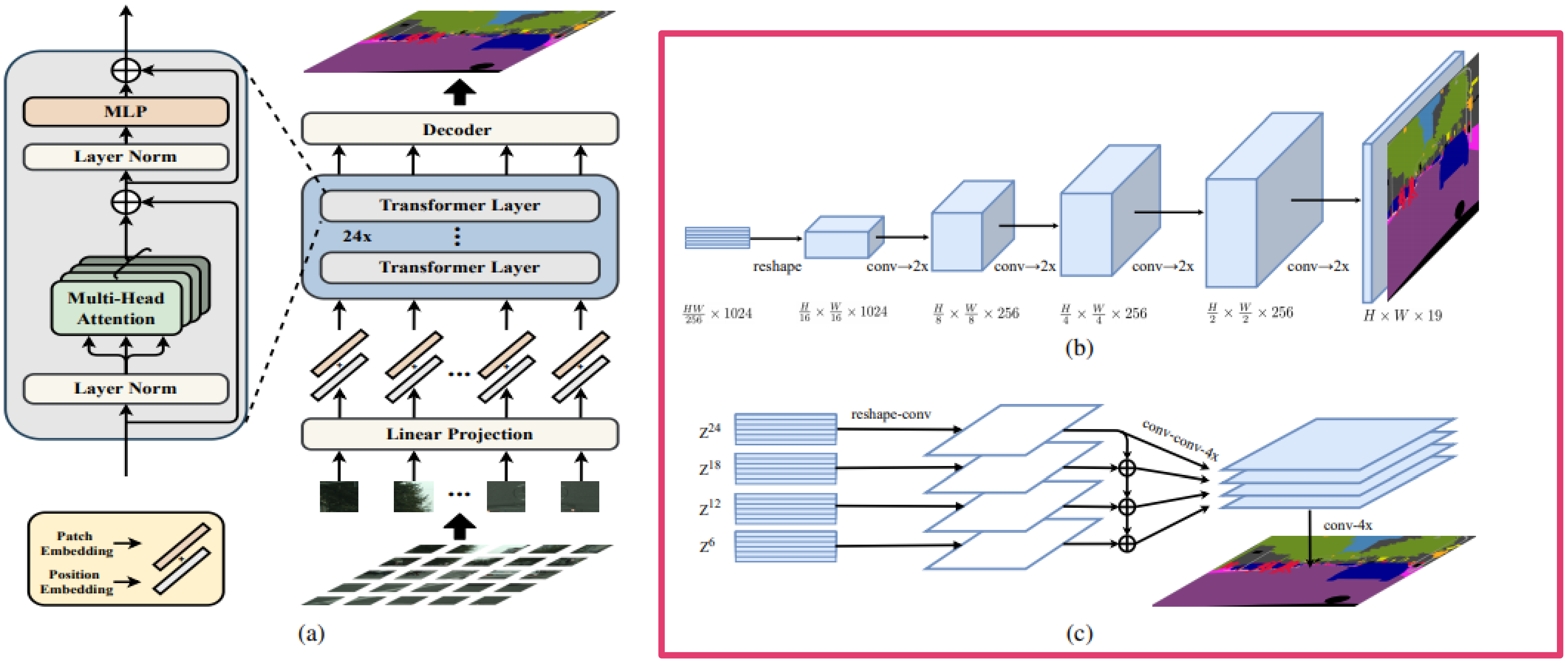
MLP head:
Process on the **learnable embedding(*)**

An Image is Worth 16x16 Words:Transformers for Image Recognition at Scale

ViT & others

SETR

ViT + decoder



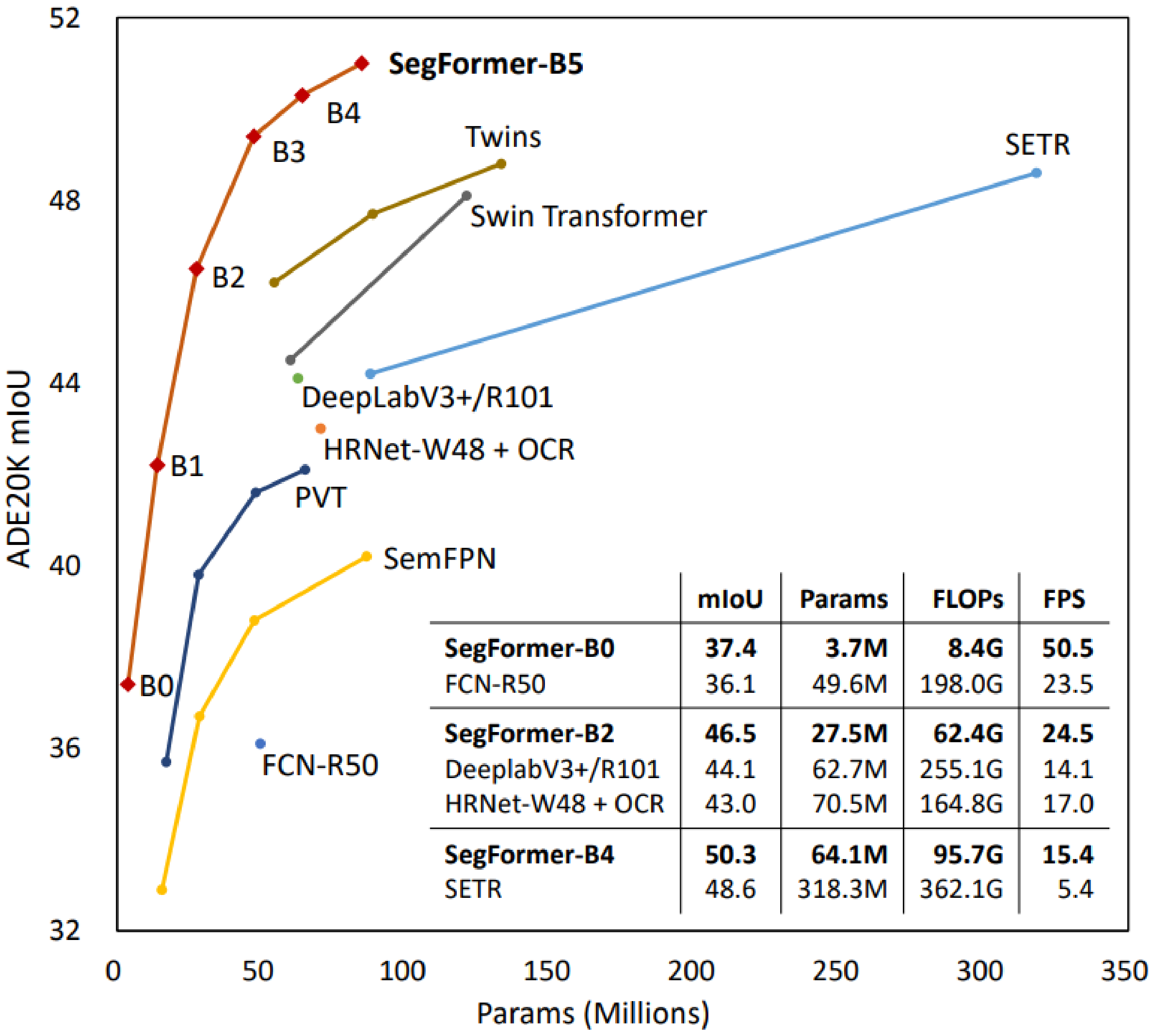
Rethinking Semantic Segmentation from a Sequence-to-Sequence Perspective with Transformers

Segformer

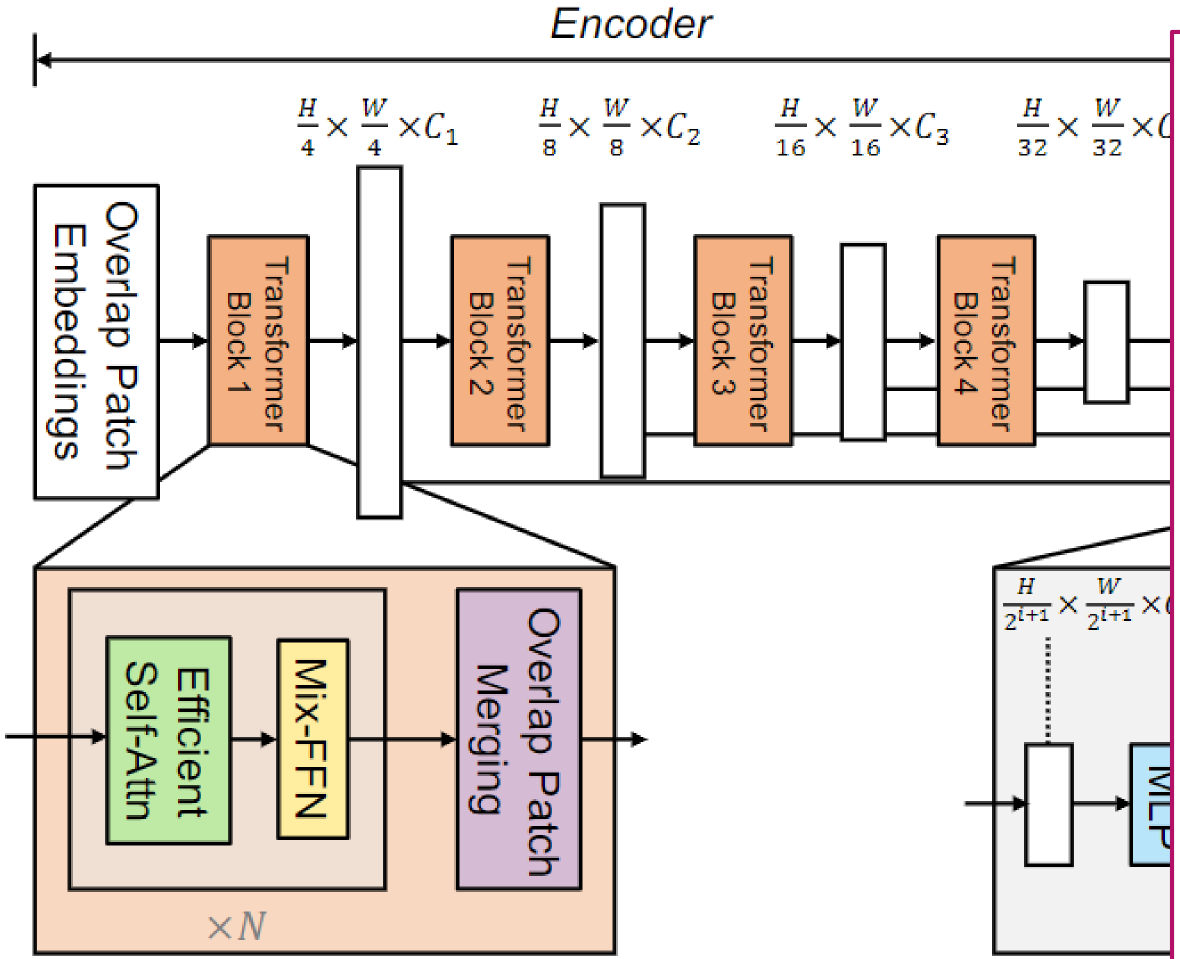
1.Contains a novel hierarchically structured Transformer encoder which outputs multiscale features

2. **MLP decoder** aggregates features from different layers and merge global &local information

3.Without Position encoding



	mIoU	Params	FLOPs	FPS
SegFormer-B0	**37.4**	**3.7M**	**8.4G**	**50.5**
FCN-R50	36.1	49.6M	198.0G	23.5
SegFormer-B2	**46.5**	**27.5M**	**62.4G**	**24.5**
DeeplabV3+/R101	44.1	62.7M	255.1G	14.1
HRNet-W48 + OCR	43.0	70.5M	164.8G	17.0
SegFormer-B4	**50.3**	**64.1M**	**95.7G**	**15.4**
SETR	48.6	318.3M	362.1G	5.4

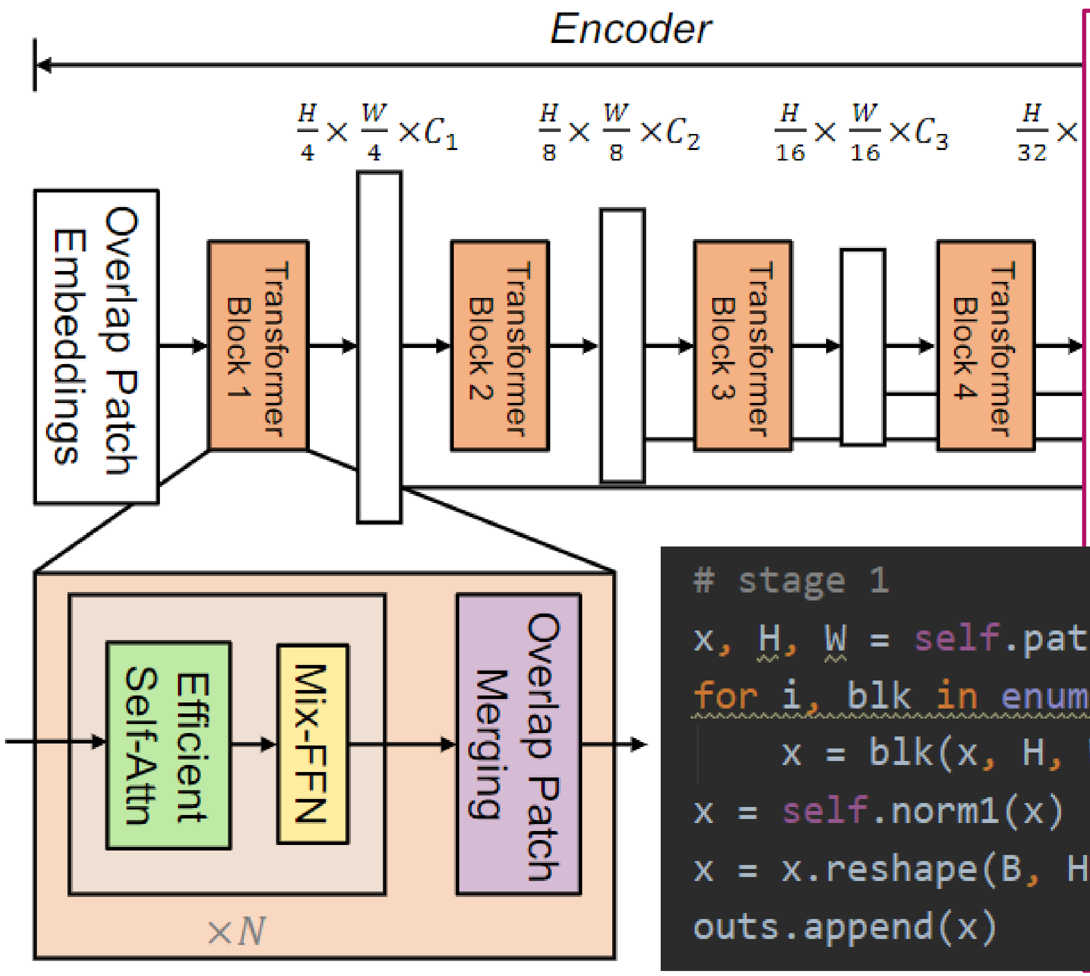


Mix-FFN:
Linear + 3*3 Conv
(substitute Position Encoding
<from CVPT[1]>)

$$\mathbf{x}_{out} = \text{MLP}(\text{GELU}(\text{Conv}_{3\times3}(\text{MLP}(\mathbf{x}_{in})))) + \mathbf{x}_{in}$$

Figure 2: **The proposed SegFormer framework** consists of two main modules: A hierarchical Transformer encoder to extract coarse and fine features; and a lightweight All-MLP decoder to directly fuse these multi-level features and predict the semantic segmentation mask. “FFN” indicates feed-forward network.

[1]Conditional positional encodings for vision transformers



Patch embedding/Merging:

·Change size

·Overlapped

```
# stage 1
x, H, W = self.patch_embed1(x)
for i, blk in enumerate(self.block1):
    x = blk(x, H, W)
x = self.norm1(x)
x = x.reshape(B, H, W, -1).permute(0, 3, 1, 2).contiguous()
outs.append(x)
```

Patch embedding/Merging:

Attention & FFN:

Figure 2: **The proposed SegFormer framework** consists of two main modules: A hierarchical Transformer encoder to extract coarse and fine features; and a lightweight All-MLP decoder to directly fuse these multi-level features and predict the semantic segmentation mask. "FFN" indicates feed-forward network.

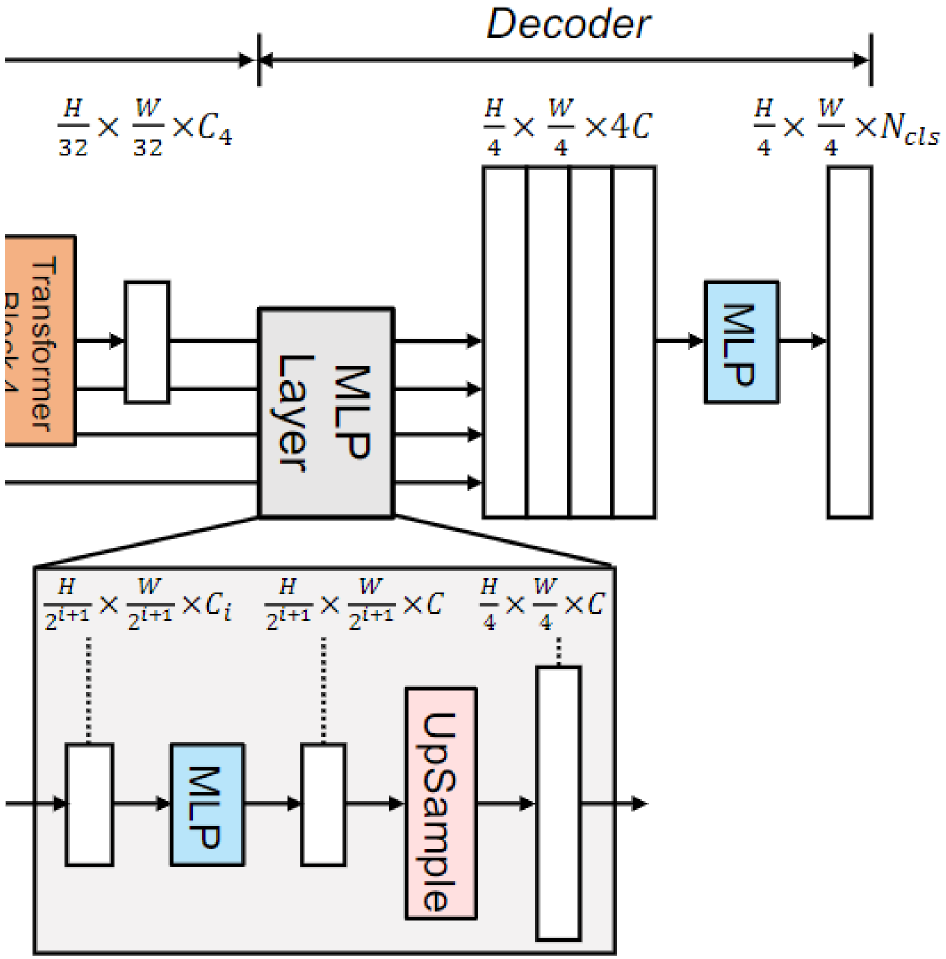


Figure 2: **The proposed SegFormer framework** consists of two main modules: A hierarchical Transformer encoder to extract coarse and fine features; and a lightweight All-MLP decoder to directly fuse these multi-level features and predict the semantic segmentation mask. "FFN" indicates feed-forward network.

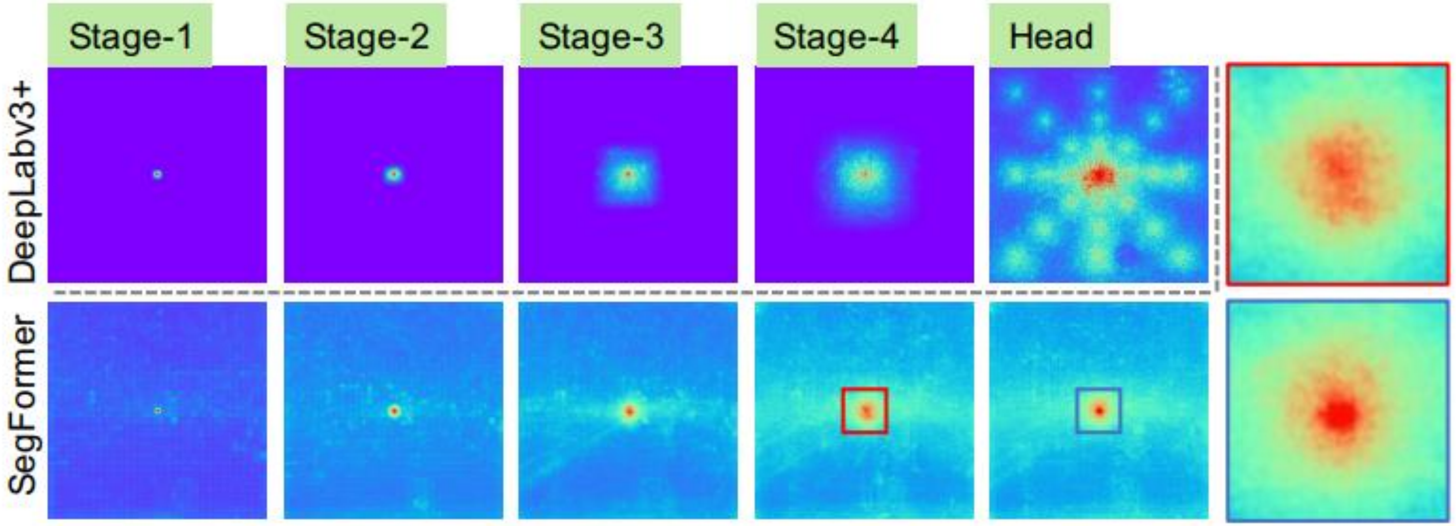


Figure 3: **Effective Receptive Field (ERF) on Cityscapes** (average over 100 images). Top row: Deeplabv3+. Bottom row: SegFormer. ERFs of the four stages and the decoder heads of both architectures are visualized. Best viewed with zoom in.

(c) Mix-FFN vs. positional encoding (PE) for different test resolution on Cityscapes.

Inf Res	Enc Type	mIoU ↑
768×768	PE	77.3
1024×2048	PE	74.0
768×768	Mix-FFN	80.5
1024×2048	Mix-FFN	79.8

	Method	Encoder	Params ↓	ADE20K			Cityscapes		
				Flops ↓	FPS ↑	mIoU ↑	Flops ↓	FPS ↑	mIoU ↑
Real-Time	FCN [1]	MobileNetV2	9.8	39.6	64.4	19.7	317.1	14.2	61.5
	ICNet [11]	-	-	-	-	-	-	30.3	67.7
	PSPNet [17]	MobileNetV2	13.7	52.9	57.7	29.6	423.4	11.2	70.2
	DeepLabV3+ [20]	MobileNetV2	15.4	69.4	43.1	34.0	555.4	8.4	75.2
	SegFormer (Ours)	MiT-B0	**3.8**	**8.4**	**50.5**	**37.4**	125.5	15.2	**76.2**
				-	-	-	51.7	26.3	75.3
				-	-	-	31.5	37.1	73.7
				-	-	-	**17.7**	**47.6**	71.9
Non Real-Time	FCN [1]	ResNet-101	68.6	275.7	14.8	41.4	2203.3	1.2	76.6
	EncNet [24]	ResNet-101	**55.1**	218.8	14.9	44.7	1748.0	1.3	76.9
	PSPNet [17]	ResNet-101	68.1	256.4	15.3	44.4	2048.9	1.2	78.5
	CCNet [41]	ResNet-101	68.9	278.4	14.1	45.2	2224.8	1.0	80.2
	DeeplabV3+ [20]	ResNet-101	62.7	255.1	14.1	44.1	2032.3	1.2	80.9
	OCRNet [23]	HRNet-W48	70.5	164.8	**17.0**	45.6	1296.8	**4.2**	81.1
	GSCNN [35]	WideResNet38	-	-	-	-	-	-	80.8
	Axial-DeepLab [74]	AxialResNet-XL	-	-	-	-	2446.8	-	81.1
	Dynamic Routing [75]	Dynamic-L33-PSP	-	-	-	-	**270.0**	-	80.7
	Auto-Deeplab [50]	NAS-F48-ASPP	-	-	-	44.0	695.0	-	80.3
	SETR [7]	ViT-Large	318.3	-	5.4	50.2	-	0.5	82.2
	SegFormer (Ours)	MiT-B4	64.1	**95.7**	15.4	51.1	1240.6	3.0	83.8
	SegFormer (Ours)	MiT-B5	84.7	183.3	9.8	**51.8**	1447.6	2.5	**84.0**