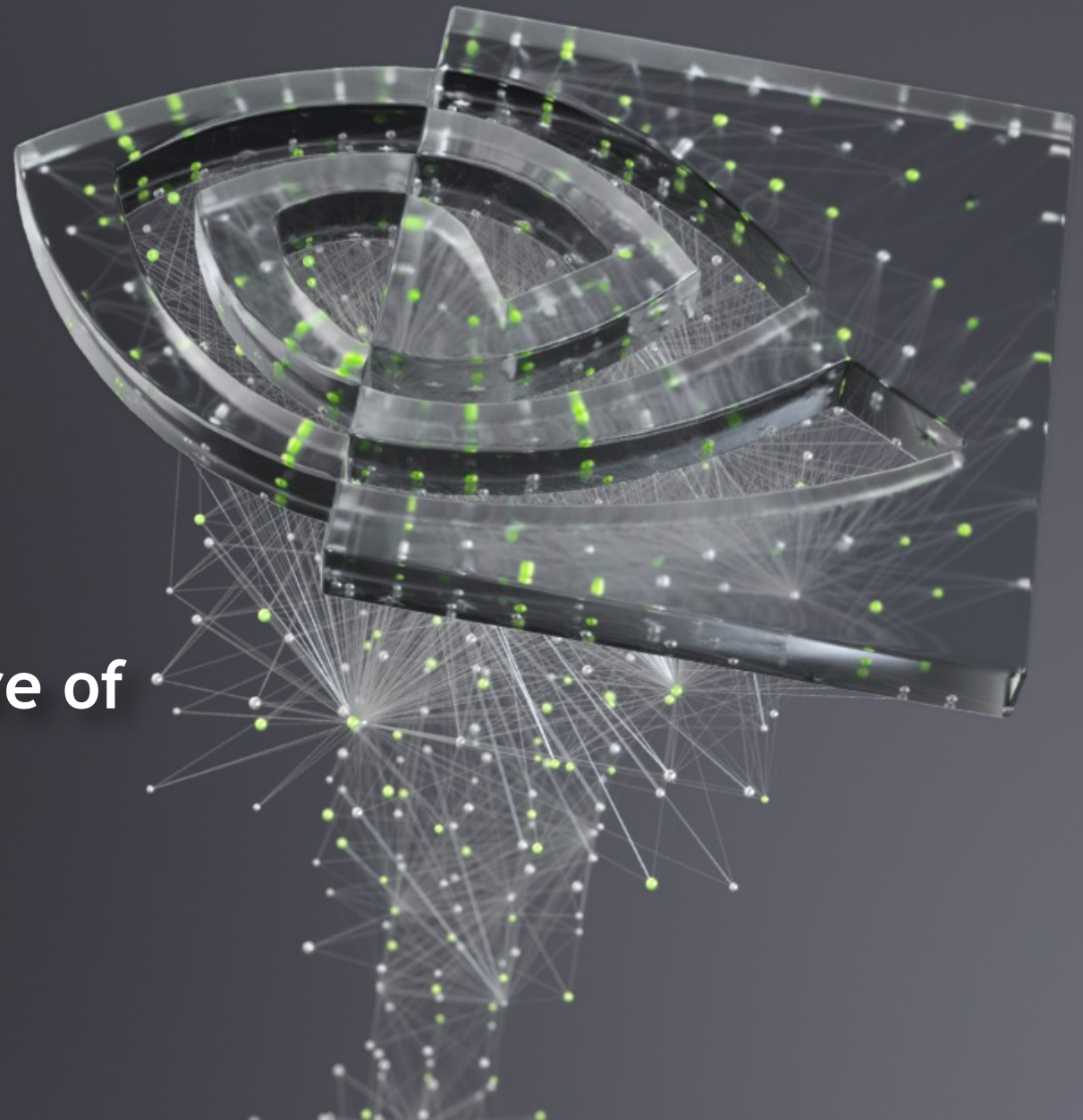
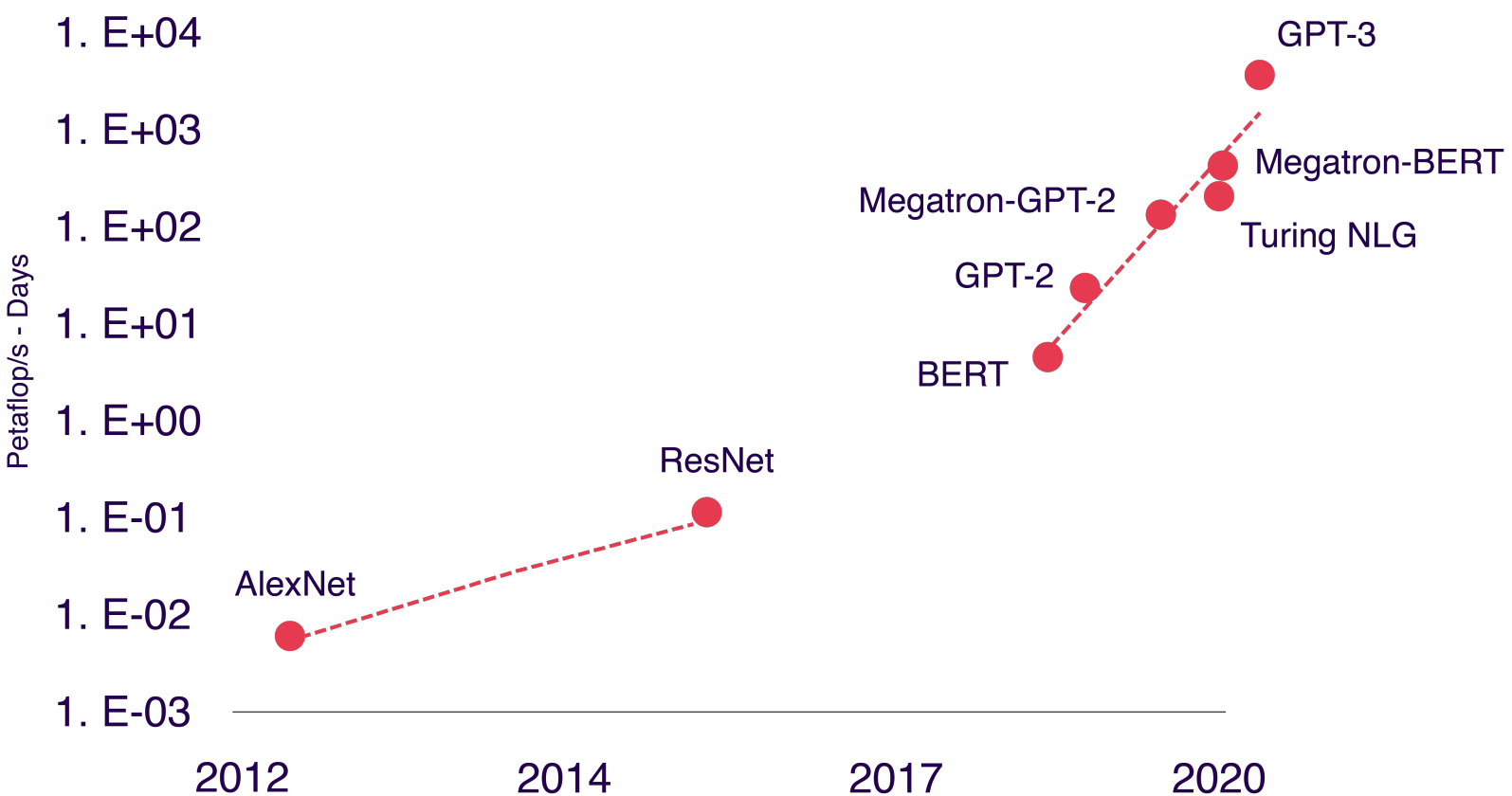


Are transformers the future of vision?

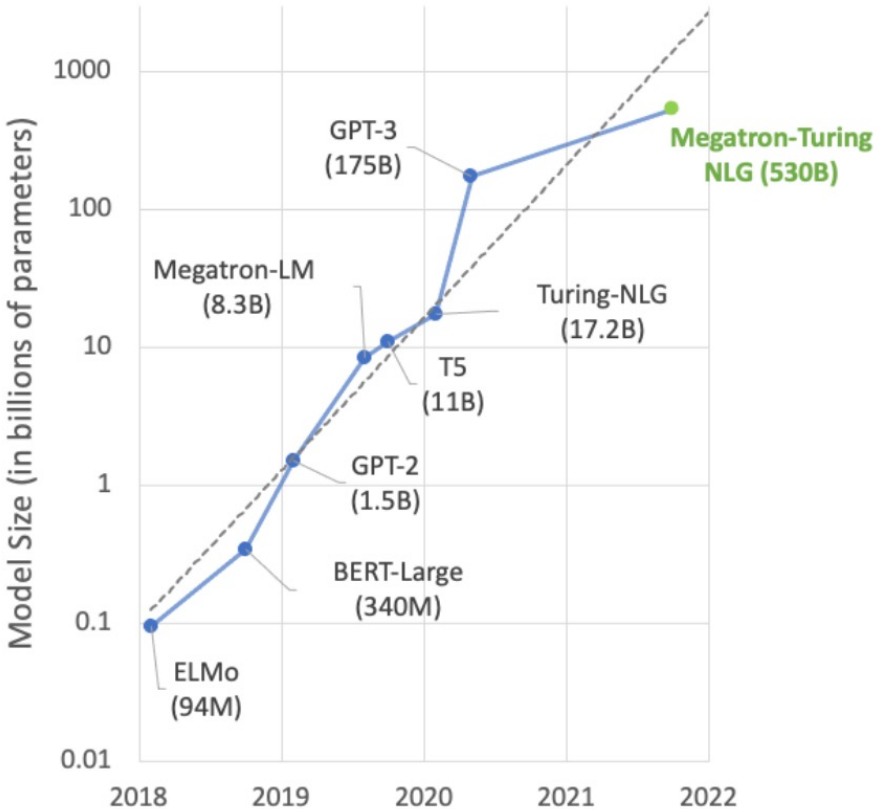
Anima Anandkumar

EXPLODING MODEL COMPLEXITY

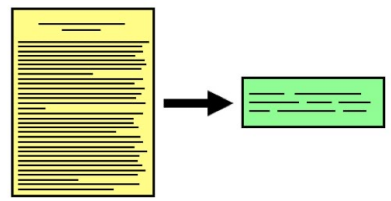
30,000X in 5 years. Doubling every 2 months



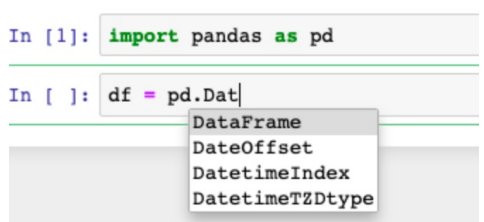
NVIDIA Announced the World's Largest and Most Powerful Generative Language Model



Translation



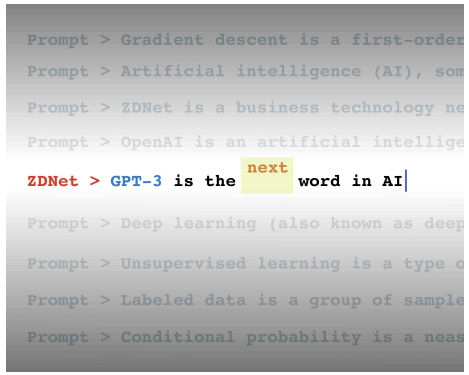
Summarization



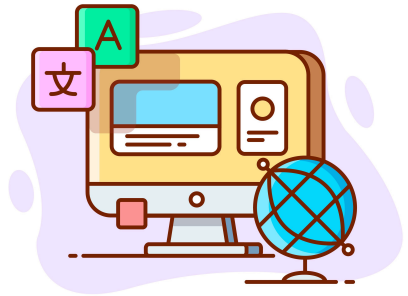
Autocompletion

https://developer.nvidia.com/blog/using-deepspeed-and-megatron-to-train-megatron-turing-nlg-530b-the-worlds-largest-and-most-powerful-generative-language-model/

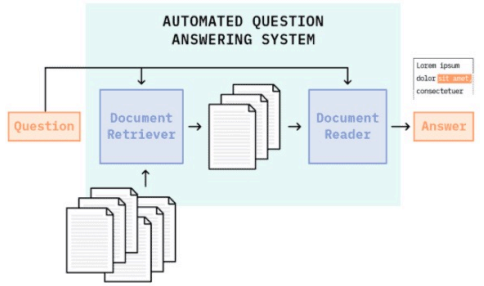
TRANSFORMER IN NLP AND VISION



Language Modeling

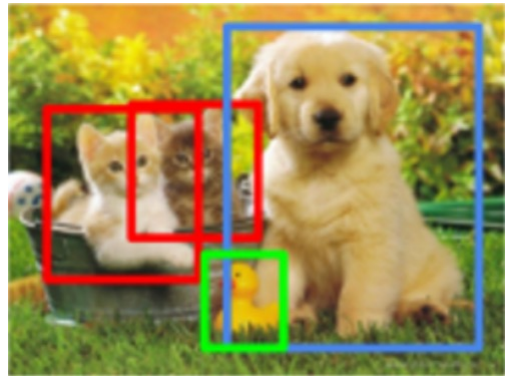


Machine Translation



Question Answering

Image Classification



Detection

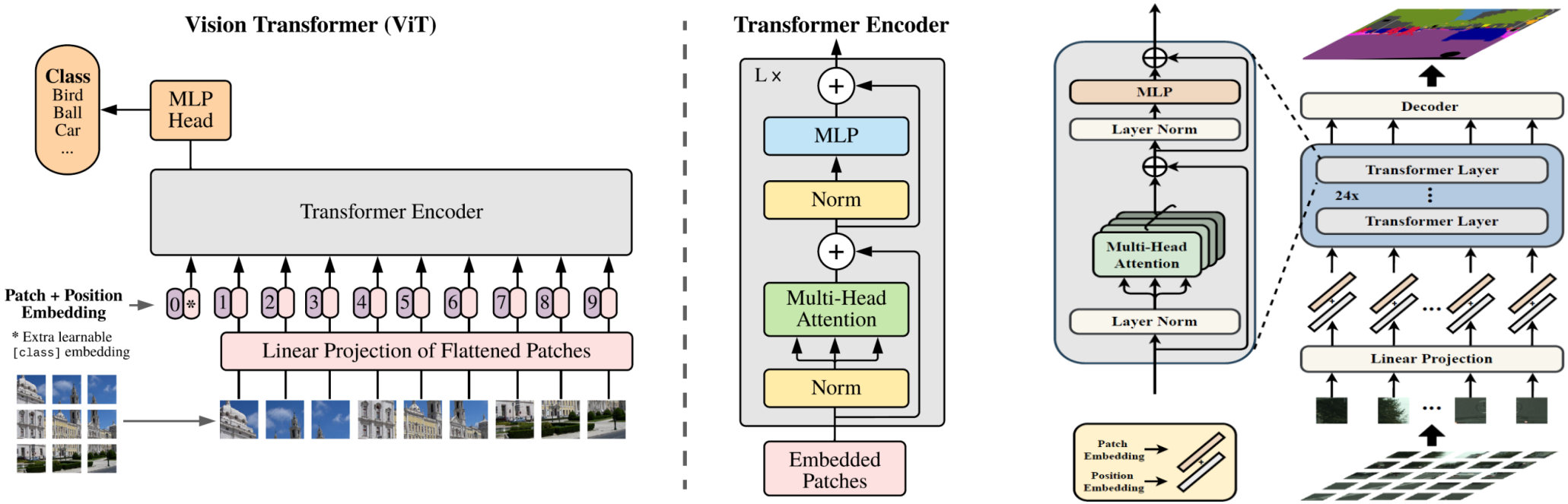
Segmentation

SegFormer: Simple and Efficient Design for Semantic Segmentation with Transformers

Enze Xie, Wenhai Wang, Zhiding Yu, Anima Anandkumar, Jose M. Alvarez, Ping Luo

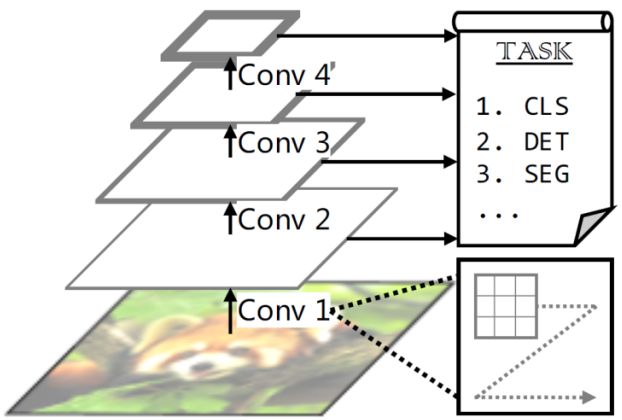
The University of Hong Kong Nanjing University NVIDIA Caltech

Vision Transformer (ViT)

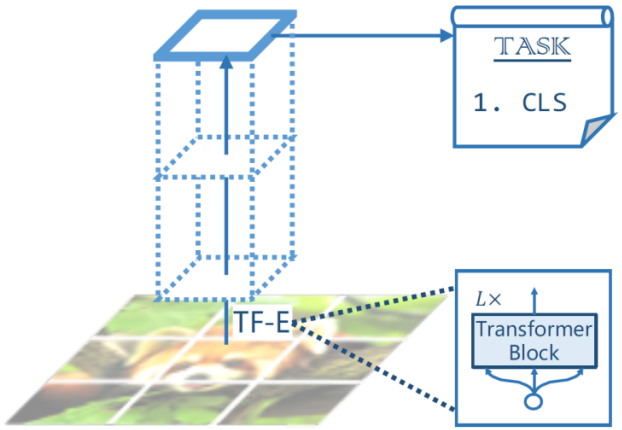


Dosovitskiy et al., **An image is worth 16x16 words: Transformers for image recognition at scale**, ICLR21.
Zheng et al., **Rethinking semantic segmentation from a sequence-to-sequence perspective with transformers**, CVPR21.

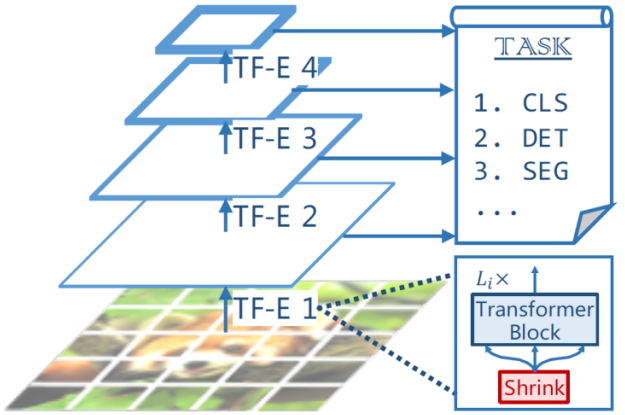
Pyramid Vision Transformer (PVT)



(a) CNNs: VGG [41], ResNet [15], *etc.*



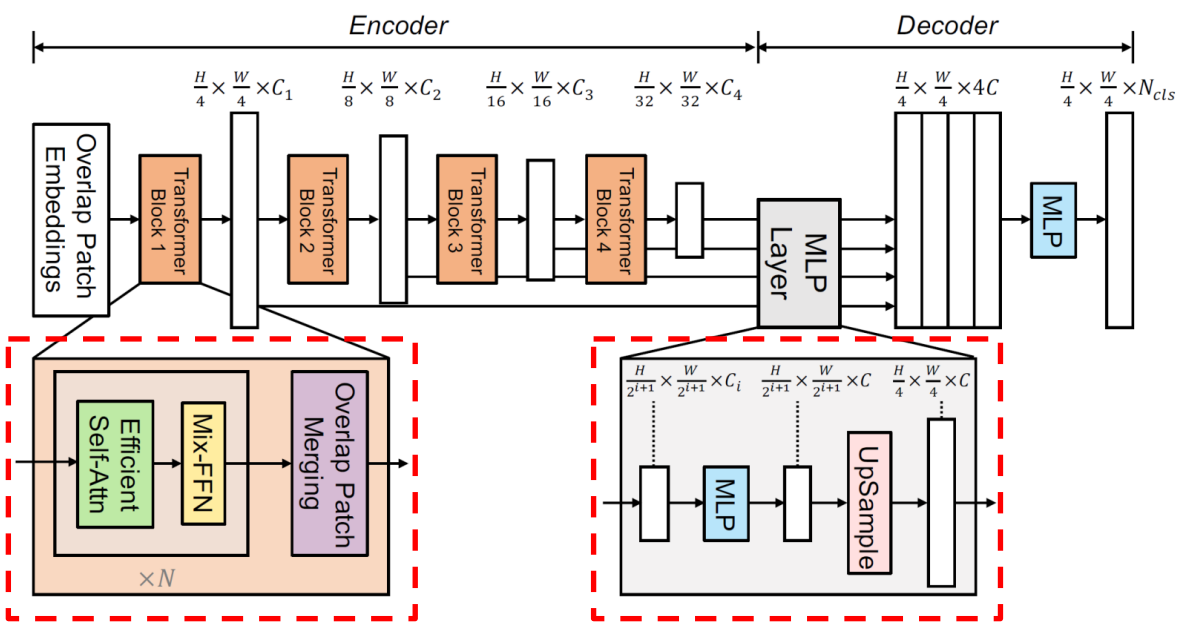
(b) Vision Transformer [10]



(c) Pyramid Vision Transformer (ours)

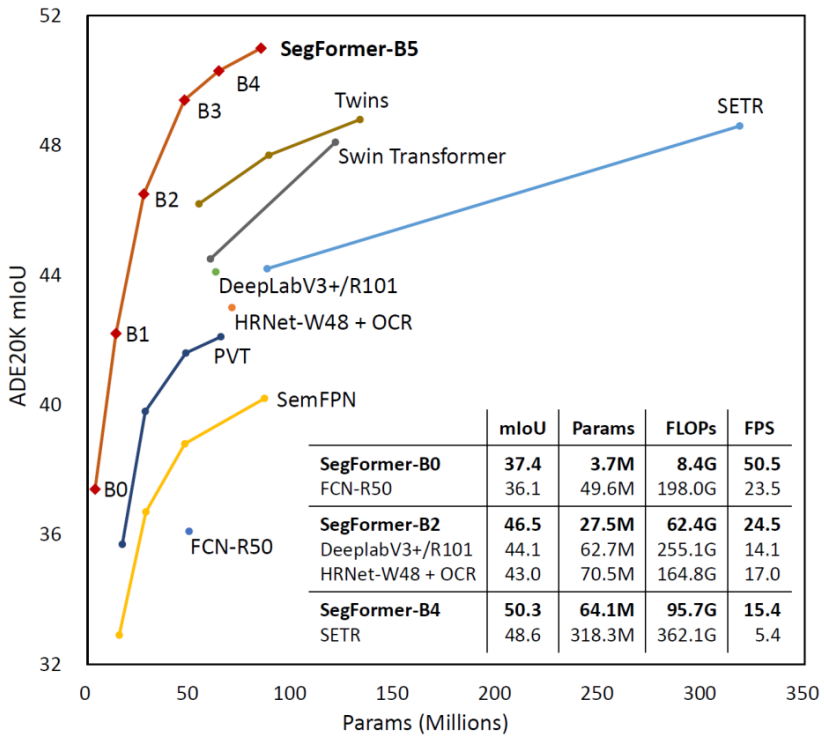
Wang et al., **Pyramid Vision Transformer: A Versatile Backbone for Dense Prediction without Convolutions**, arXiv21.

SegFormer: Architecture Overview



Positional encoding free, pyramid structured encoder

Lightweight all-MLP decoder

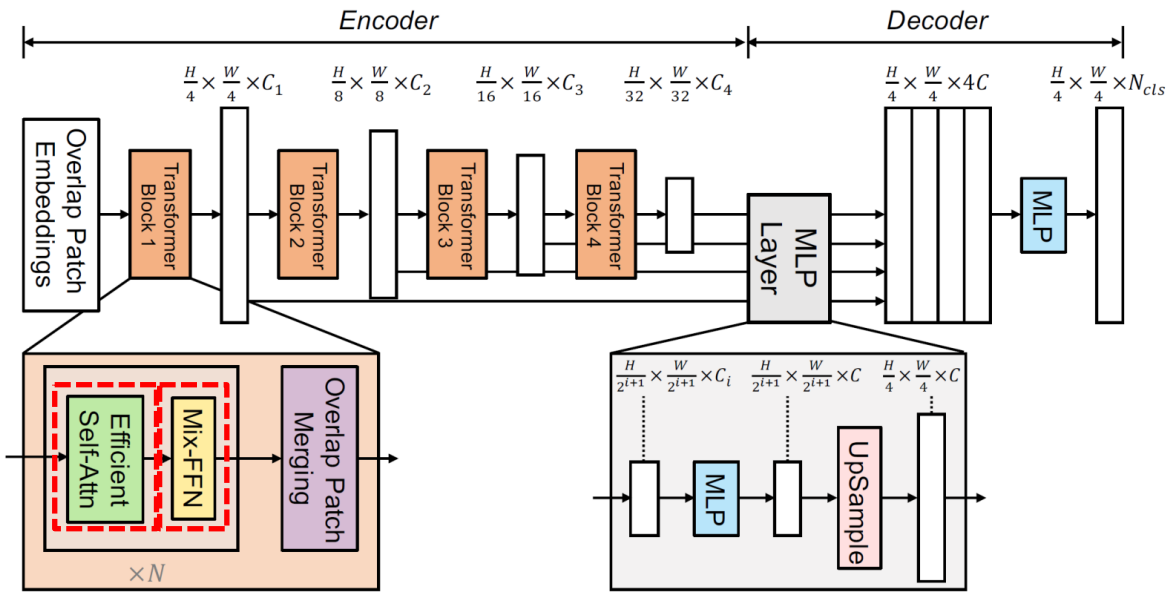


	mIoU	Params	FLOPs	FPS
SegFormer-B0	**37.4**	**3.7M**	**8.4G**	**50.5**
FCN-R50	36.1	49.6M	198.0G	23.5
SegFormer-B2	**46.5**	**27.5M**	**62.4G**	**24.5**
DeeplabV3+/R101	44.1	62.7M	255.1G	14.1
HRNet-W48 + OCR	43.0	70.5M	164.8G	17.0
SegFormer-B4	**50.3**	**64.1M**	**95.7G**	**15.4**
SETR	48.6	318.3M	362.1G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

Xie et al., **SegFormer: Simple and Efficient Design for Semantic Segmentation with Transformers**, arXiv21.

NVIDIA

SegFormer: Encoder Design



Efficient Self-Attention

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

$$\hat{K}=\text{Reshape}(\frac{N}{R},C\cdot R)(K)$$

$$K=\text{Linear}(C\cdot R,C)(\hat{K}),$$

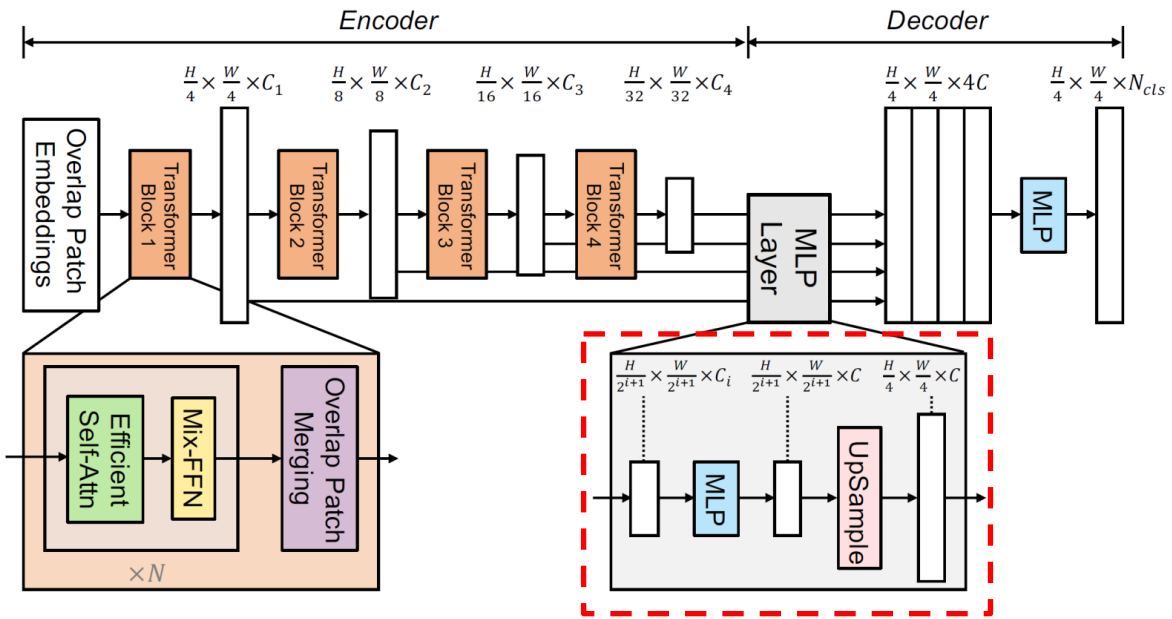
O(N^2) -> O(N^2/R)

Mix-FFN (PE Free)

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

Xie et al., **SegFormer: Simple and Efficient Design for Semantic Segmentation with Transformers**, arXiv21.

SegFormer: Decoder Design

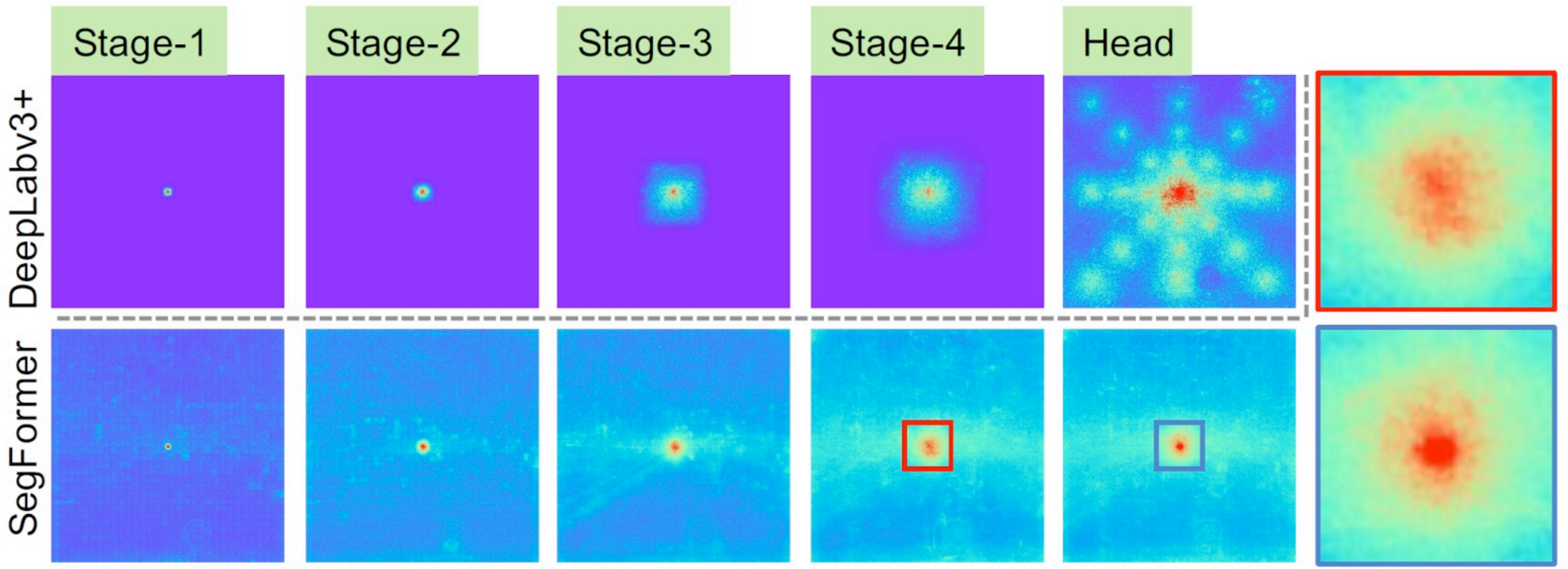


Lightweight all-MLP decoder

$$\hat{F}_i = \text{Linear}(C_i, C)(F_i), \forall i$$
$$\hat{F}_i = \text{Upsample}(\frac{W}{4} \times \frac{W}{4})(\hat{F}_i), \forall i$$
$$F = \text{Linear}(4C, C)(\text{Concat}(\hat{F}_i)), \forall i$$
$$M = \text{Linear}(C, N_{cls})(F),$$

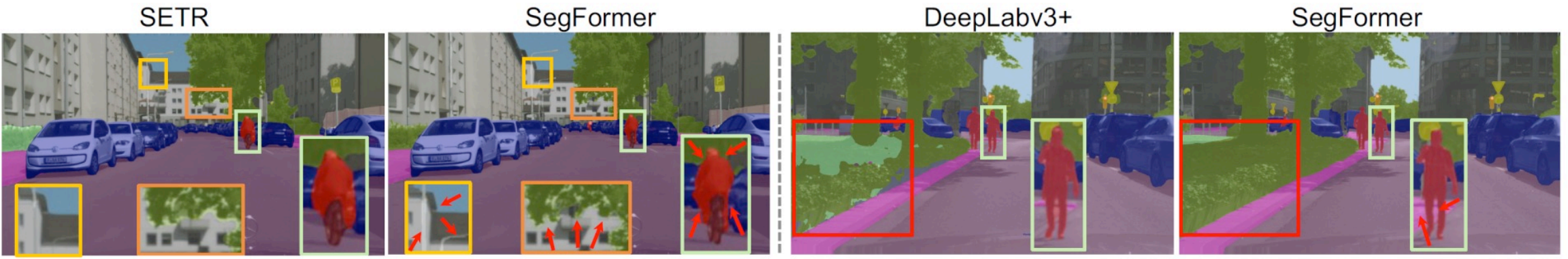
Xie et al., **SegFormer: Simple and Efficient Design for Semantic Segmentation with Transformers**, arXiv21.

SegFormer: Decoder Design

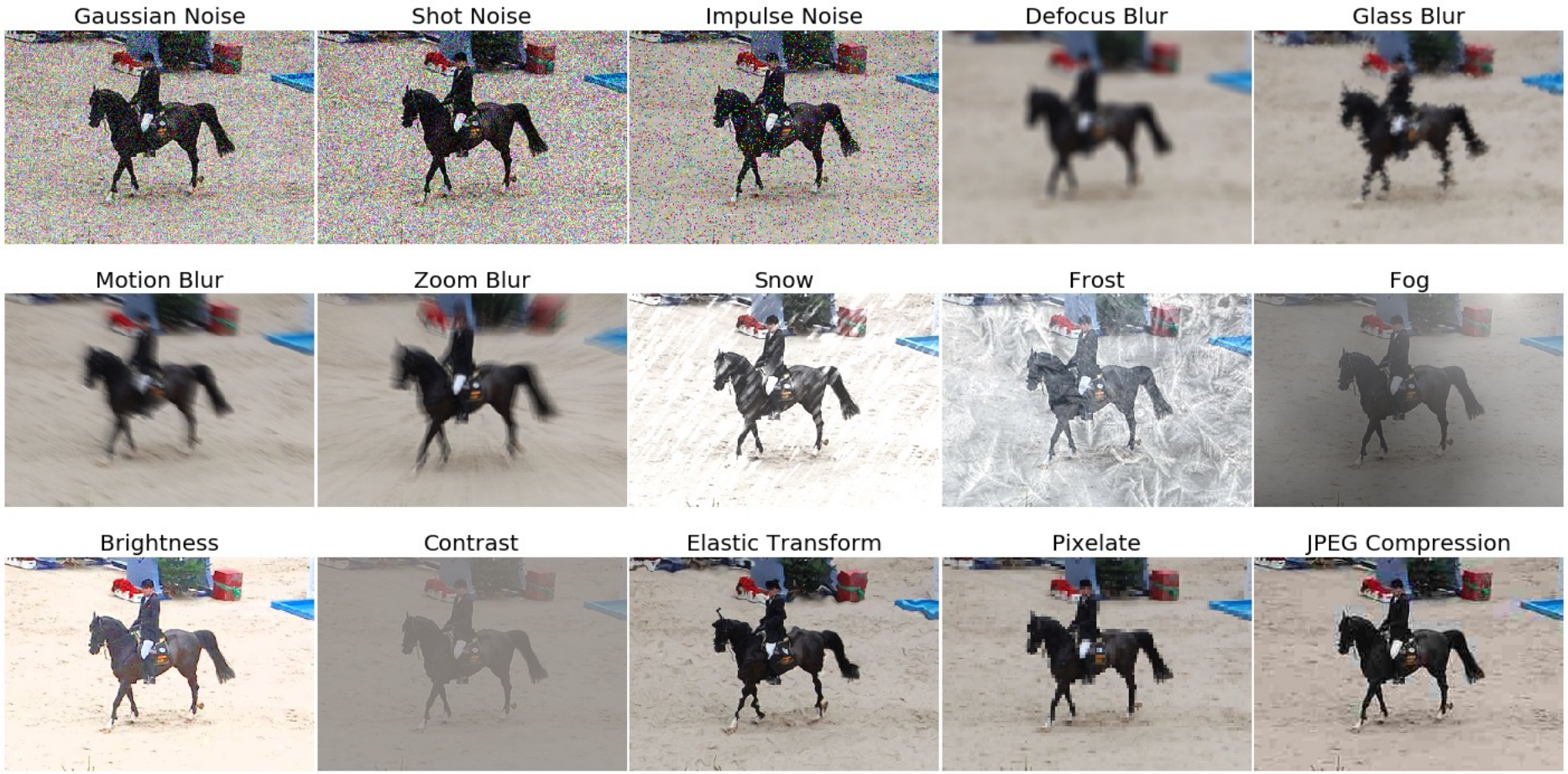


Luo et al., **Understanding the Effective Receptive Field in Deep Convolutional Neural Networks**, NeurIPS16.

Experiment: Qualitative Results



Experiment: Robustness to Corruptions



Experiment: Comparison to SOTA

Table 2: **Comparison to state of the art methods on ADE20K and Cityscapes.** SegFormer has significant advantages on #Params, #Flops, #Speed and #Accuracy. Note that for SegFormer-B0 we scale the short side of image to {1024, 768, 640, 512} to get speed-accuracy tradeoffs.

	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**

Panoptic Segmentation

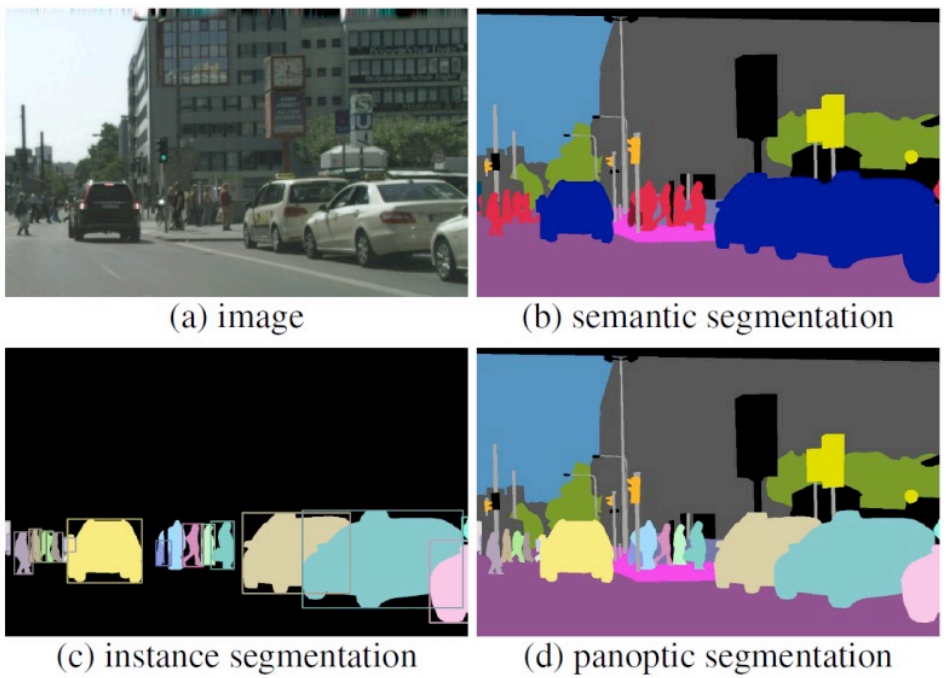
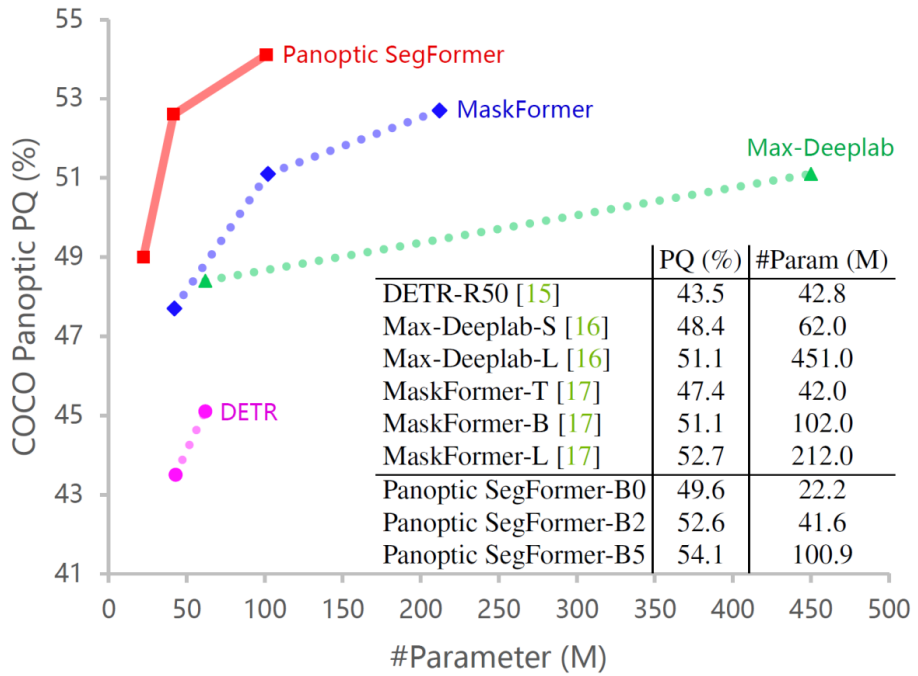


Image credit: Kirillov et al., Panoptic Segmentation, CVPR19.

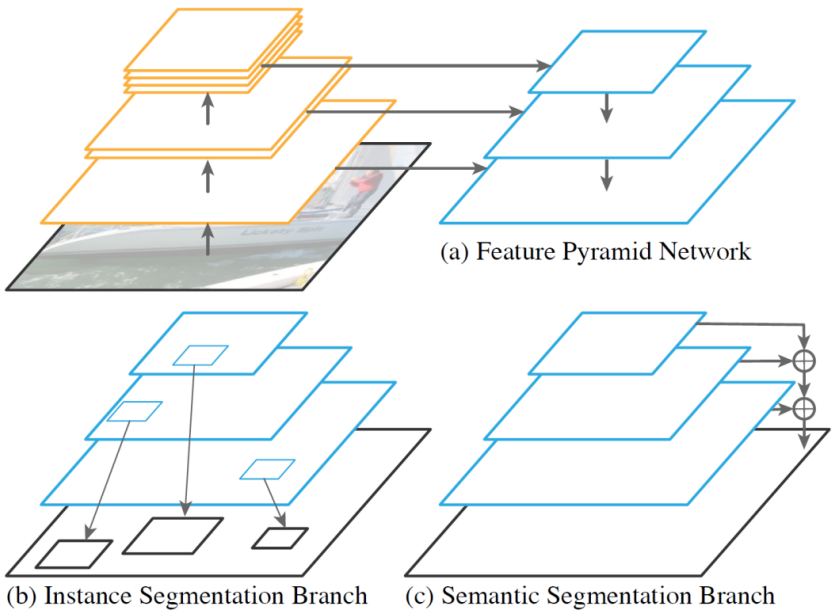


	PQ (%)	#Param (M)
DETR-R50 [15]	43.5	42.8
Max-Deeplab-S [16]	48.4	62.0
Max-Deeplab-L [16]	51.1	451.0
MaskFormer-T [17]	47.4	42.0
MaskFormer-B [17]	51.1	102.0
MaskFormer-L [17]	52.7	212.0
Panoptic SegFormer-B0	49.6	22.2
Panoptic SegFormer-B2	52.6	41.6
Panoptic SegFormer-B5	54.1	100.9

Comparison to the prior arts on the COCO val2017 split.

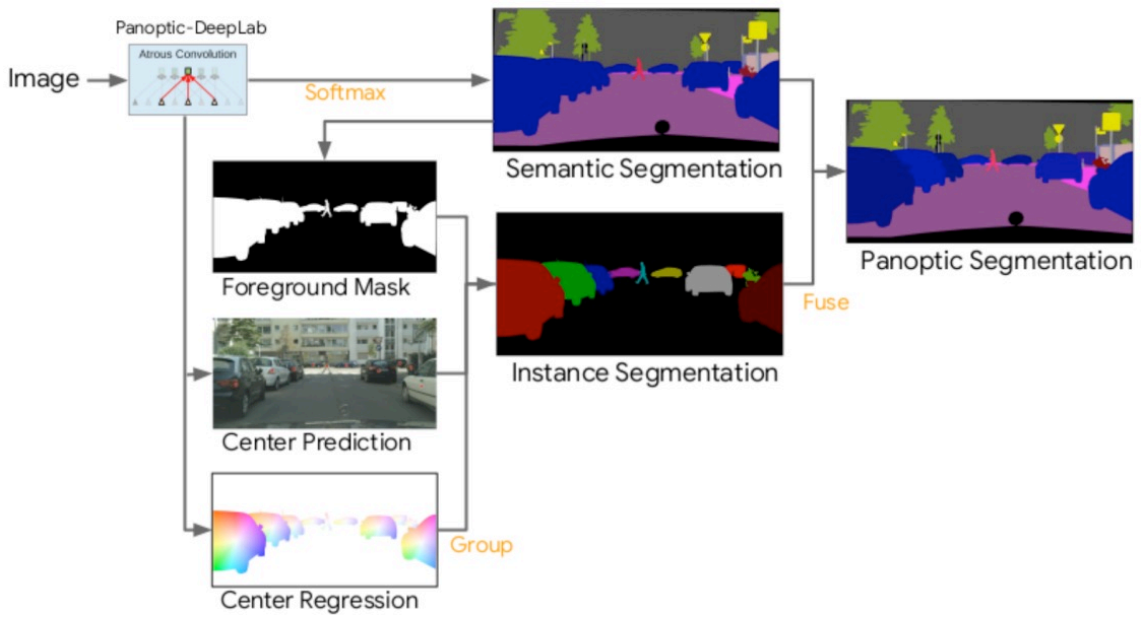
Prior Strategies: Top-Down vs. Bottom-Up

Panoptic-FPN [1]



Top-down strategy: Proposal-conditioned detection and instance segmentation of things (objects)

Panoptic-DeepLab [2]



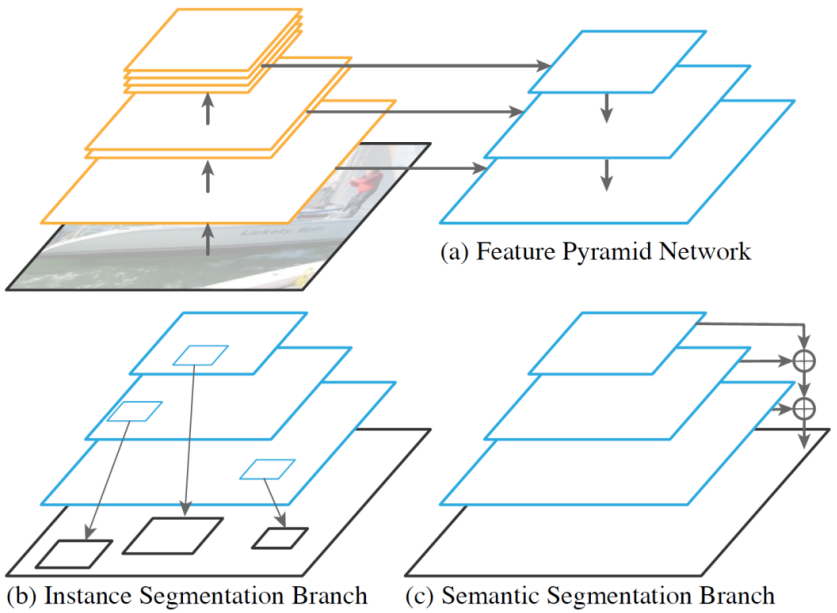
Bottom-up strategy: Proposal-less detection and instance segmentation of things (objects)

[1] Kirillov et al., Panoptic Feature Pyramid Network, CVPR19

[2] Cheng et al., Panoptic-DeepLab: A Simple, Strong, and Fast Baseline for Bottom-Up Panoptic Segmentation, CVPR20

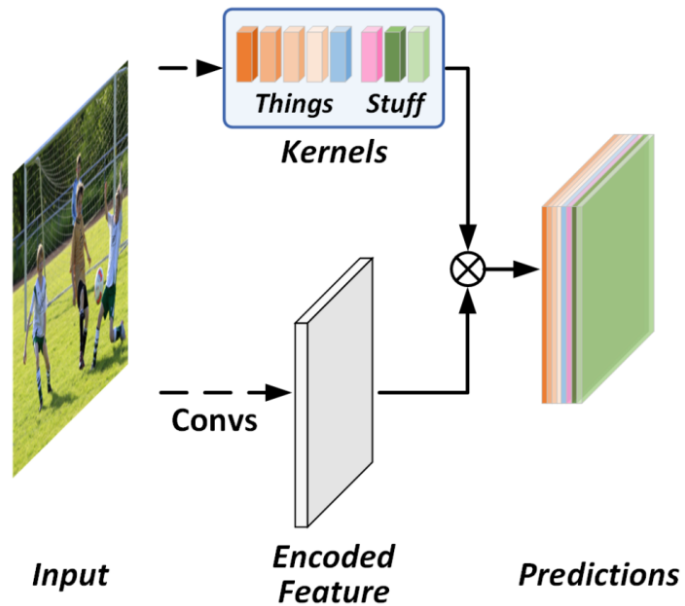
Prior Strategies: Separated vs. Unified

Panoptic-FPN [1]



Separated strategy: Instance segmentation (objects) and semantic segmentation (stuff) tasks are independent.

Panoptic FCN [2]



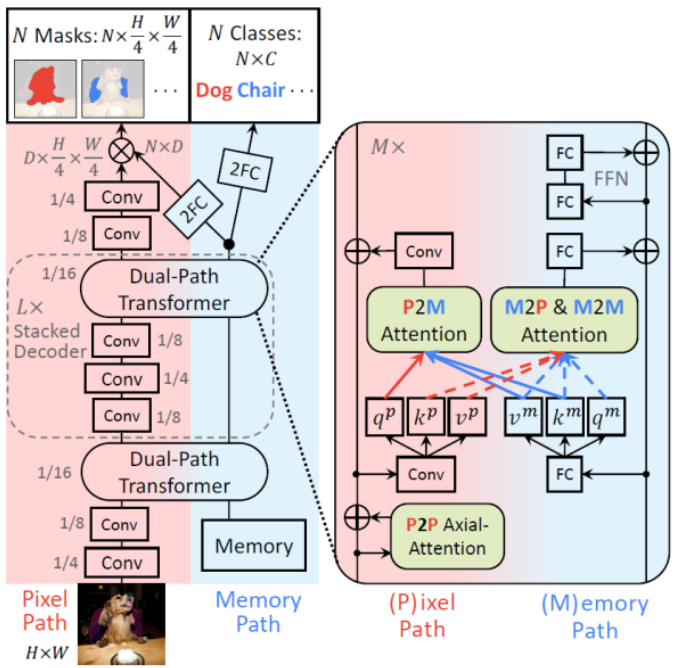
Unified strategy: Top-down (kernel branch) meets bottom-up (feature branch).

[1] Kirillov et al., Panoptic Feature Pyramid Network, CVPR19

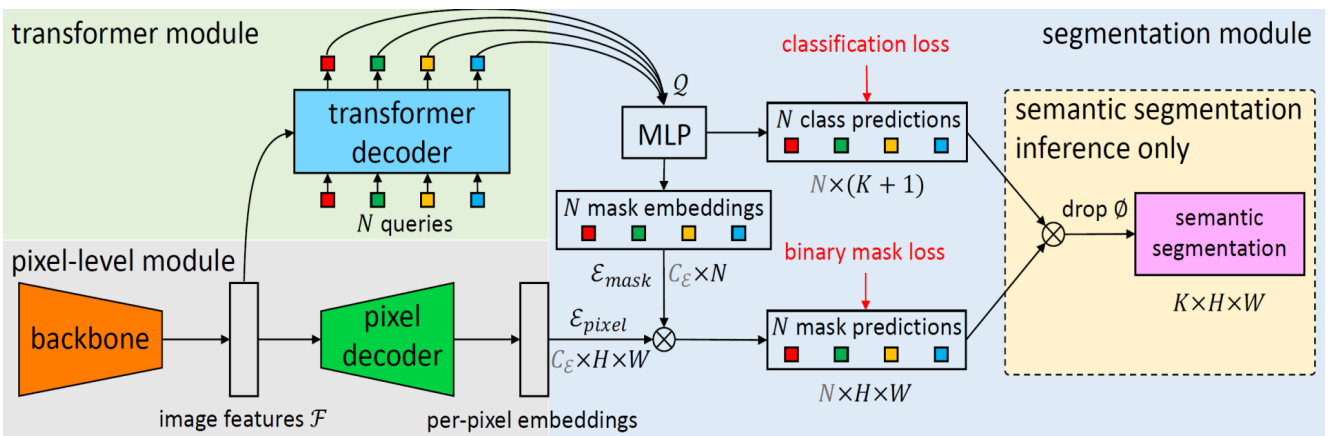
[2] Li et al., Fully Convolutional Networks for Panoptic Segmentation, CVPR21

Panoptic Segmentation with Transformers

Max-DeepLab [1]



MaskFormer [2]

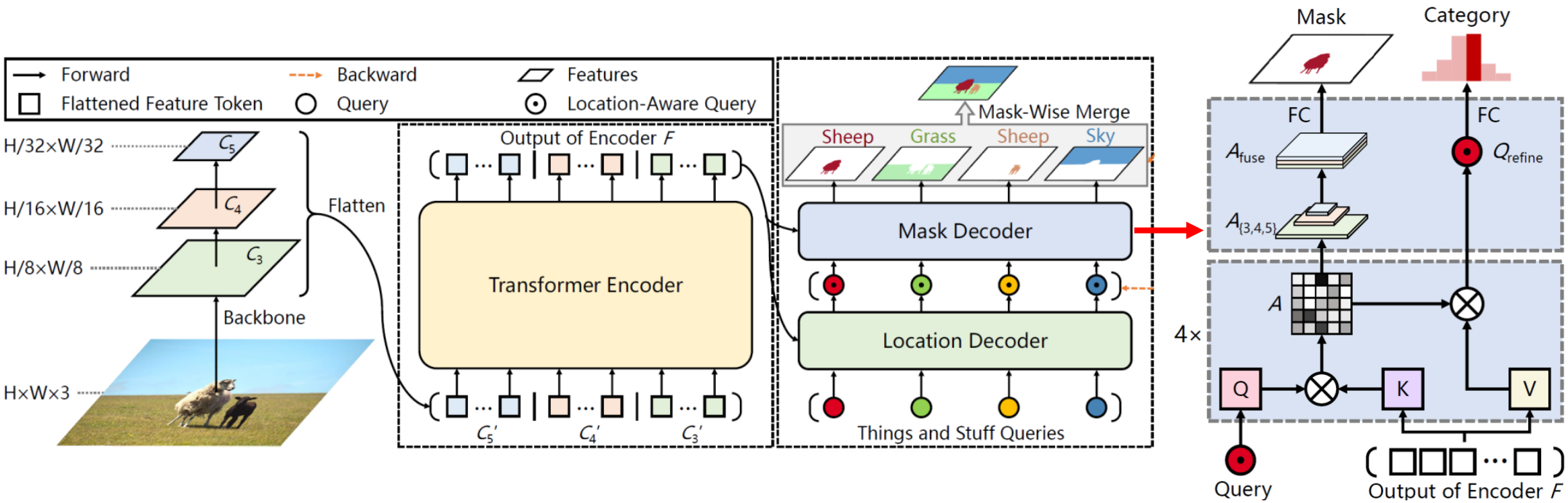


[1] Wang et al., MaX-DeepLab: End-to-End Panoptic Segmentation with Mask Transformers, CVPR21

[2] Cheng et al., MaskFormer: Per-Pixel Classification is Not All You Need for Semantic Segmentation, NeurIPS21

NVIDIA

Proposed Framework



Li et al., Panoptic SegFormer, arXiv21

Quantitative Results (COCO Test-dev)

CodaLab — Search Competitions

Results

#	User	Entries	Date of Last Entry	PQ ▲	SQ ▲	RQ ▲
1	Panoptic-SegFormer-PVTv2	3	09/01/21	0.544 (1)	0.833 (2)	0.646 (1)
2	Innovation	14	10/05/19	0.535 (2)	0.834 (1)	0.633 (3)
3	bc.ifp.uiuc	2	08/04/21	0.533 (3)	0.820 (6)	0.641 (2)
4	MaX-DeepLab	1	12/18/20	0.513 (4)	0.825 (3)	0.613 (4)
5	chensnathan	19	10/03/19	0.502 (5)	0.817 (9)	0.599 (5)
6	DetectoRS	2	11/03/20	0.500 (6)	0.824 (4)	0.595 (8)
7	AntVision	1	10/05/19	0.500 (7)	0.821 (5)	0.598 (7)

Panoptic Segmentation

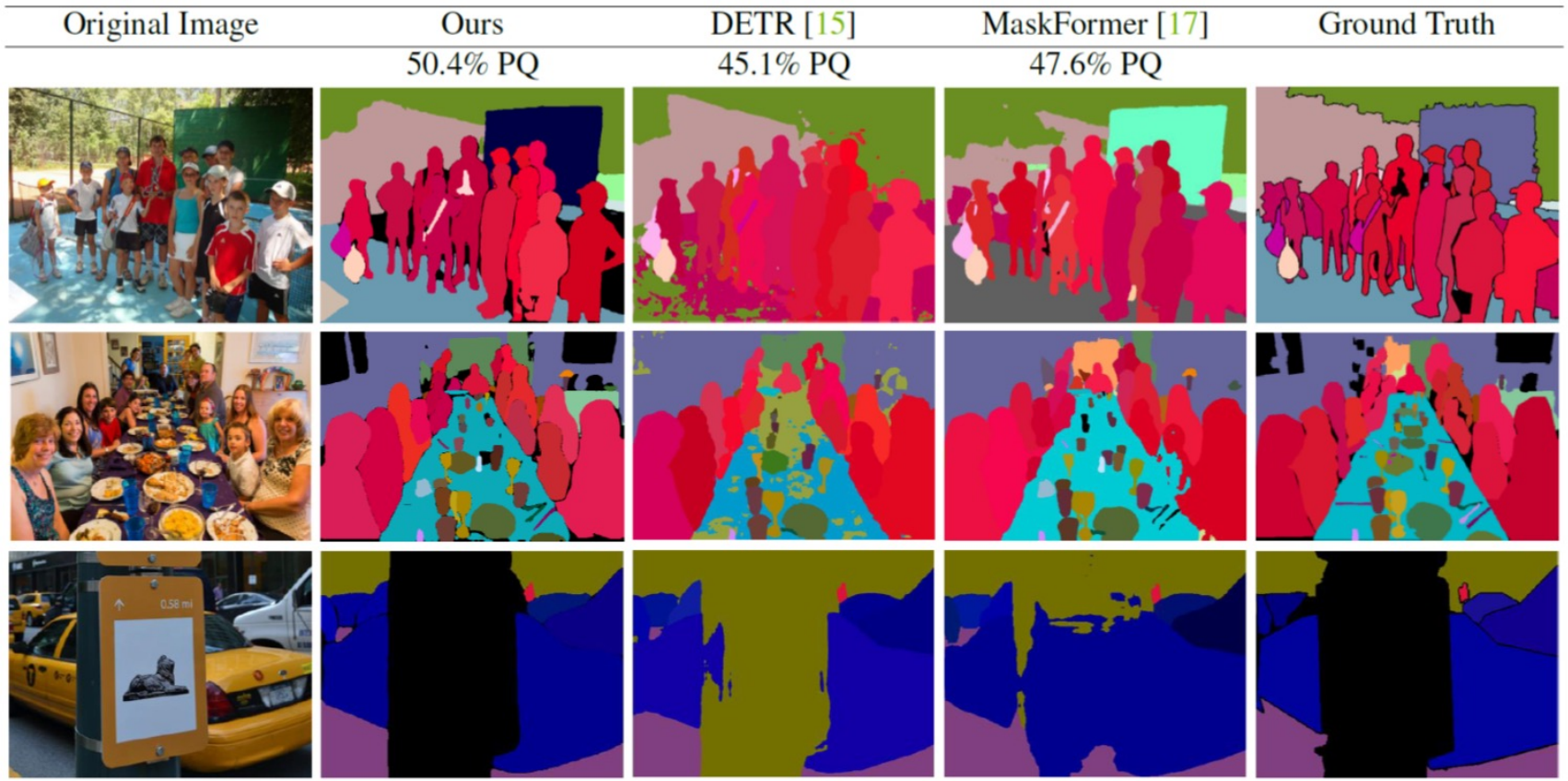
Method	Backbone	Epochs	PQ	PQ^{th}	PQ^{st}	#Param	FLOPs
Panoptic FPN [2]	R101-FPN	36	43.5	50.8	32.5	-	-
DETR [15]	R101	$\sim 150 + 25$	46.0	-	-	61.8M	157G
Panoptic FCN [13]	R101-FPN	36	45.5	51.4	36.4	56.0M	310G
K-Net [14]	R101-FPN	36	47.0	52.8	38.2	-	-
Max-Deeplab-S [16]	Max-S [16]	54	49.0	54.0	41.6	61.9M	162G
K-net [14]	Swin-L†	36	52.1	58.2	42.8	-	-
Max-Deeplab-L [16]	Max-L [16]	54	51.3	57.2	42.4	451.0M	1846G
Innovation [22]	ensemble	-	53.5	61.8	41.1	-	-
Panoptic SegFormer	R50	50	50.0	56.2	40.8	47.0M	246G
Panoptic SegFormer	R101	50	50.9	57.1	41.4	65.9M	322G
Panoptic SegFormer	PVTv2-B5 [40]	50	54.4	61.1	44.3	100.9M	391G

Instance Segmentation

Method	Backbone	Epochs	AP^{seg}	AP^{seg}_S	AP^{seg}_M	AP^{seg}_L
Mask R-CNN [29]	R50-FPN	36	37.5	21.1	39.6	48.3
SOLOv2 [12]	R50-FPN	36	38.8	16.5	41.7	56.2
SOLQ (300 queries) [31]	R50	50	39.7	21.5	42.5	53.1
HTC [41]	R50-FPN	36	40.1	23.3	42.1	52.0
QueryInst (300 queries) [32]	R50-FPN	36	40.6	**23.4**	42.5	52.8
Panoptic SegFormer (300 queries)	R50	50	**41.7**	21.9	**45.3**	**56.3**

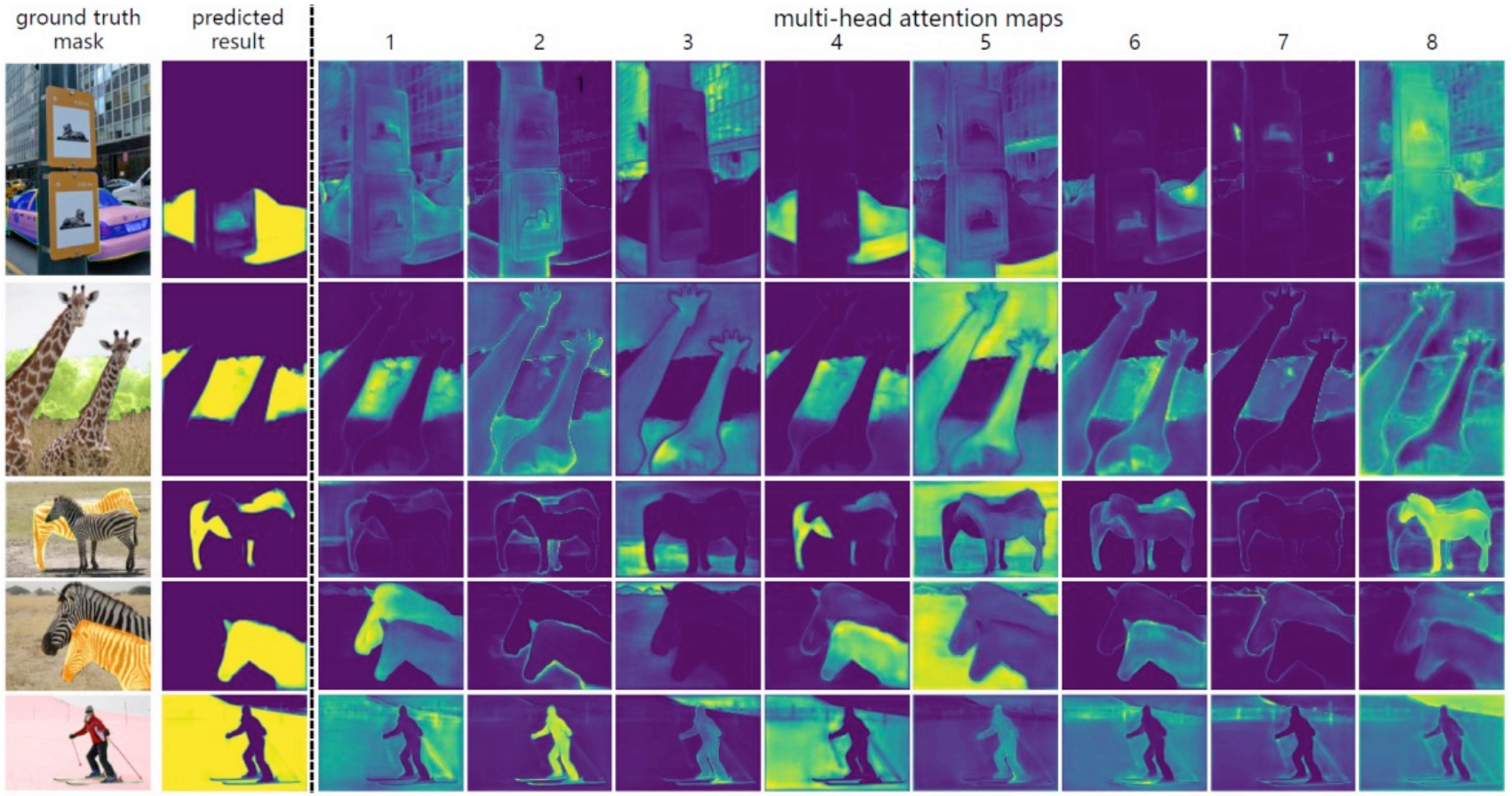
Li et al., Panoptic SegFormer, arXiv21

Segmentation Visualization



Li et al., Panoptic SegFormer, arXiv21

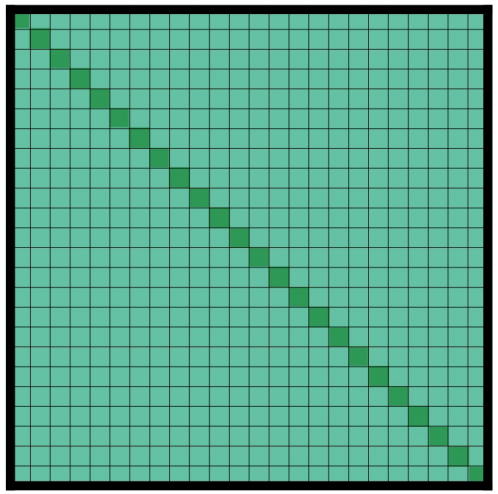
Attention Visualization



Li et al., Panoptic SegFormer, arXiv21

LONG-SHORT TRANSFORMERS

EXPENSIVE SELF ATTENTION



Full n^2 attention

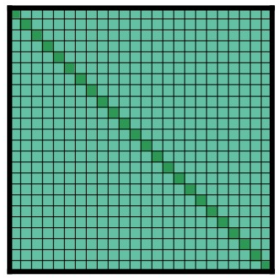
Complexity of self attention scales quadratically with sequence length, prohibitive for long-sequence tasks

CHALLENGES FOR EXISTING EFFICIENT MECHANISMS

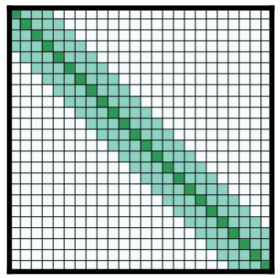
Sparse attention: efficient but attend to only a subset of tokens

Efficient implementation of different patterns for different heads is non-trivial

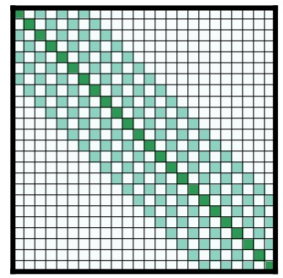
Bidirectional



(a) Full n^2 attention

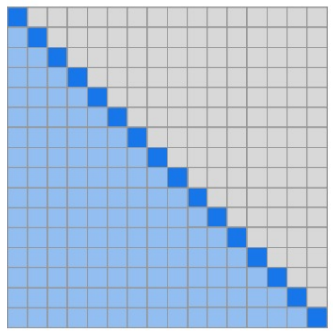


(b) Sliding window attention

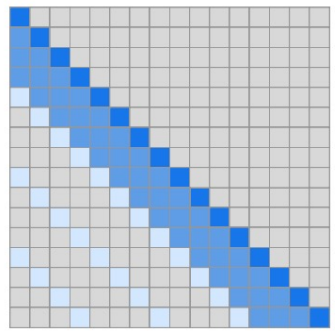


(c) Dilated sliding window

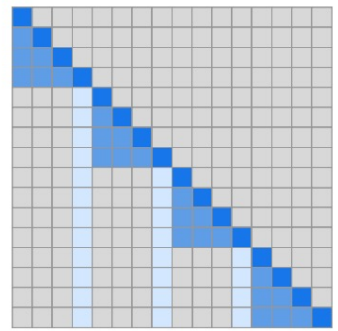
Autoregressive



(a) Transformer



(b) Sliding Transformer (strided)

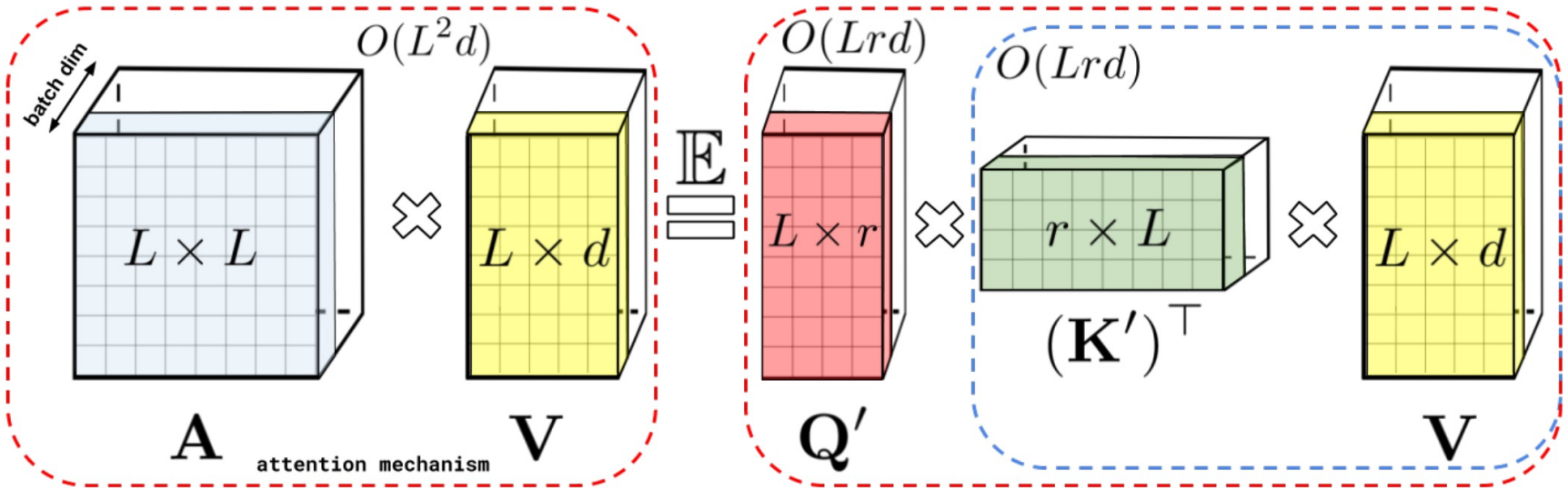


(c) Sparse Transformer (fixed)

CHALLENGES FOR EXISTING METHODS

Low-rank approximation

- Fast, captures global information but loses fine-grained local correlations
- How to efficiently incorporate low-rank methods for autoregressive models?



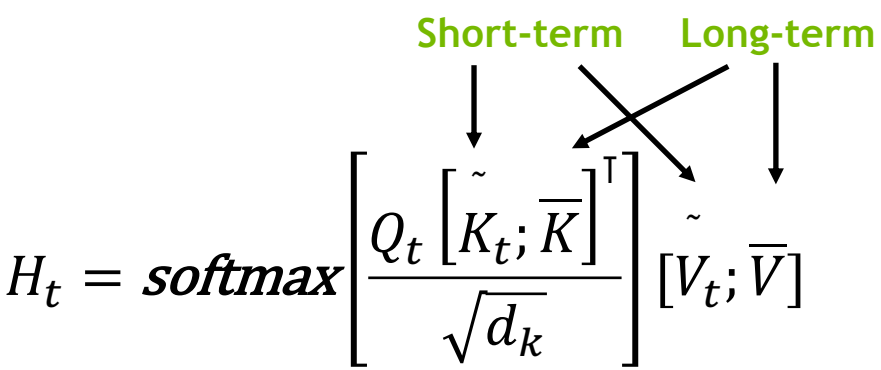
Choromanski, Krzysztof, et al. "Rethinking attention with performers." *ICLR* (2021).

OUR APPROACH

- An aggregation of long-term and short-term attentions
  - Long-term attention: a dynamic low-rank projection
  - Short-term attention: segment-wise sliding windows
- Advantages
  - Better model capacity: can capture the correlation between every pair of tokens for both short-term and long-range pairs
  - Aggregation: works better simple multi-head aggregation
  - Long-term attention: a dynamic low-rank projection flexible to positional changes
  - Short-term attention as a complement for long-term attention in autoregressive models to enable parallelism
  - Short-term attention as an inductive bias to prevent over-smoothing in vision Transformers

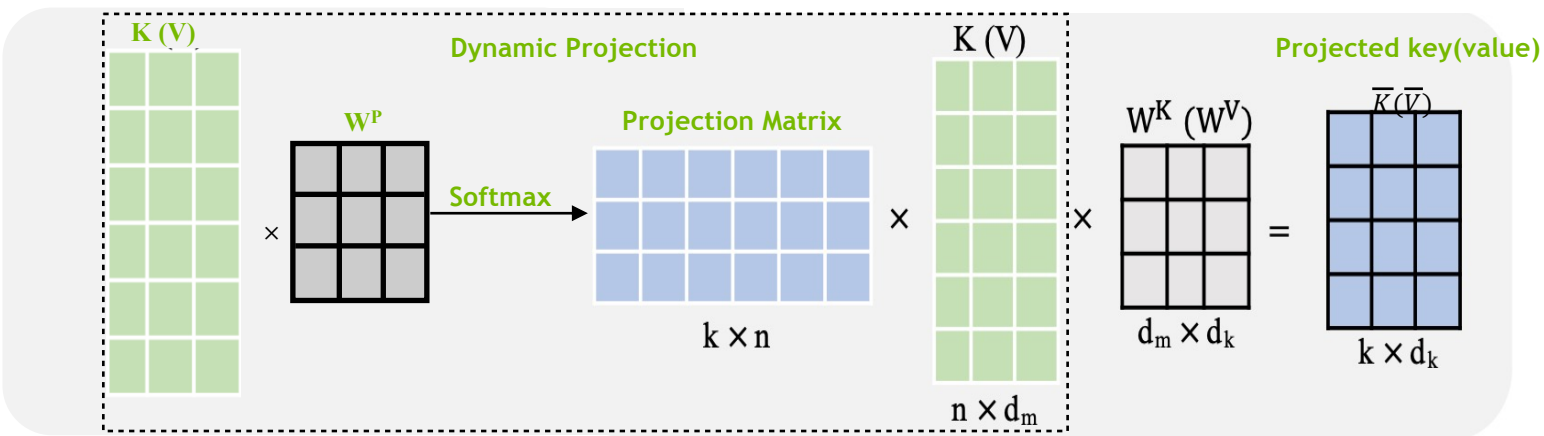
LONG-SHORT TERM ATTENTION

- Allowing each head to use both long-term and short-term representations
- Concatenate the long-term (dynamic projection) and short-term (sliding window) key-value pairs



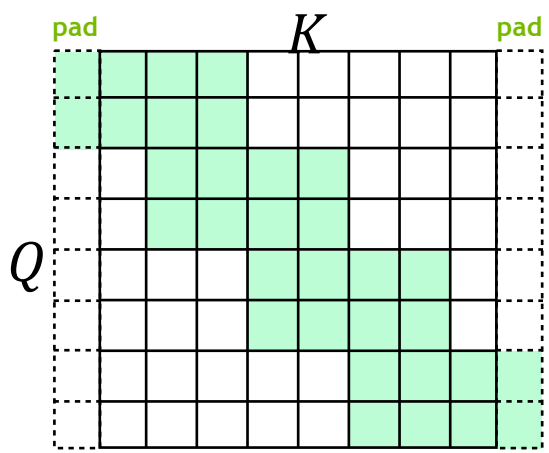
DYNAMIC PROJECTION

- Dynamic projection: predict a low-rank projection matrix for each head
  - The predicted projection matrix is adaptive to position changes

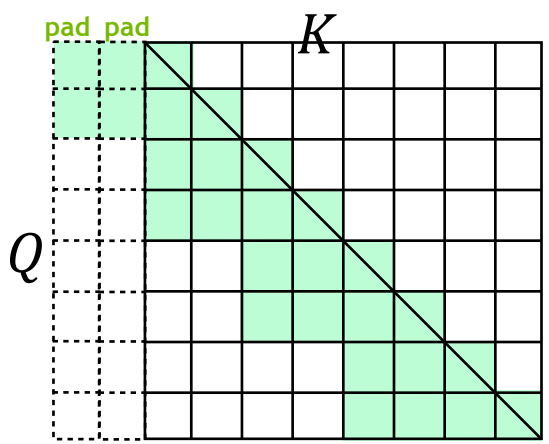


SHORT-TERM ATTENTION

- Segment-wise sliding window attention is currently the fastest while being memory efficient
- Divide the input sequence into disjoint segments with length w
- Each segment attend to: $w/2$ tokens to the left/right of the segment, and the w tokens within the segment.



For Bidirectional Models



For Autoregressive Models

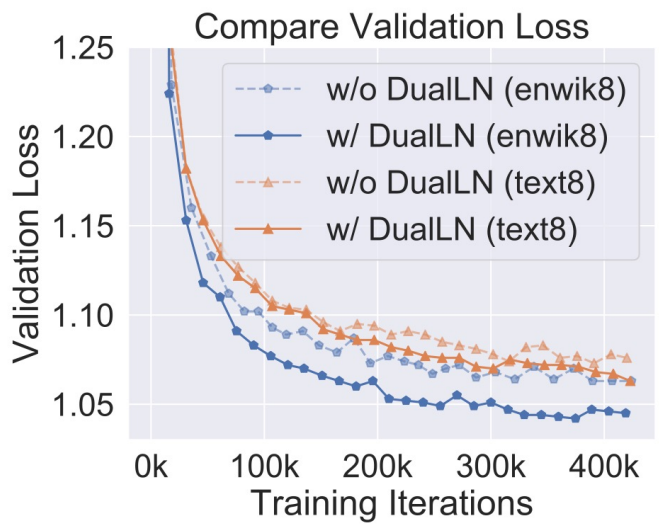
DUAL LAYERNORM

- Apply a pair of LayerNorm to the key-value pairs from long-term and short-term attentions

$$H_t = \boldsymbol{softmax}\left[\frac{Q_t\left[\boldsymbol{LN}_S(\tilde{K}_t); \boldsymbol{LN}_L(\overline{K})\right]^{\mathsf{T}}}{\sqrt{d_k}}\right][\boldsymbol{LN}_S(\tilde{V}_t); \boldsymbol{LN}_L(\overline{V})]$$

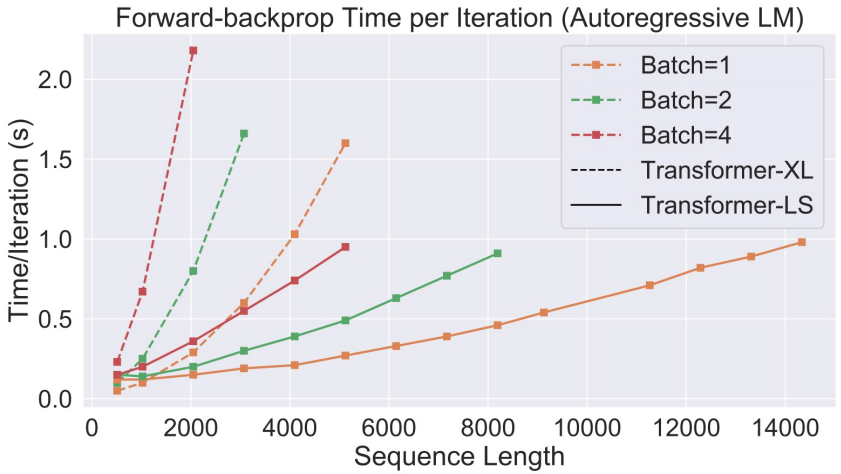
- Align the scales at initialization

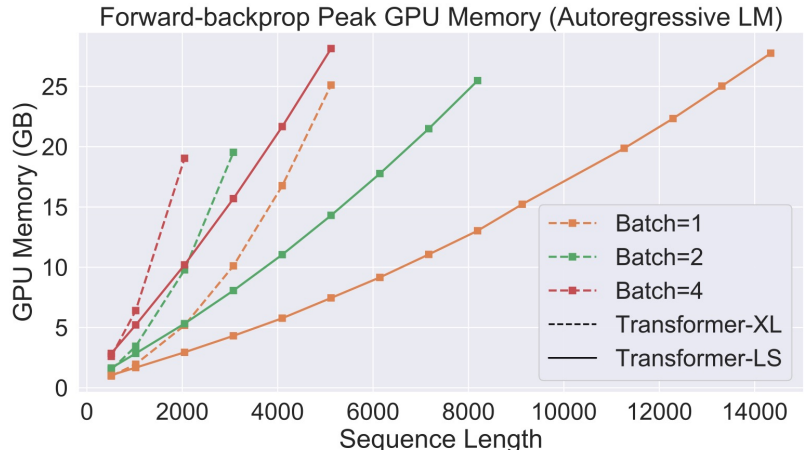
- Allow difference after training



PARALLELISM FOR AUTOREGRESSIVE MODELS

Faster and more memory-efficient than full attention





EXPERIMENTS ON LONG RANGE MODELS

Obtaining better test accuracy than other efficient Transformers with fewer FLOPs

Dynamic projection outperforms Linformer, especially on Text

	ListOps (888 ± 339)		Text (1296 ± 893)		Retrieval (3987 ± 560)		Average
Model	Acc.	FLOPs	Acc.	FLOPs	Acc.	FLOPs	Acc.
Full Attention [1]	37.13	1.21	65.35	4.57	**82.30**	9.14	61.59
Reformer [31] (2)	36.44	0.27	64.88	0.58	78.64	1.15	59.99
Linformer [17] (k=256)	37.38	0.41	56.12	0.81	79.37	1.62	57.62
Performer [28] ($r = 256$)	32.78	0.41	65.21	0.82	81.70	1.63	59.90
Nyströmformer [18] ($l = 128$)	37.34	0.61	65.75	1.02	81.29	2.03	61.46
Transformer-LS ($w, r = 8, 32$)	**37.50**	0.20	**66.01**	0.40	81.79	0.80	**61.77**
Dynamic Projection (best)	37.79	0.15	66.28	0.69	81.86	2.17	61.98
Transformer-LS (best)	**38.36**	0.16	**68.40**	0.29	**81.95**	2.17	**62.90**

EXPERIMENTS ON CHAR-LEVEL LM

Improved results

- Memory and computational Efficient; allowing training on longer sequences, get good results with fewer parameters
- Benefit from improved capacity compared with Longformer; every attention block gets access to global information

Table 2: BPC (↓) of smaller models on enwik8 and text8 (left), and larger models on enwik8 (right).

Method	#Param	text8		enwik8	
		Dev	Test	Dev	Test
T12 [51]	44M	-	1.18	-	1.11
Transformer-XL [9]	41M	-	-	-	1.06
Reformer [31]	-	-	-	-	1.05
Adaptive [52]	38M	1.05	1.11	1.04	1.02
BP-Transformer [53]	38M	-	1.11	-	1.02
Longformer [20]	41M	1.04	1.10	1.02	1.00
Transformer-LS	44M	1.03	**1.09**	1.01	**0.99**

Method	#Param	Test BPC
Transformer-XL [9]	88M	1.03
Transformer-XL [9]	277M	0.99
Routing [32]	223M	0.99
Longformer [14]	102M	0.99
Sparse [12]	95M	0.99
Adaptive [52]	209M	0.98
Compressive [10]	227M	**0.97**
Transformer-LS	110M	**0.97**

EXPERIMENTS ON IMAGENET

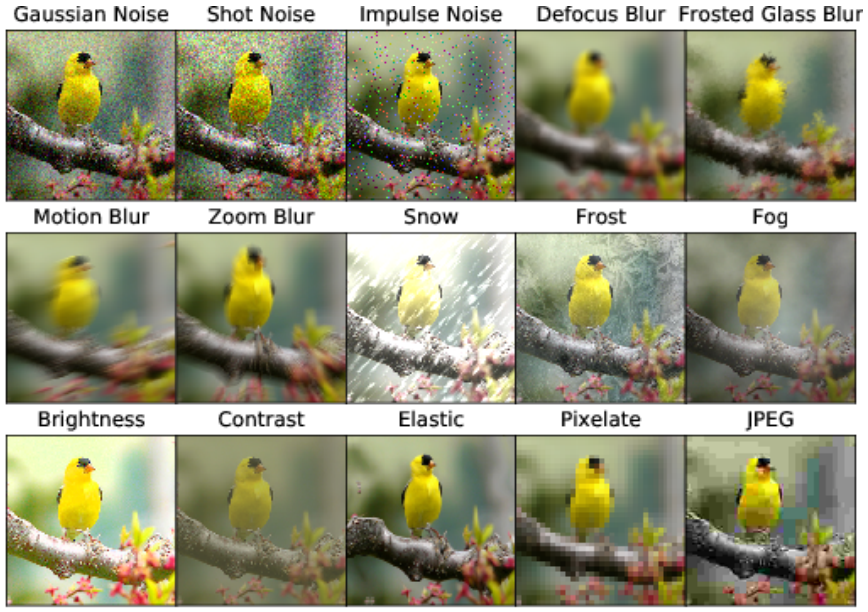
- 12K image patches at 448×448, prohibitive for full attention
- Linear attention, more efficient in terms of FLOPs

Method	Model	#Param (M)	Image Size	FLOPs (G)	ImageNet top-1 (%)	Real top-1 (%)	V2 top-1 (%)
CNN	ResNet-50	25	224^2	4.1	76.2	82.5	63.3
	ResNet-101	45	224^2	7.9	77.4	83.7	65.7
	ResNet-152	60	224^2	11	78.3	84.1	67.0
Transformer	DeiT-S [36]	22	224^2	4.6	79.8	85.7	68.5
	DeiT-B [36]	86	224^2	17.6	81.8	86.7	70.9
	PVT-Medium [5]	44	224^2	6.7	81.2	-	-
	PVT-Large [5]	61	224^2	9.8	81.7	-	-
	Swin-S [38]	50	224^2	8.7	83.2	-	-
	Swin-B [38]	88	224^2	15.4	83.5	-	-
	PVTv2-B4 [56]	62.6	224^2	10.1	83.6	-	-
	PVTv2-B5 [56]	82.0	224^2	11.8	83.8	-	-
	ViT-B/16 [4]	86	384^2	55.5	77.9	-	-
	ViT-L/16 [4]	307	384^2	191.1	76.5	-	-
	DeiT-B [36]	86	384^2	55.5	83.1	-	-
	Swin-B [38]	88	384^2	47.1	**84.5**	-	-
	CvT-13 [6]	20	224^2	6.7	81.6	86.7	70.4
	CvT-21 [6]	32	224^2	10.1	82.5	87.2	71.3
	CvT*-LS-13	20.3	224^2	4.9	81.9	87.0	70.5
	CvT*-LS-17	23.7	224^2	9.8	82.5	87.2	71.6
	CvT*-LS-21	32.1	224^2	7.9	82.7	87.5	71.9
	CvT*-LS-21S	30.1	224^2	11.3	**82.9**	87.4	71.7
	CvT-13 [6]	20	384^2	31.9	83.0	87.9	71.9
	CvT-21 [6]	32	384^2	45.0	83.3	87.7	71.9
	CvT*-LS-21	32.1	384^2	23.9	83.2	88.0	72.5
	CvT*-LS-21	32.1	448^2	34.2	**83.6**	88.2	72.9
	ViL-Small [14]	24.6	224^2	4.9	82.4	-	-
	ViL-Medium [14]	39.7	224^2	8.7	83.5	-	-
	ViL-Base [14]	55.7	224^2	13.4	83.7	-	-
	ViL-LS-Medium	39.8	224^2	8.7	83.8	-	-
	ViL-LS-Base	55.8	224^2	13.4	**84.1**	-	-
	ViL-LS-Medium	39.9	384^2	28.7	**84.4**	-	-

ROBUSTNESS

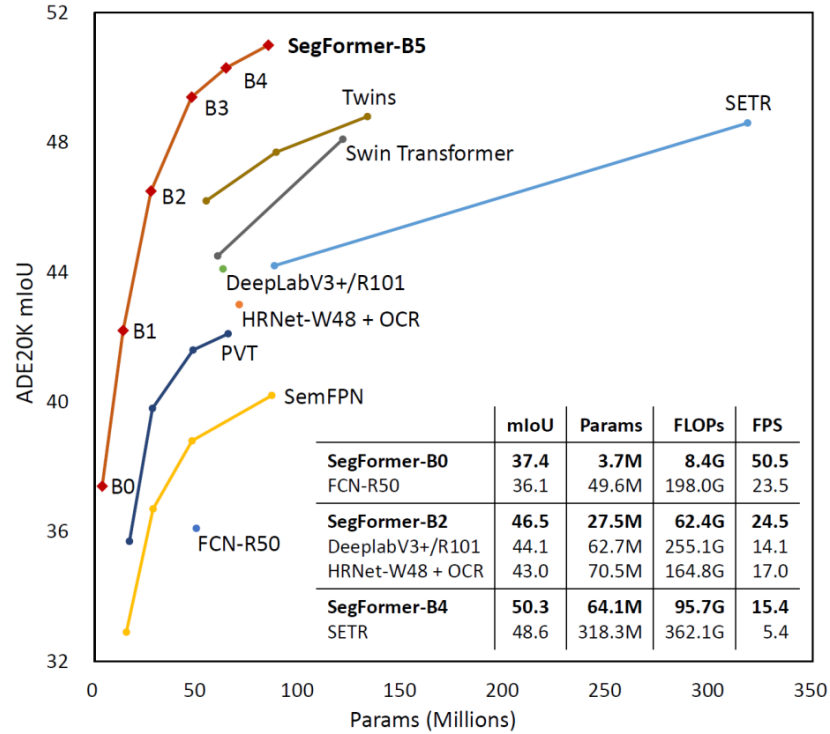
Consistent improvement in robustness with our efficient attention

Model	Params	ImageNet	IN-C [57]	IN-A [58]	IN-R [59]	ImageNet-9 [60]	
	(M)	Top-1	mCE (↓)	Acc.	Acc.	Mixed-same	Mixed-rand
ResNet-50 [35]	25.6	76.2	78.9	6.2	35.3	87.1	81.6
DeiT-S [36]	22.1	79.8	57.1	19.0	41.9	89.1	84.2
CvT*-LS-13	20.3	81.9	58.7	27.0	42.6	90.7	85.6
CvT*-LS-21	32.1	**82.7**	**55.2**	**29.3**	**45.0**	**91.5**	**85.8**

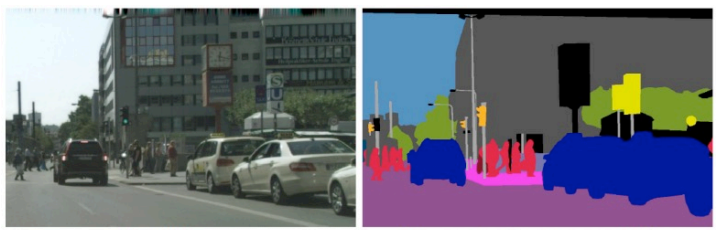


Conclusion

- The emergence of Transformers has transformed vision tasks
- Self-attention in transformers has ability to capture long-range correlations
- Segformer is a simple and efficient design for Semantic Segmentation with Transformers
- Long-short transformer captures both long-range and short attentions efficiently

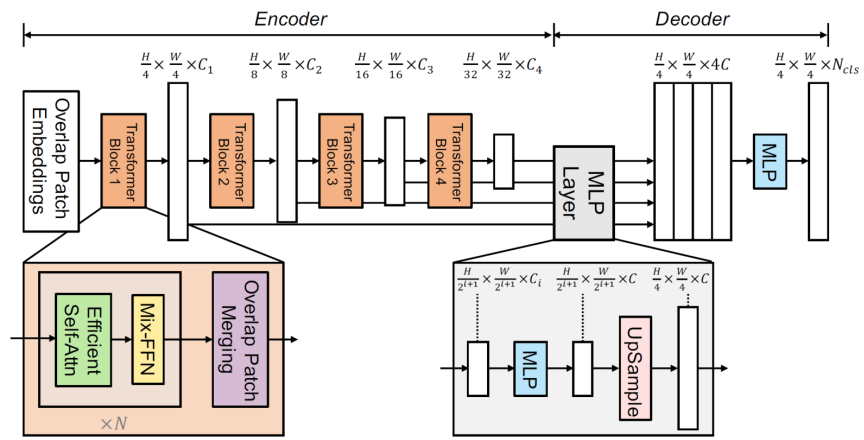


	mIoU	Params	FLOPs	FPS
SegFormer-B0	**37.4**	**3.7M**	**8.4G**	**50.5**
FCN-R50	36.1	49.6M	198.0G	23.5
SegFormer-B2	**46.5**	**27.5M**	**62.4G**	**24.5**
DeeplabV3+/R101	44.1	62.7M	255.1G	14.1
HRNet-W48 + OCR	43.0	70.5M	164.8G	17.0
SegFormer-B4	**50.3**	**64.1M**	**95.7G**	**15.4**
SETR	48.6	318.3M	362.1G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



(a) image (b) semantic segmentation

(c) instance segmentation (d) panoptic segmentation



NVIDIA

Thank You!