Deep Learning

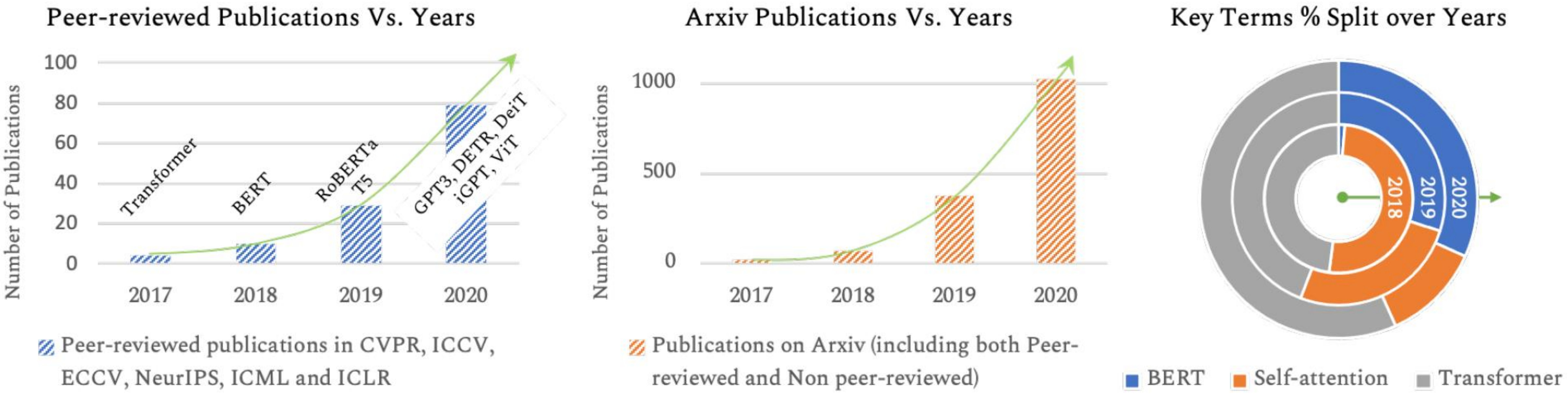
Transformers in computer vision

Danijel Skočaj
University of Ljubljana
Faculty of Computer and Information Science

Academic year: 2022/23

Transformers in computer vision

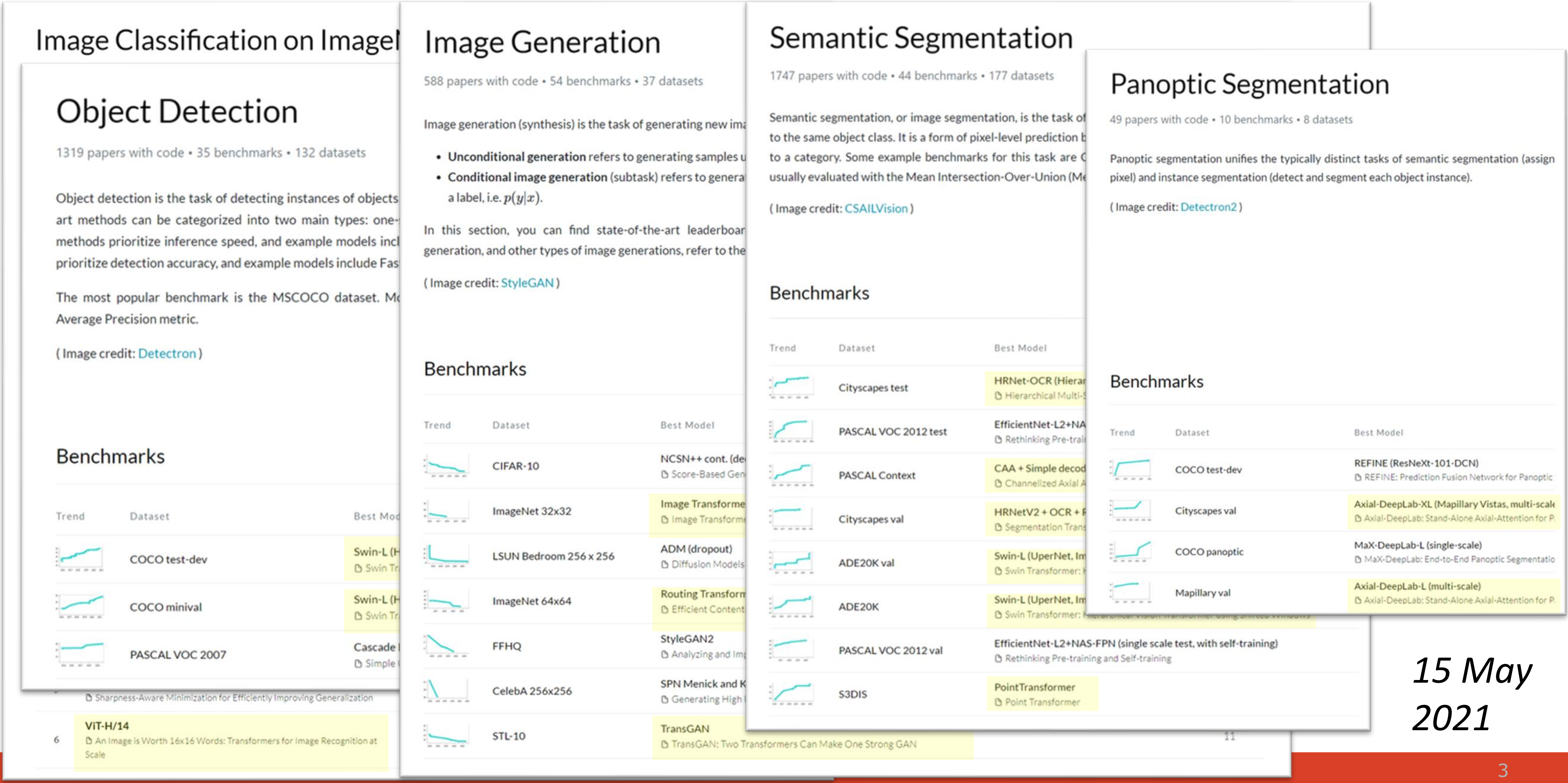
- Transformers in Vision: A Survey



[Khan et al., 2021]

Benchmark leader-boards

[https://paperswithcode.com]



15 May 2021

Image Classification on ImageNet

Leaderboard Dataset

TOP 1 ACCURACY 100 90 80 70 60 50 40

Filter: Ima PatchConv Operations

Rank	Mod			
1	Mo (Vi			
2	Co/			
3	ViT			
4	Co/			
5	DaViT-G	90.4%		1437M
6	Meta Pseudo Labels (EfficientNet-L2)	90.2%	98.8%	480M

Object Detection

2056 papers with code • 60 benchmarks • 188

Object detection is the task of detecting insta art methods can be categorized into two ma methods prioritize inference speed, and exam prioritize detection accuracy, and example mo

The most popular benchmark is the MSCOC Average Precision metric.

(Image credit: Detectron)

Benchmarks

These leaderboards are used to track progres

Trend	Dataset	Best M
	COCO test-dev	🏆 DI
	COCO minival	🏆 DI
	PASCAL VOC 2007	🏆 Ca single-

Image Generation

890 papers with code • 72 benchmarks •

Image generation (synthesis) is the task o

- **Unconditional generation** refers to g
- **Conditional image generation** (subt on a label, i.e. $p(y|x)$.

In this section, you can find state-of- generation, and other types of image gen

(Image credit: StyleGAN)

Benchmarks

These leaderboards are used to track pro

Trend	Dataset
	CIFAR-10
	ImageNet 64x64
	ImageNet 32x32
	STL-10
	FFHQ 256 x 256
	LSUN Bedroom 256 x 256
	FFHQ

🏆 StyleGAN-XL See all

Semantic Segmentation

2714 papers with code • 74 benchm

Semantic segmentation, or image s to the same object class. It is a according to a category. Some exa Models are usually evaluated with

(Image credit: CSAILVision)

Benchmarks

These leaderboards are used to tra

Trend	Dataset	Best Model			
	Cityscapes test				
	ADE20K				
	ADE20K val				
	NYU Depth v2				
	PASCAL VOC 2012 test				
	Cityscapes val	🏆 HRNetV2-OCR+PSA			See all
	PASCAL Context	🏆 CAA + CAR (ConvNeXt-Large + JPU)			See all

Panoptic Segmentation

84 papers with code • 10 benchmarks • 14 datasets

Panoptic segmentation unifies the typically distinct tasks of semantic segmentation (assign a class label to each pixel) and instance segmentation (detect and segment each object instance).

(Image credit: Detectron2)

Benchmarks

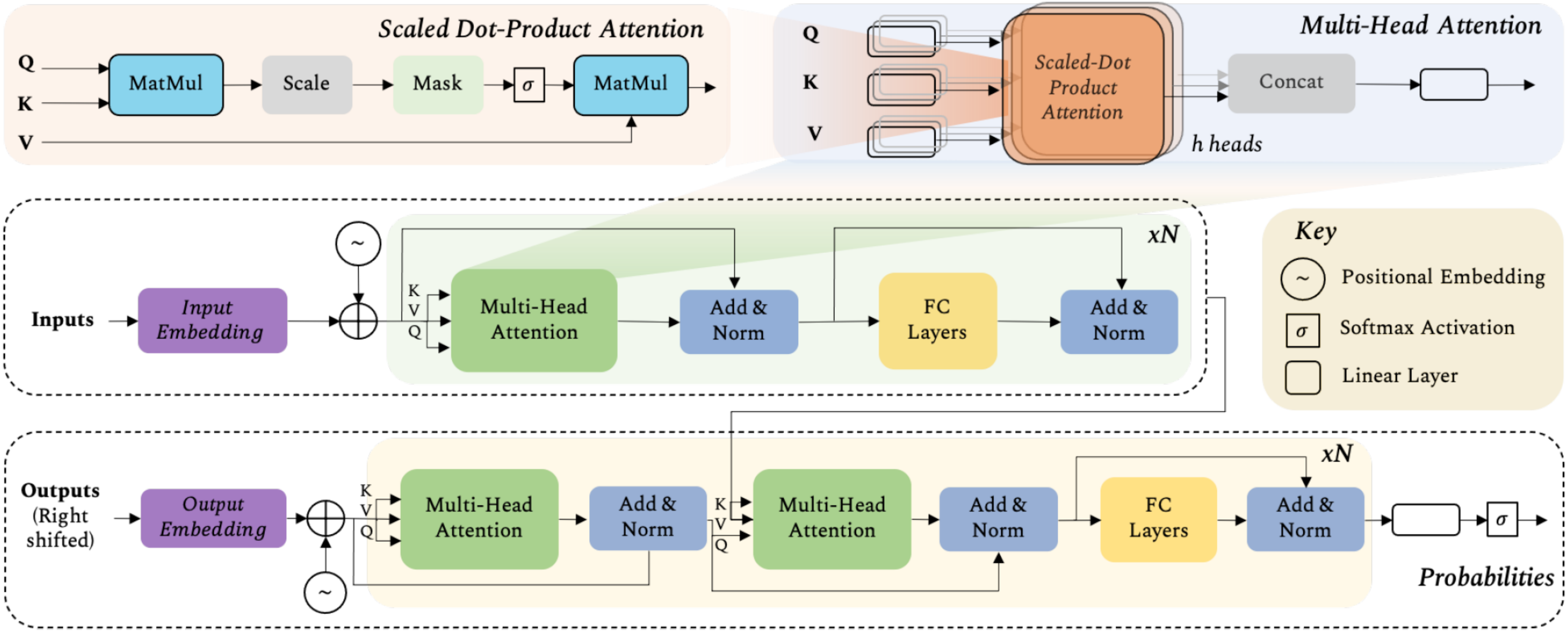
Add a Result

These leaderboards are used to track progress in Panoptic Segmentation

Trend	Dataset	Best Model	Paper	Code	Compare
	COCO test-dev	🏆 Mask2Former (Swin-L)			See all
	Cityscapes val	🏆 Panoptic-DeepLab (SWideRNet-(1, 1, 4.5), Mapillary Vistas, multi-scale)			See all
	COCO minival	🏆 Mask2Former (single-scale)			See all
	Mapillary val	🏆 Panoptic FCN* (Swin-L, single-scale)			See all
	Cityscapes test	🏆 Panoptic-DeepLab (SWideRNet-(1, 1, 4.5))			See all

5 May 2022

Transformers architecture



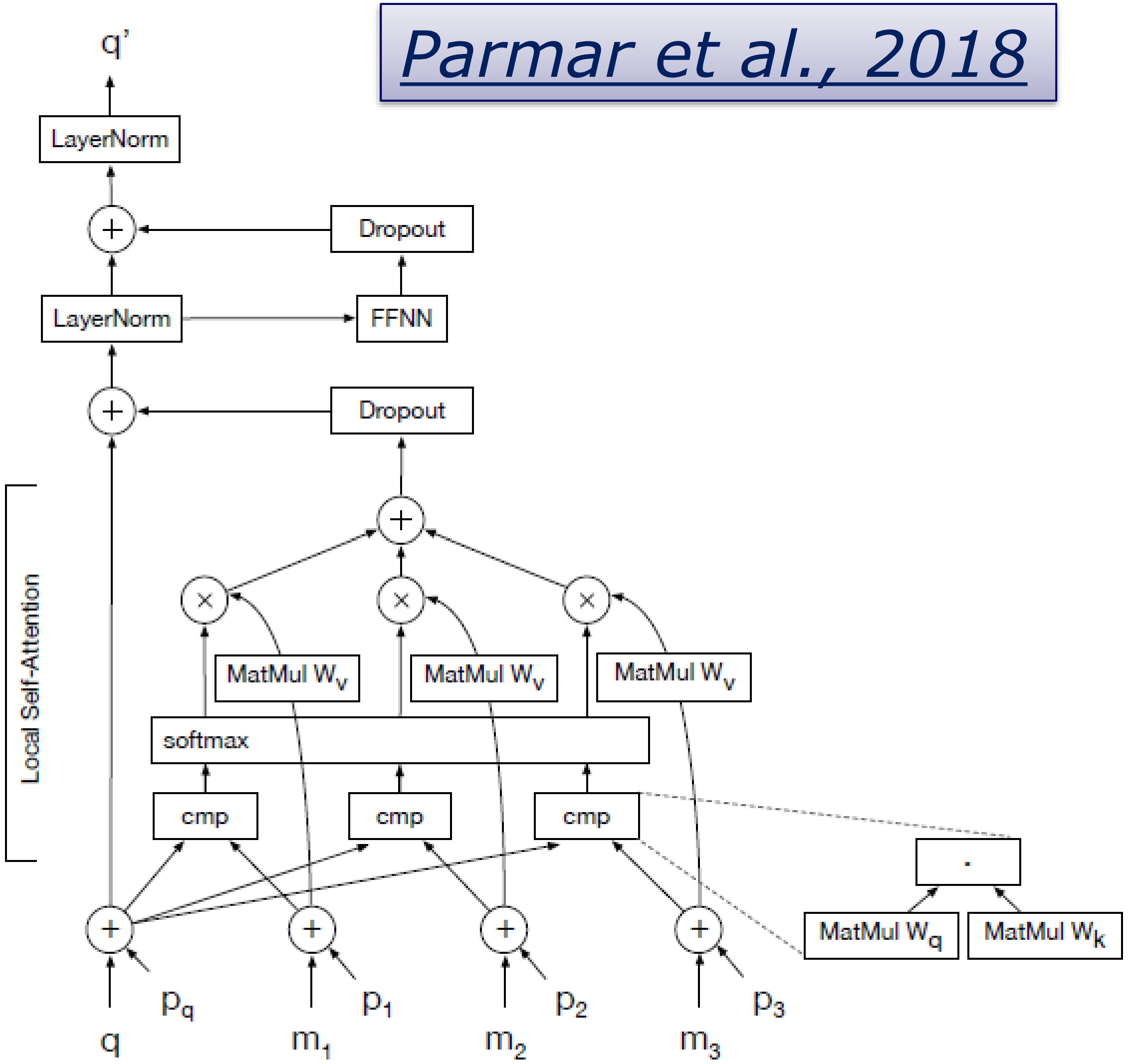
[Vaswani et al., 2017]

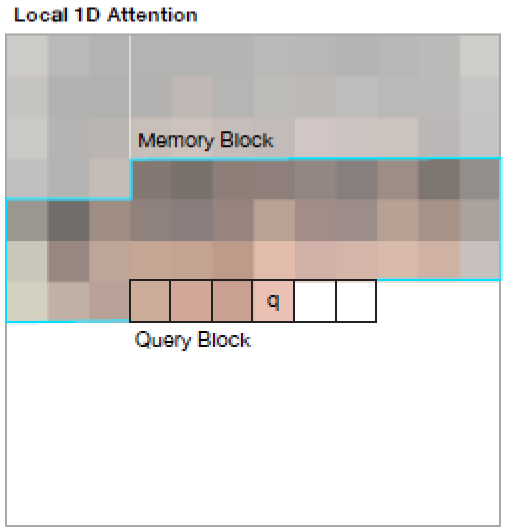
[Khan et al., 2021]

Image Transformer

- Image generation as an autoregressive sequence generation problem
- Encoder-decoder architecture
- Self-attention restricted to local neighbourhoods
- Still large receptive field
- Image generation and super-resolution

Parmar et al., 2018





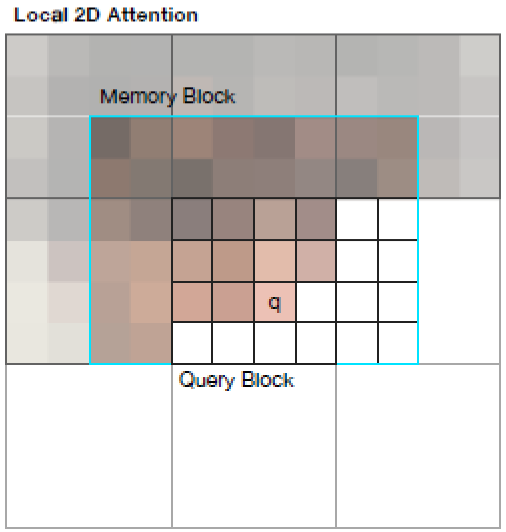
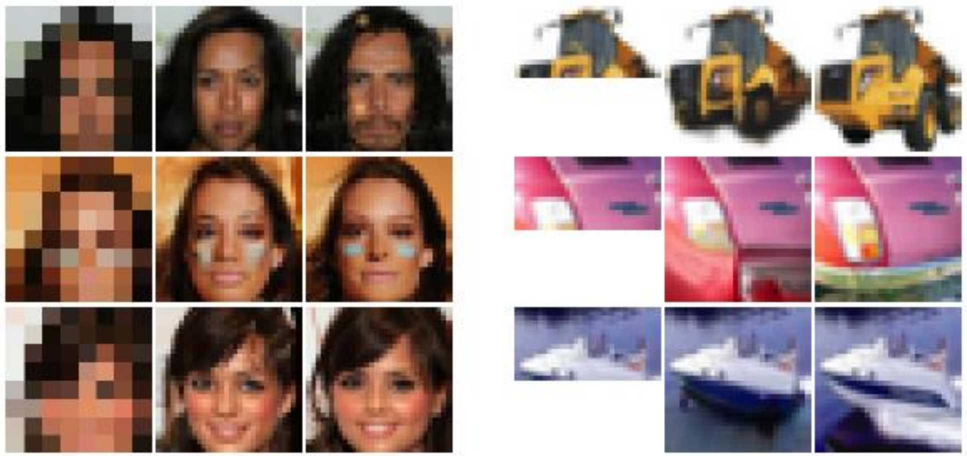
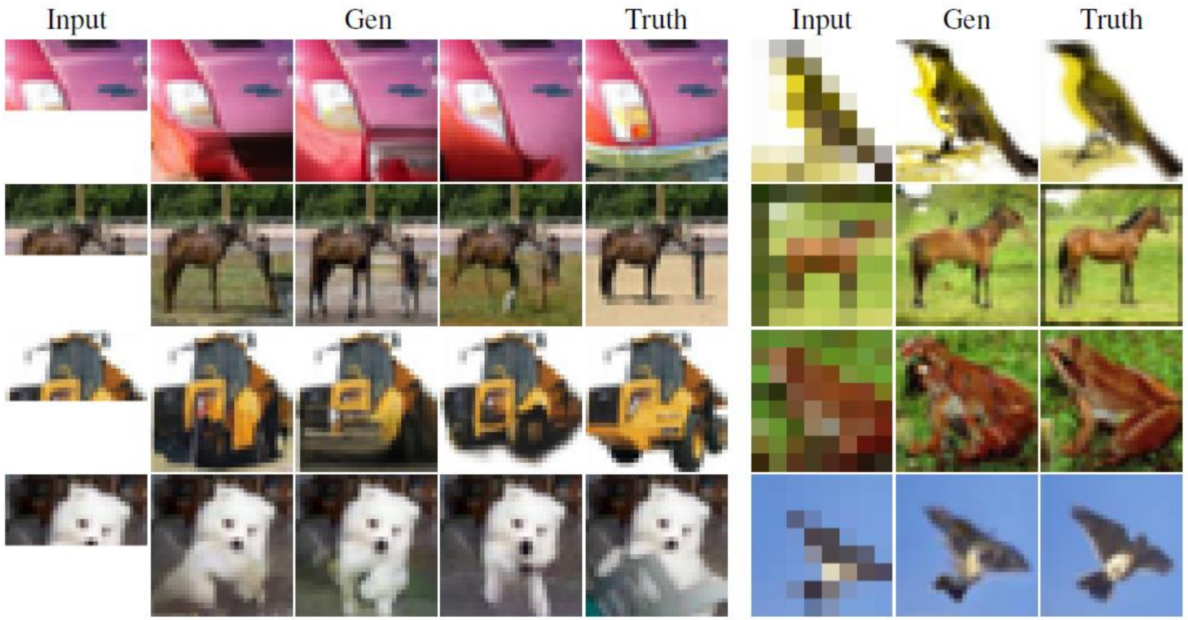
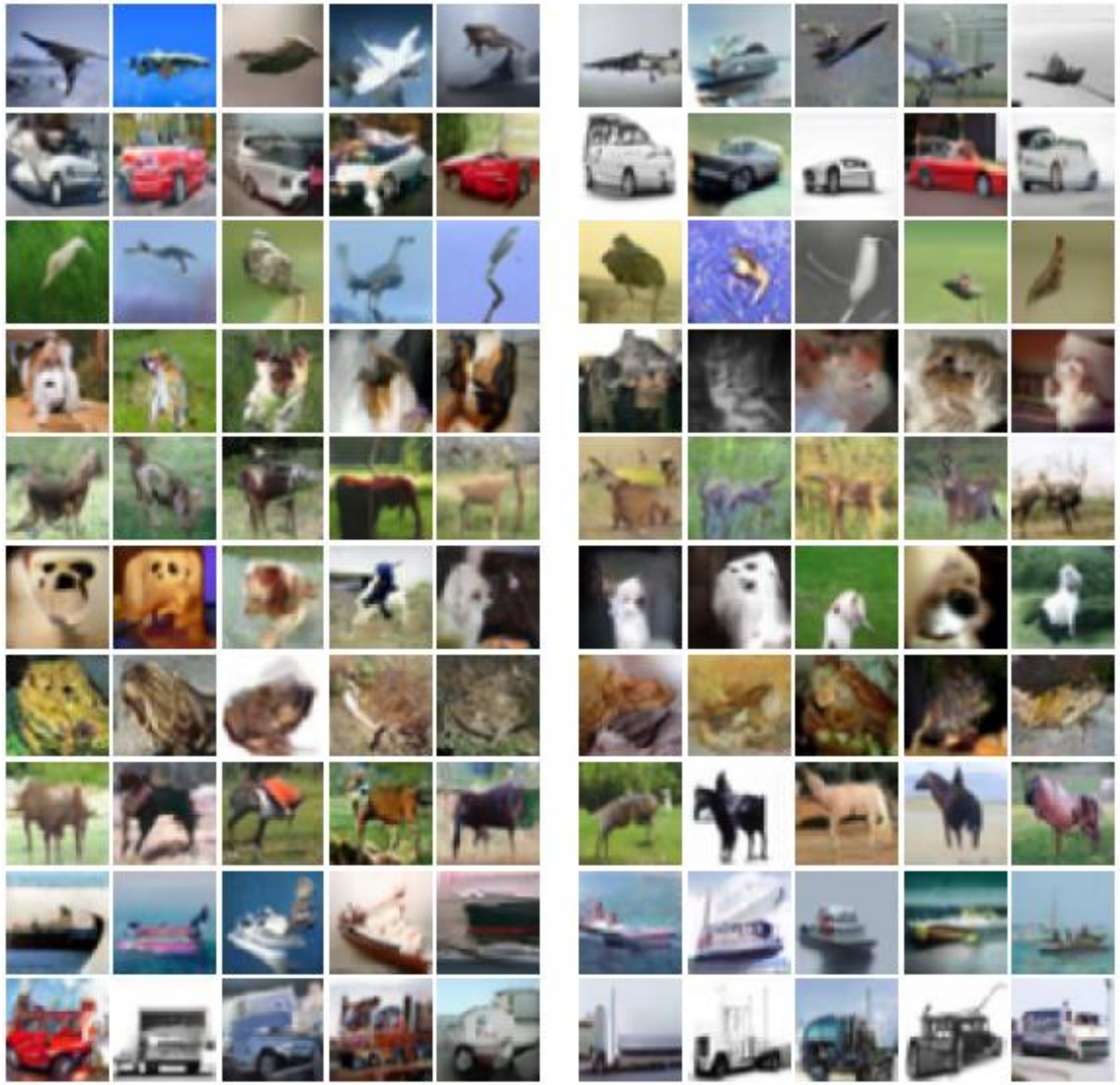


Image Transformer results







ViT - Vision Transformer

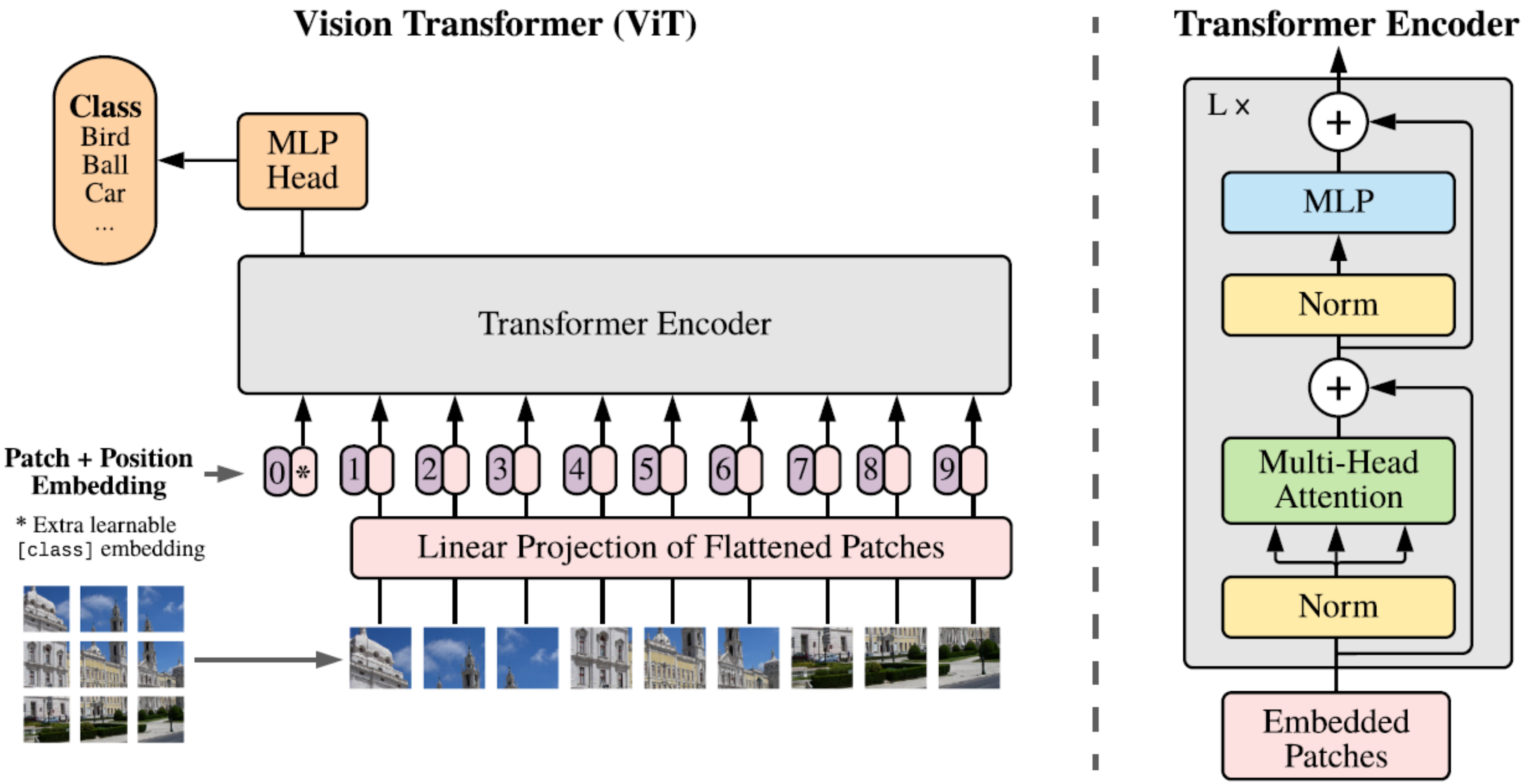
$$\mathbf{z}_0 = [\mathbf{x}_{\text{class}};\, \mathbf{x}_p^1\mathbf{E};\, \mathbf{x}_p^2\mathbf{E};\cdots;\, \mathbf{x}_p^N\mathbf{E}] + \mathbf{E}_{pos}$$

$$\mathbf{z'}_\ell = \mathrm{MSA}(\mathrm{LN}(\mathbf{z}_{\ell-1})) + \mathbf{z}_{\ell-1},$$

$$\mathbf{z}_\ell = \mathrm{MLP}(\mathrm{LN}(\mathbf{z'}_\ell)) + \mathbf{z'}_\ell,$$

$$\mathbf{y} = \mathrm{LN}(\mathbf{z}_L^0)$$

- AN IMAGE IS WORTH 16X16 WORDS: TRANSFORMERS FOR IMAGE RECOGNITION AT SCALE



Dosovitskiy et al., 2020

ViT results

Model	Layers	Hidden size D	MLP size	Heads	Params
ViT-Base	12	768	3072	12	86M
ViT-Large	24	1024	4096	16	307M
ViT-Huge	32	1280	5120	16	632M

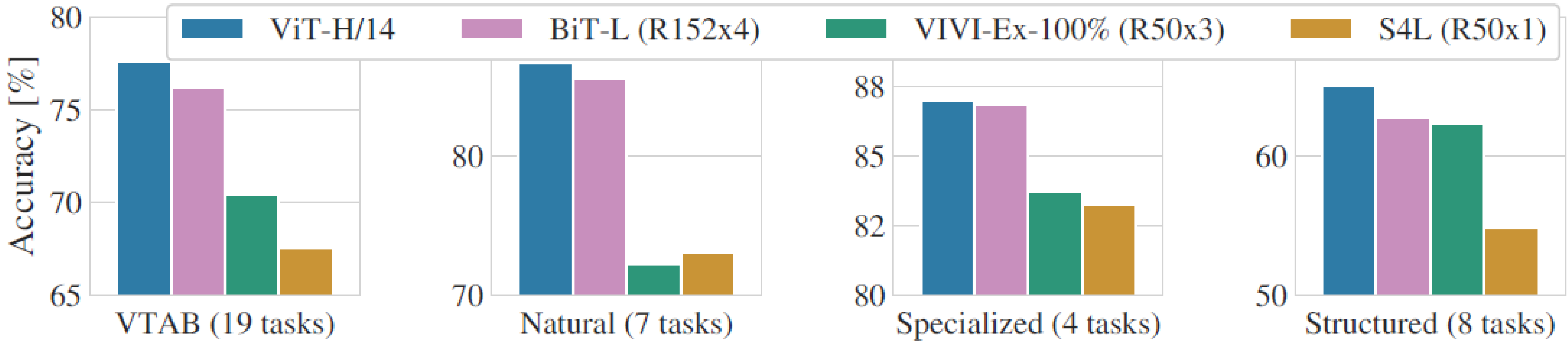
[Deng et al., 2009]

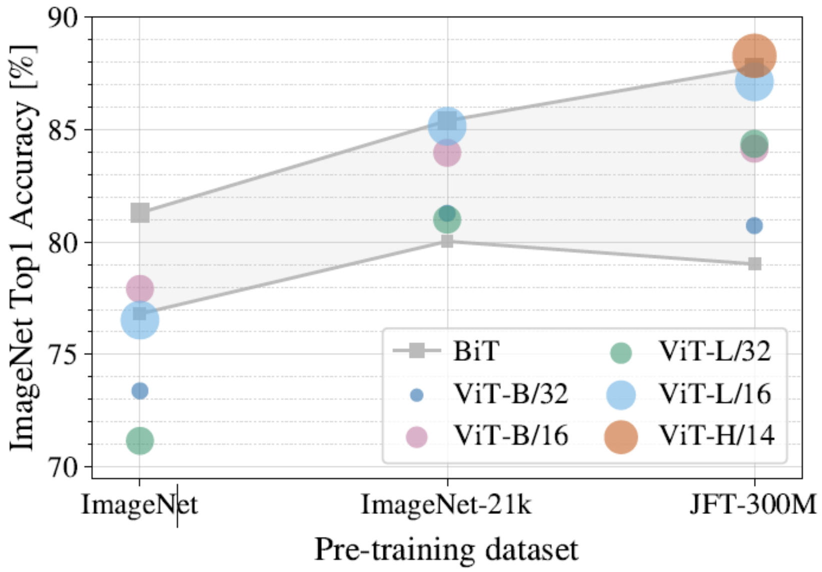
[Ridnik et al., 2021]

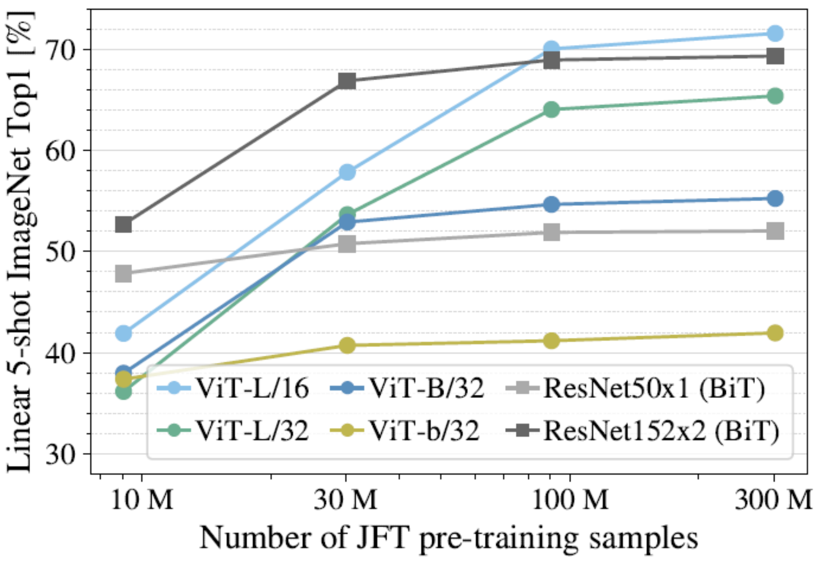
[Sun et al., 2017]

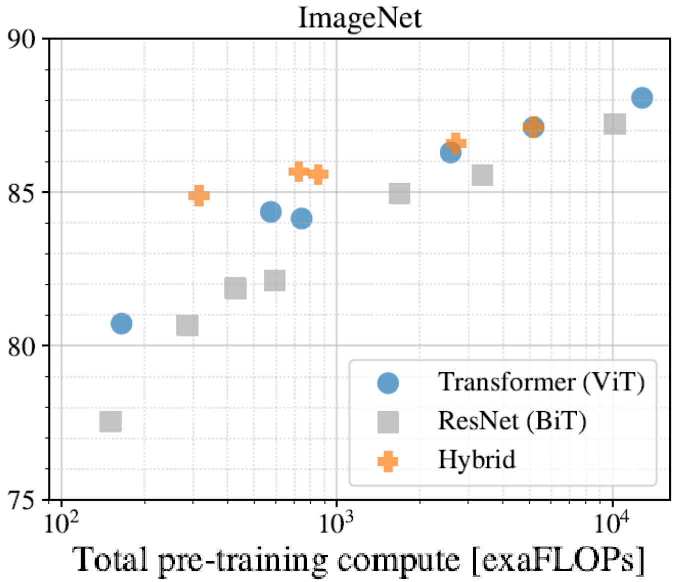
	Ours-JFT (ViT-H/14)	Ours-JFT (ViT-L/16)	Ours-I21K (ViT-L/16)	BiT-L (ResNet152x4)	Noisy Student (EfficientNet-L2)
ImageNet	**88.55** ± 0.04	87.76 ± 0.03	85.30 ± 0.02	87.54 ± 0.02	88.4/88.5*
ImageNet ReaL	**90.72** ± 0.05	90.54 ± 0.03	88.62 ± 0.05	90.54	90.55
CIFAR-10	**99.50** ± 0.06	99.42 ± 0.03	99.15 ± 0.03	99.37 ± 0.06	–
CIFAR-100	**94.55** ± 0.04	93.90 ± 0.05	93.25 ± 0.05	93.51 ± 0.08	–
Oxford-IIIT Pets	**97.56** ± 0.03	97.32 ± 0.11	94.67 ± 0.15	96.62 ± 0.23	–
Oxford Flowers-102	99.68 ± 0.02	**99.74** ± 0.00	99.61 ± 0.02	99.63 ± 0.03	–
VTAB (19 tasks)	**77.63** ± 0.23	76.28 ± 0.46	72.72 ± 0.21	76.29 ± 1.70	–
TPUv3-core-days	2.5k	0.68k	0.23k	9.9k	12.3k

ViT performance





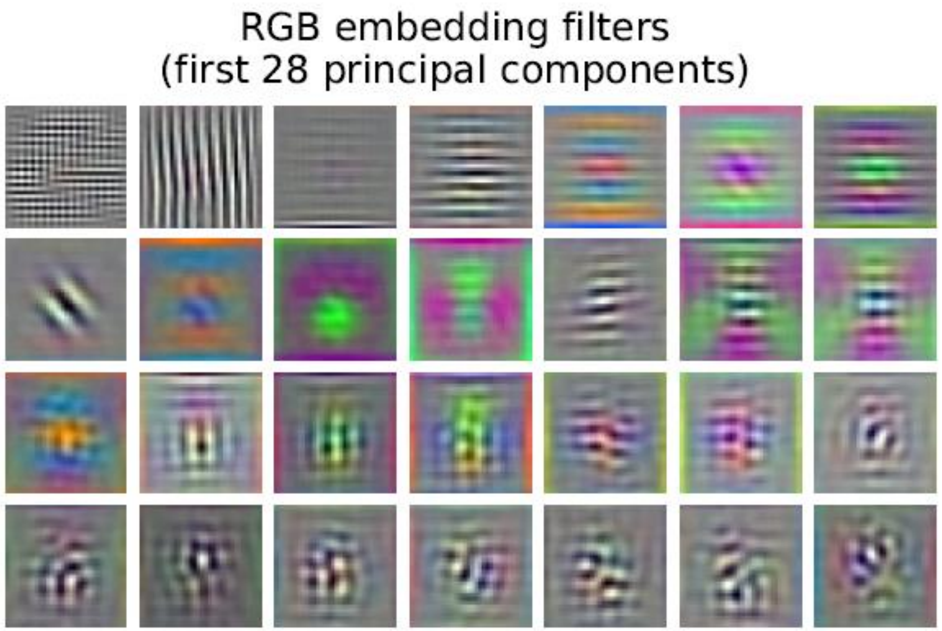


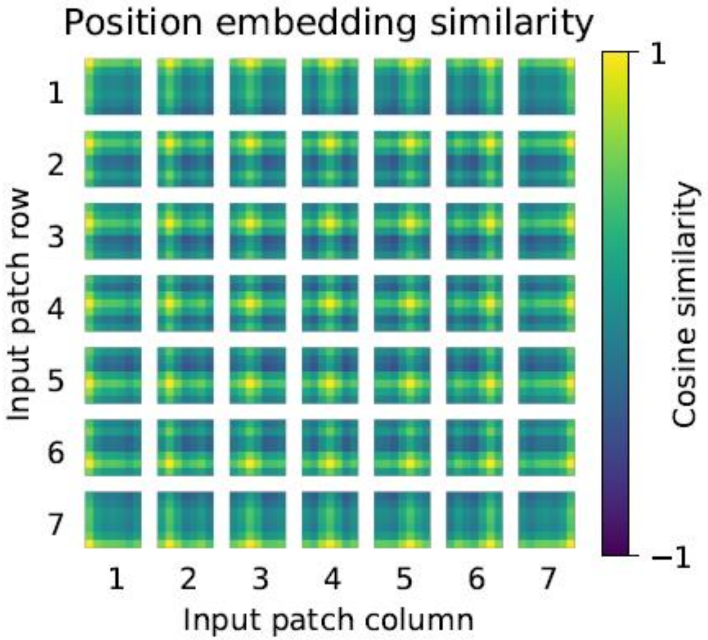


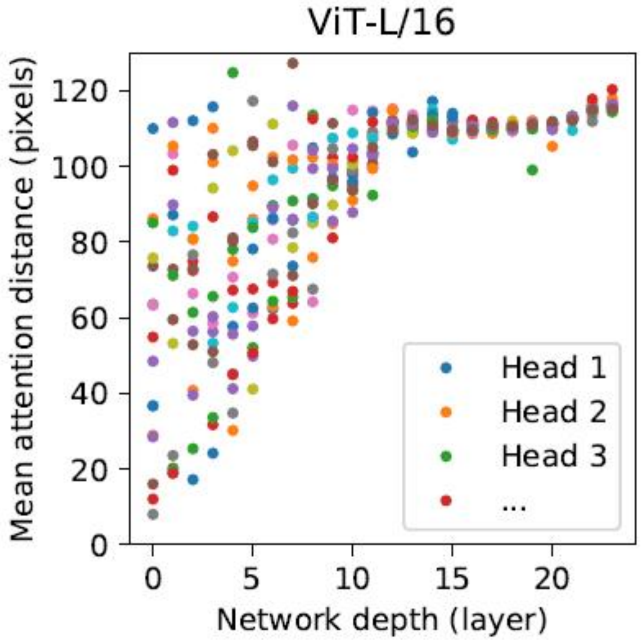
ViT details

- Initial linear embedding of RGB values
- Similarity of position embeddings
- Attention distance

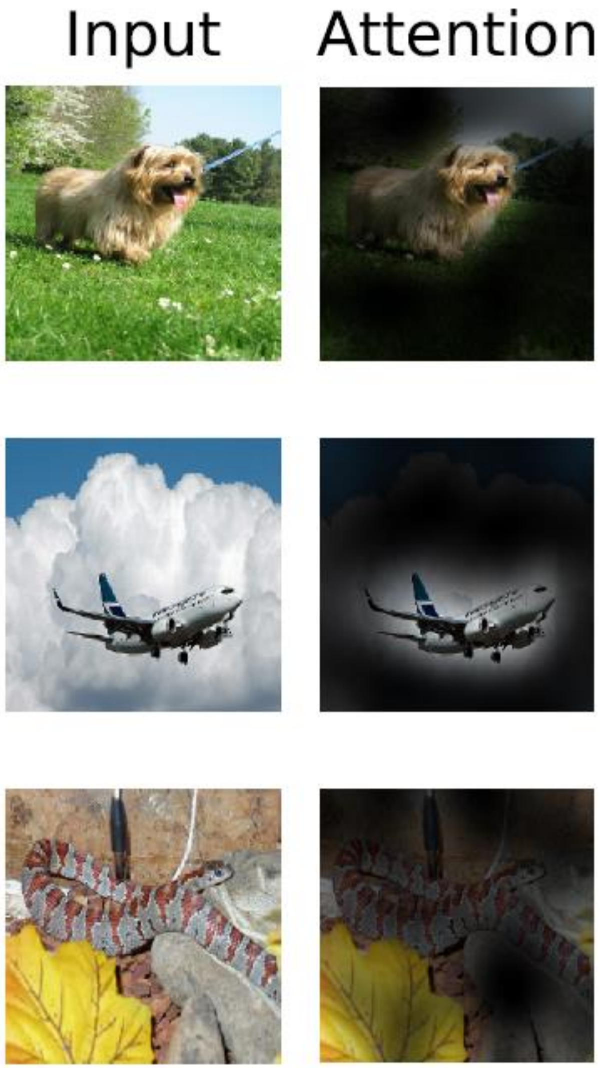
Pos. Emb.	Default/Stem	Every Layer	Every Layer-Shared
No Pos. Emb.	0.61382	N/A	N/A
1-D Pos. Emb.	0.64206	0.63964	0.64292
2-D Pos. Emb.	0.64001	0.64046	0.64022
Rel. Pos. Emb.	0.64032	N/A	N/A

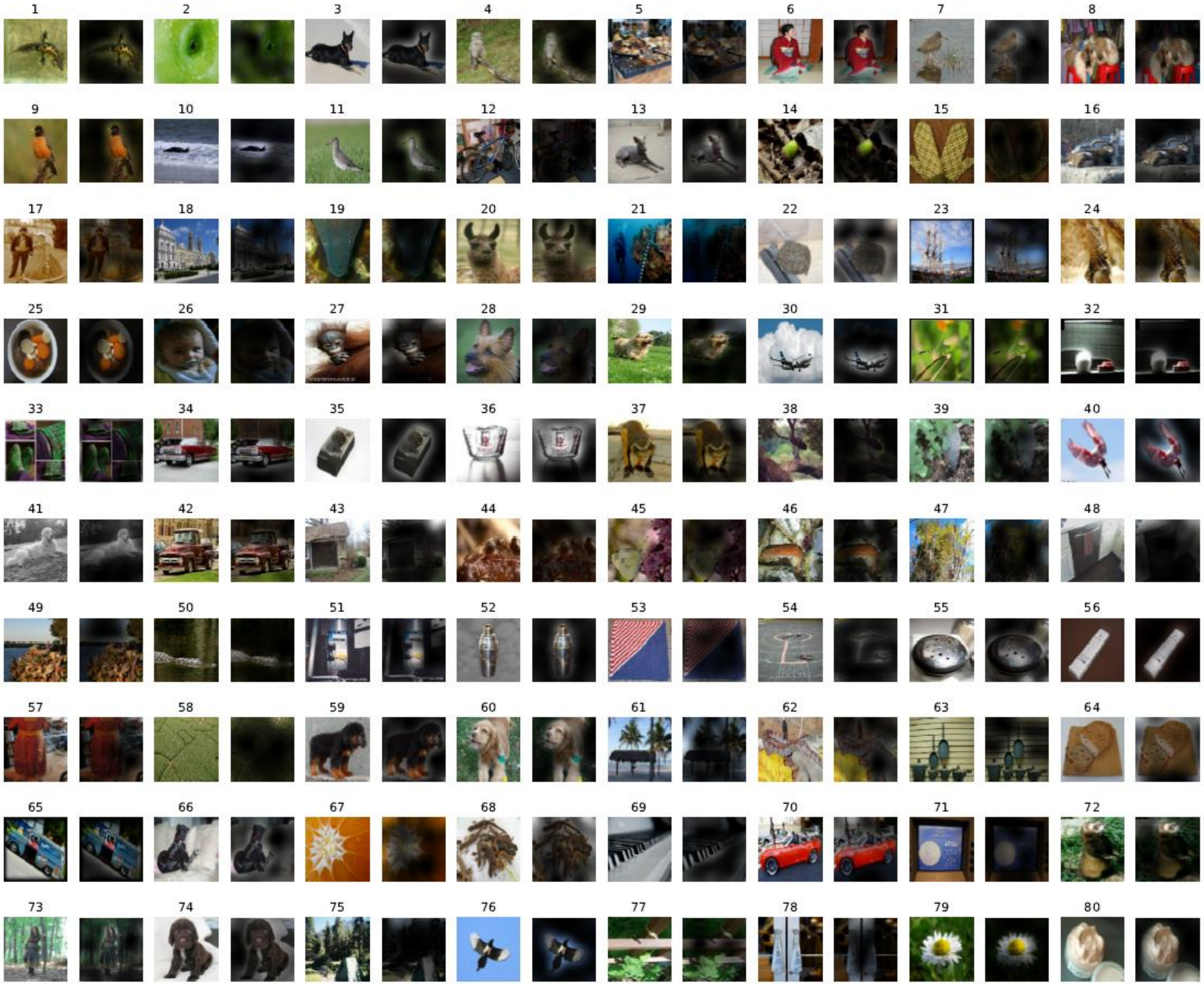






Attention maps





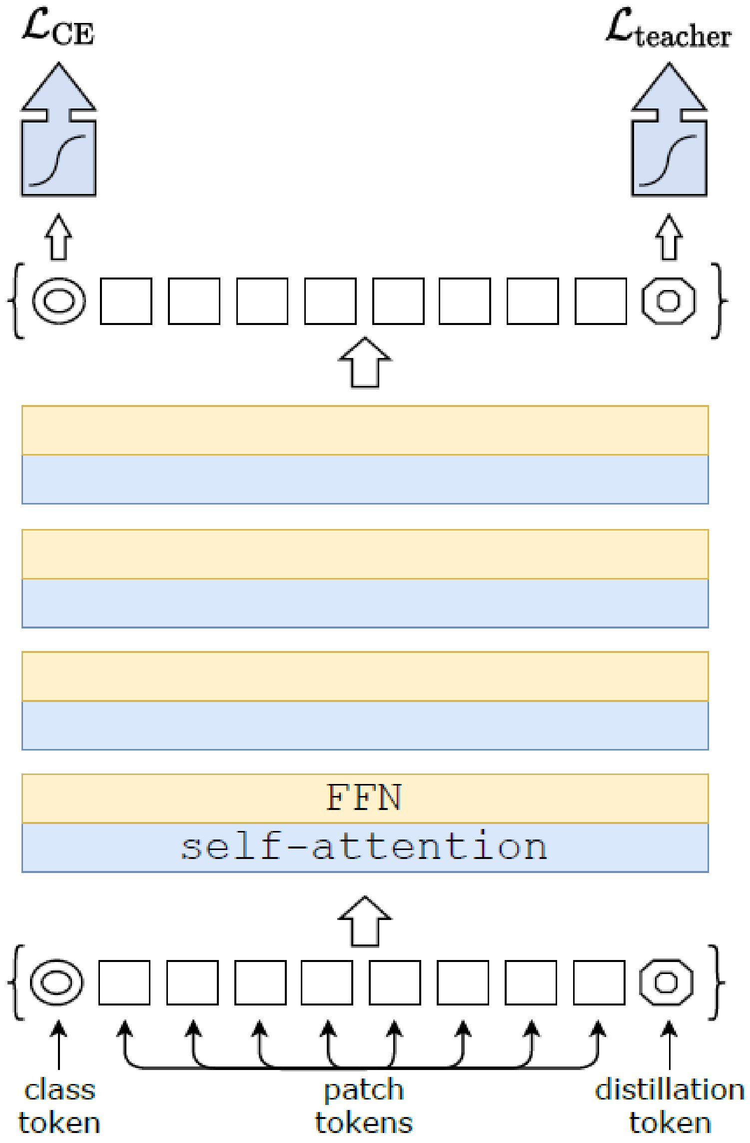
DeiT

Touvron et al., 2021

- DeiT - Training data-efficient image transformers & distillation through attention
- Soft and hard-label distillation
- Student-teacher architecture
  - CNN or Transformer-based teacher
- Distillation token
  - to reproduce the label predicted by the teacher
- Fine-tuning with distillation
- Classification with joint classifiers
- Trained on a single 8-GPU node in 2-3 days
- Imagenet as the sole training set
- Data and compute efficient!

$$\mathcal{L}_{\text{global}} = (1-\lambda)\mathcal{L}_{\text{CE}}(\psi(Z_{\text{s}}), y) + \lambda\tau^2 \text{KL}(\psi(Z_{\text{s}}/\tau), \psi(Z_{\text{t}}/\tau))$$

$$\mathcal{L}_{\text{global}}^{\text{hardDistill}} = \frac{1}{2}\mathcal{L}_{\text{CE}}(\psi(Z_s), y) + \frac{1}{2}\mathcal{L}_{\text{CE}}(\psi(Z_s), y_{\text{t}})$$



DeiT ablation study

method ↓	Supervision: label	Supervision: teacher	ImageNet top-1 (%): Ti 224	S 224	B 224	B↑384
DeiT– no distillation	✓	✗	72.2	79.8	81.8	83.1
DeiT– usual distillation	✗	soft	72.2	79.8	81.8	83.2
DeiT– hard distillation	✗	hard	74.3	80.9	83.0	84.0
DeiT⚗: class embedding	✓	hard	73.9	80.9	83.0	84.2
DeiT⚗: distil. embedding	✓	hard	74.6	81.1	83.1	84.4
DeiT⚗: class+distillation	✓	hard	74.5	81.2	83.4	84.5

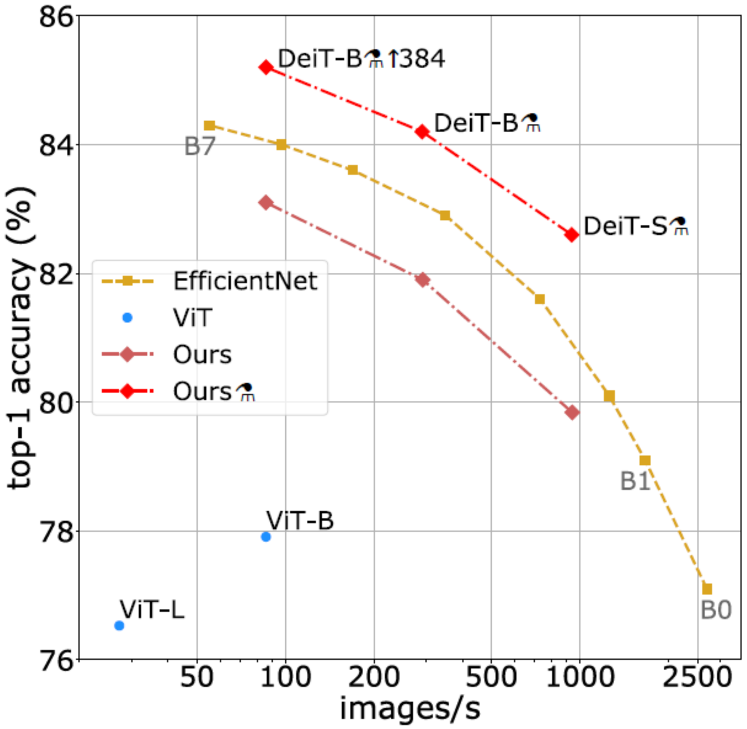
	groundtruth	no distillation: convnet	no distillation: DeiT	DeiT⚗ student (of the convnet): class	distillation	DeiT⚗
groundtruth	0.000	0.171	0.182	0.170	0.169	0.166
convnet (RegNetY)	0.171	0.000	0.133	0.112	0.100	0.102
DeiT	0.182	0.133	0.000	0.109	0.110	0.107
DeiT⚗– class only	0.170	0.112	0.109	0.000	0.050	0.033
DeiT⚗– distil. only	0.169	0.100	0.110	0.050	0.000	0.019
DeiT⚗– class+distil.	0.166	0.102	0.107	0.033	0.019	0.000

DeiT results

Teacher Models	Teacher acc.	Student: DeiT-B⚗ pretrain	Student: DeiT-B⚗ ↑384
DeiT-B	81.8	81.9	83.1
RegNetY-4GF	80.0	82.7	83.6
RegNetY-8GF	81.7	82.7	83.8
RegNetY-12GF	82.4	83.1	84.1
RegNetY-16GF	82.9	83.1	84.2

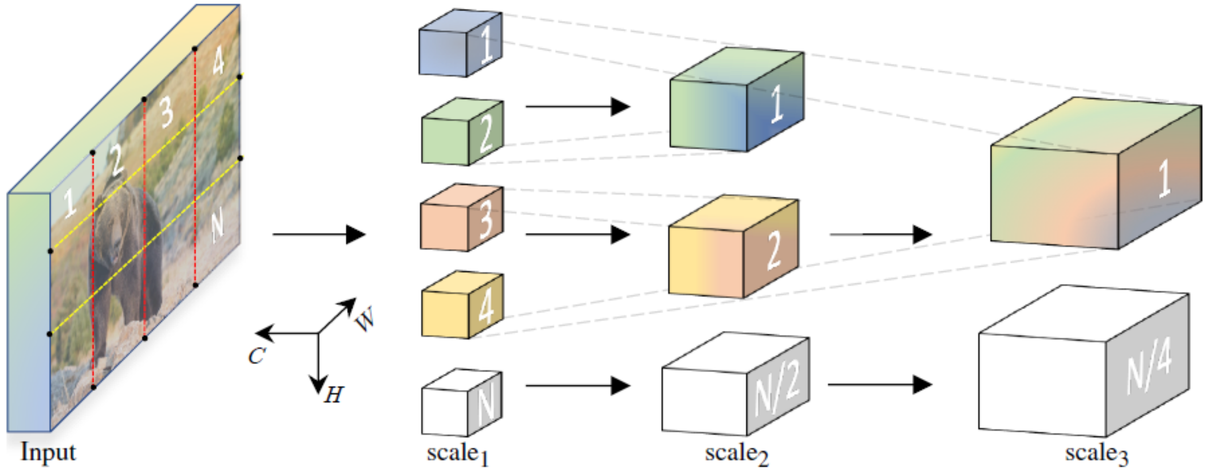
Model	ViT model	embedding dimension	#heads	#layers	#params
DeiT-Ti	N/A	192	3	12	5M
DeiT-S	N/A	384	6	12	22M
DeiT-B	ViT-B	768	12	12	86M

Model	ImageNet	CIFAR-10	CIFAR-100	Flowers	Cars	iNat-18	iNat-19	im/sec
Grafit ResNet-50 [49]	79.6	–	–	98.2	92.5	69.8	75.9	1226.1
Grafit RegNetY-8GF [49]	–	–	–	99.0	94.0	76.8	80.0	591.6
ResNet-152 [10]	–	–	–	–	–	69.1	–	526.3
EfficientNet-B7 [48]	84.3	98.9	91.7	98.8	94.7	–	–	55.1
ViT-B/32 [15]	73.4	97.8	86.3	85.4	–	–	–	394.5
ViT-B/16 [15]	77.9	98.1	87.1	89.5	–	–	–	85.9
ViT-L/32 [15]	71.2	97.9	87.1	86.4	–	–	–	124.1
ViT-L/16 [15]	76.5	97.9	86.4	89.7	–	–	–	27.3
DeiT-B	81.8	99.1	90.8	98.4	92.1	73.2	77.7	292.3
DeiT-B↑384	83.1	99.1	90.8	98.5	93.3	79.5	81.4	85.9
DeiT-B⚗	83.4	99.1	91.3	98.8	92.9	73.7	78.4	290.9
DeiT-B⚗ ↑384	84.4	99.2	91.4	98.9	93.9	80.1	83.0	85.9

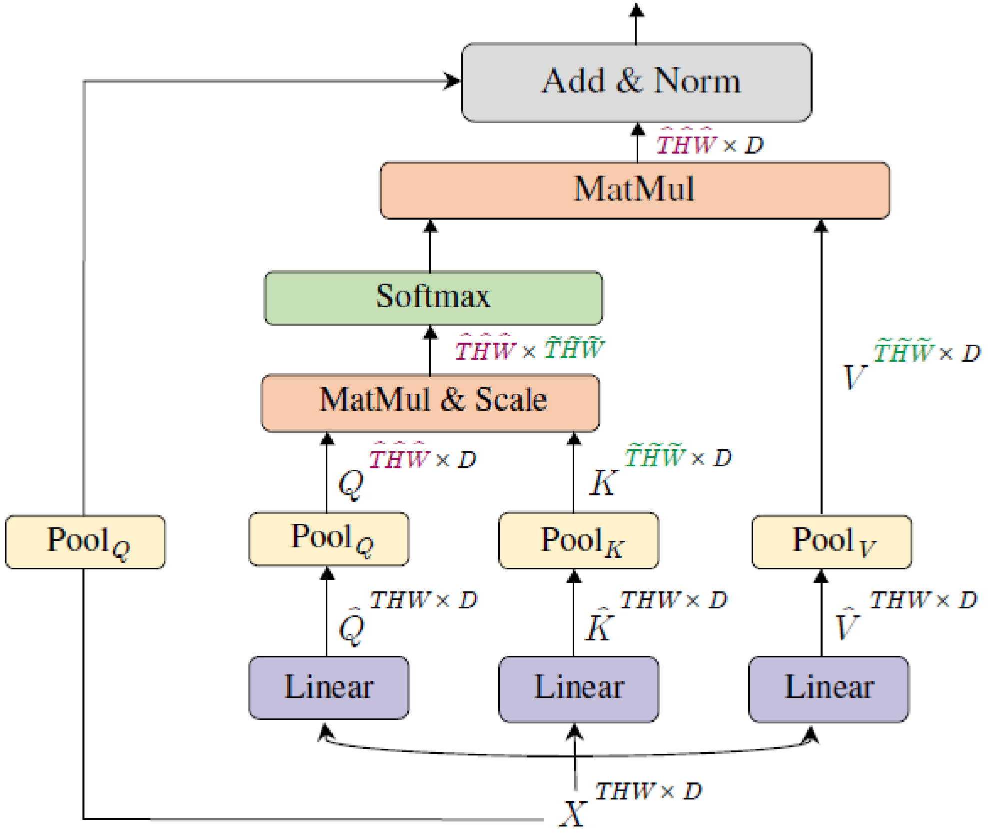


MVIT - Multiscale Vision Transformers

- Several channel-resolution 'scale' stages
- From image resolution and small channel dimension to reduced resolution and expanded channel capacity
- Pooling attention



Fan et al., 2021



MViT

ViT

stage	operators	output sizes
data	stride $8\times1\times1$	$8\times224\times224$
patch_1	$1\times16\times16$, 768 stride $1\times16\times16$	$768\times8\times14\times14$
scale_2	$\begin{bmatrix}\text{MHA}(768)\\ \text{MLP}(3072)\end{bmatrix}\times12$	$768\times8\times14\times14$

179.6G FLOPS
87.2M param
68.5% top1 acc.

MViT

stage	operators	output sizes
data	stride $4\times1\times1$	$16\times224\times224$
cube_1	$3\times7\times7$, 96 stride $2\times4\times4$	$96\times8\times56\times56$
scale_2	$\begin{bmatrix}\text{MHPA}(96)\\ \text{MLP}(384)\end{bmatrix}\times1$	$96\times8\times56\times56$
scale_3	$\begin{bmatrix}\text{MHPA}(192)\\ \text{MLP}(768)\end{bmatrix}\times2$	$192\times8\times28\times28$
scale_4	$\begin{bmatrix}\text{MHPA}(384)\\ \text{MLP}(1536)\end{bmatrix}\times11$	$384\times8\times14\times14$
scale_5	$\begin{bmatrix}\text{MHPA}(768)\\ \text{MLP}(3072)\end{bmatrix}\times2$	$768\times8\times7\times7$

70.5G FLOPS
36.5M param
77.2% top1 acc.

MViT results

Video recognition on Kinetics-400

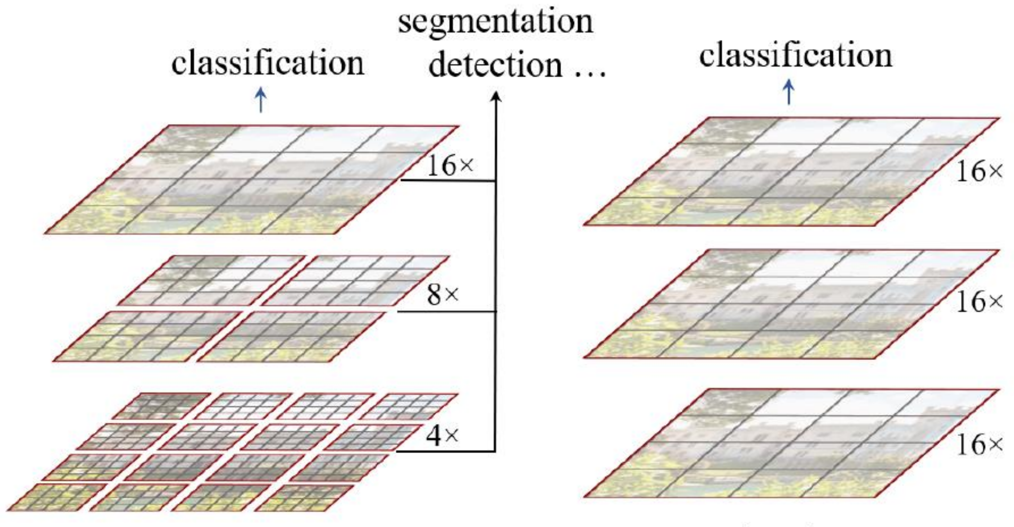
model	pre-train	top-1	top-5	FLOPs×views	Param
Two-Stream I3D [11]	-	71.6	90.0	216 × NA	25.0
ip-CSN-152 [96]	-	77.8	92.8	109×3×10	32.8
SlowFast 8×8 +NL [30]	-	78.7	93.5	116×3×10	59.9
SlowFast 16×8 +NL [30]	-	79.8	93.9	234×3×10	59.9
X3D-M [29]	-	76.0	92.3	6.2×3×10	3.8
X3D-XL [29]	-	79.1	93.9	48.4×3×10	11.0
ViT-B-VTN [78]	ImageNet-1K	75.6	92.4	4218×1×1	114.0
ViT-B-VTN [78]	ImageNet-**21K**	78.6	93.7	4218×1×1	114.0
ViT-B-TimeSformer [6]	ImageNet-**21K**	80.7	94.7	2380×3×1	121.4
ViT-L-ViViT [1]	ImageNet-**21K**	81.3	94.7	3992×3×4	310.8
ViT-B (our baseline)	ImageNet-**21K**	79.3	93.9	180×1×5	87.2
ViT-B (our baseline)	-	68.5	86.9	180×1×5	87.2
MViT-S	-	76.0	92.1	32.9×1×5	26.1
MViT-B, 16×4	-	78.4	93.5	70.5×1×5	36.6
MViT-B, 32×3	-	80.2	94.4	170×1×5	36.6
MViT-B, 64×3	-	**81.2**	**95.1**	455×3×3	36.6

Image recognition on ImageNet

model	Acc	FLOPs (G)	Param (M)
RegNetZ-4GF [24]	83.1	4.0	28.1
RegNetZ-16GF [24]	84.1	15.9	95.3
EfficientNet-B7 [93]	84.3	37.0	66.0
DeiT-S [95]	79.8	4.6	22.1
DeiT-B [95]	81.8	17.6	86.6
DeiT-B ↑ 384^2 [95]	83.1	55.5	87.0
MViT-B-16, max-pool	82.5	7.8	37.0
MViT-B-24, max-pool	83.1	10.9	53.5
MViT-B-24-wide-320^2, max-pool	84.3	32.7	72.9
MViT-B-16	83.0	7.8	37.0
MViT-B-24-wide-320^2	**84.8**	32.7	72.9

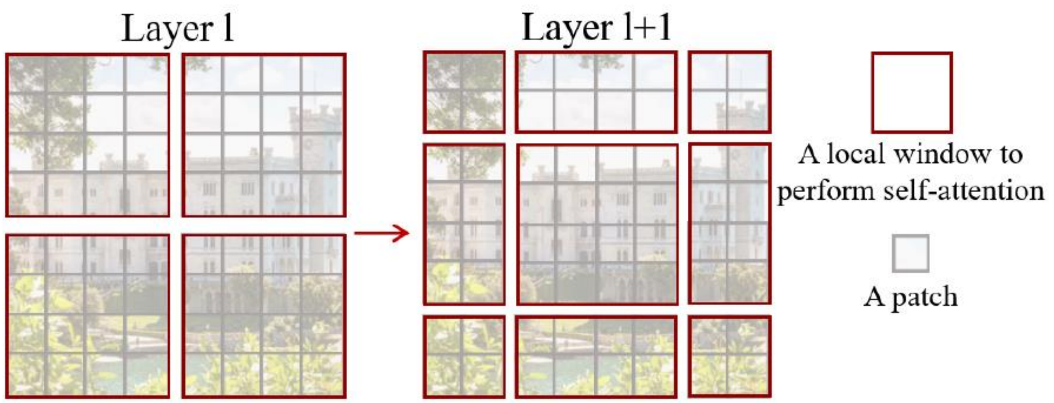
Swin Transformer

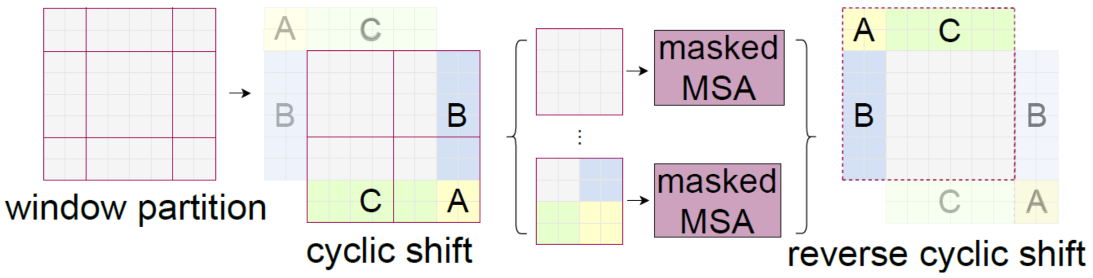
- Hierarchical Vision Transformer using Shifted Windows
- General purpose transformer backbone
- Hierarchical feature maps
- Shift of the window partition between consecutive self-attention layers



(a) Swin Transformer (ours)

(b) ViT

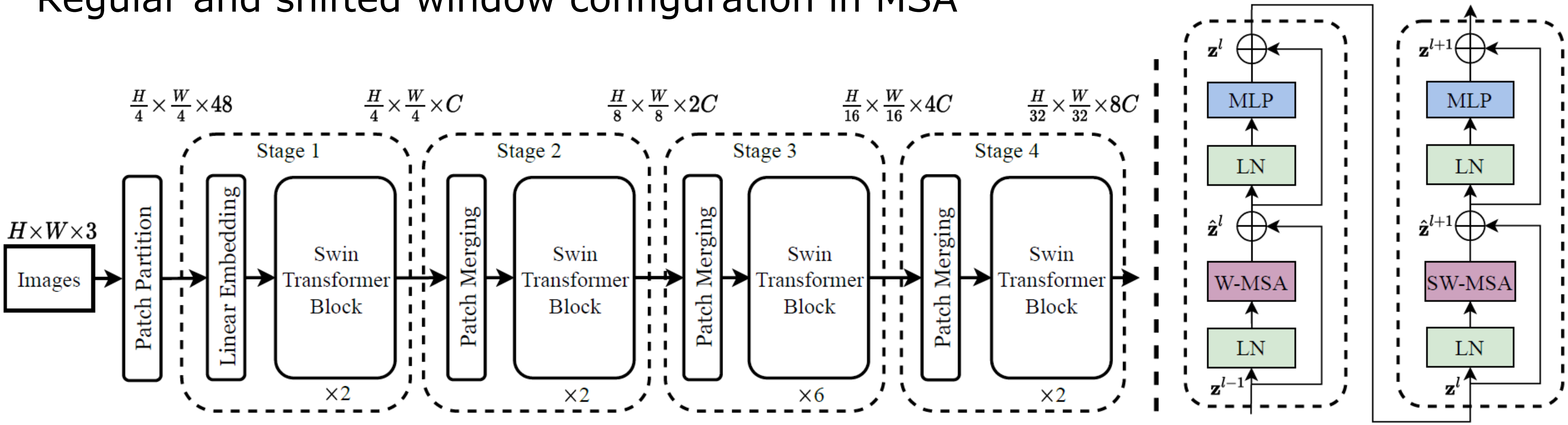




Liu et al., 2021

Swin architecture

- Patch merging
- Regular and shifted window configuration in MSA



(a) Architecture

(b) Two Successive Swin Transformer Blocks

$$\hat{\mathbf{z}}^l = \text{W-MSA}\left(\text{LN}\left(\mathbf{z}^{l-1}\right)\right) + \mathbf{z}^{l-1}$$
$$\mathbf{z}^l = \text{MLP}\left(\text{LN}\left(\hat{\mathbf{z}}^l\right)\right) + \hat{\mathbf{z}}^l$$
$$\hat{\mathbf{z}}^{l+1} = \text{SW-MSA}\left(\text{LN}\left(\mathbf{z}^l\right)\right) + \mathbf{z}^l$$
$$\mathbf{z}^{l+1} = \text{MLP}\left(\text{LN}\left(\hat{\mathbf{z}}^{l+1}\right)\right) + \hat{\mathbf{z}}^{l+1}$$

Swin-T: $C = 96$, layer numbers = $\{2, 2, 6, 2\}$

Swin-S: $C = 96$, layer numbers =$\{2, 2, 18, 2\}$

Swin-B: $C = 128$, layer numbers =$\{2, 2, 18, 2\}$

Swin-L: $C = 192$, layer numbers =$\{2, 2, 18, 2\}$

$$\Omega(\text{MSA}) = 4hwC^2 + 2(hw)^2C,$$
$$\Omega(\text{W-MSA}) = 4hwC^2 + 2M^2hwC$$

Swin results

(a) Regular ImageNet-1K trained models

method	image size	#param.	FLOPs	throughput (image / s)	ImageNet top-1 acc.
RegNetY-4G [47]	224^2	21M	4.0G	1156.7	80.0
RegNetY-8G [47]	224^2	39M	8.0G	591.6	81.7
RegNetY-16G [47]	224^2	84M	16.0G	334.7	82.9
EffNet-B3 [57]	300^2	12M	1.8G	732.1	81.6
EffNet-B4 [57]	380^2	19M	4.2G	349.4	82.9
EffNet-B5 [57]	456^2	30M	9.9G	169.1	83.6
EffNet-B6 [57]	528^2	43M	19.0G	96.9	84.0
EffNet-B7 [57]	600^2	66M	37.0G	55.1	84.3
ViT-B/16 [19]	384^2	86M	55.4G	85.9	77.9
ViT-L/16 [19]	384^2	307M	190.7G	27.3	76.5
DeiT-S [60]	224^2	22M	4.6G	940.4	79.8
DeiT-B [60]	224^2	86M	17.5G	292.3	81.8
DeiT-B [60]	384^2	86M	55.4G	85.9	83.1
Swin-T	224^2	29M	4.5G	755.2	81.3
Swin-S	224^2	50M	8.7G	436.9	83.0
Swin-B	224^2	88M	15.4G	278.1	83.3
Swin-B	384^2	88M	47.0G	84.7	84.2

(b) ImageNet-22K pre-trained models

method	image size	#param.	FLOPs	throughput (image / s)	ImageNet top-1 acc.
R-101x3 [37]	384^2	388M	204.6G	-	84.4
R-152x4 [37]	480^2	937M	840.5G	-	85.4
ViT-B/16 [19]	384^2	86M	55.4G	85.9	84.0
ViT-L/16 [19]	384^2	307M	190.7G	27.3	85.2
Swin-B	224^2	88M	15.4G	278.1	85.2
Swin-B	384^2	88M	47.0G	84.7	86.0
Swin-L	384^2	197M	103.9G	42.1	86.4

Swin results

(b) Various backbones w. Cascade Mask R-CNN

	AP^{box}	AP^{box}_{50}	AP^{box}_{75}	AP^{mask}	AP^{mask}_{50}	AP^{mask}_{75}	param	FLOPs	FPS
DeiT-S†	48.0	67.2	51.7	41.4	64.2	44.3	80M	889G	10.4
R50	46.3	64.3	50.5	40.1	61.7	43.4	82M	739G	18.0
Swin-T	**50.5**	**69.3**	**54.9**	**43.7**	**66.6**	**47.1**	86M	745G	15.3
X101-32	48.1	66.5	52.4	41.6	63.9	45.2	101M	819G	12.8
Swin-S	**51.8**	**70.4**	**56.3**	**44.7**	**67.9**	**48.5**	107M	838G	12.0
X101-64	48.3	66.4	52.3	41.7	64.0	45.1	140M	972G	10.4
Swin-B	**51.9**	**70.9**	**56.5**	**45.0**	**68.4**	**48.7**	145M	982G	11.6

(c) System-level Comparison

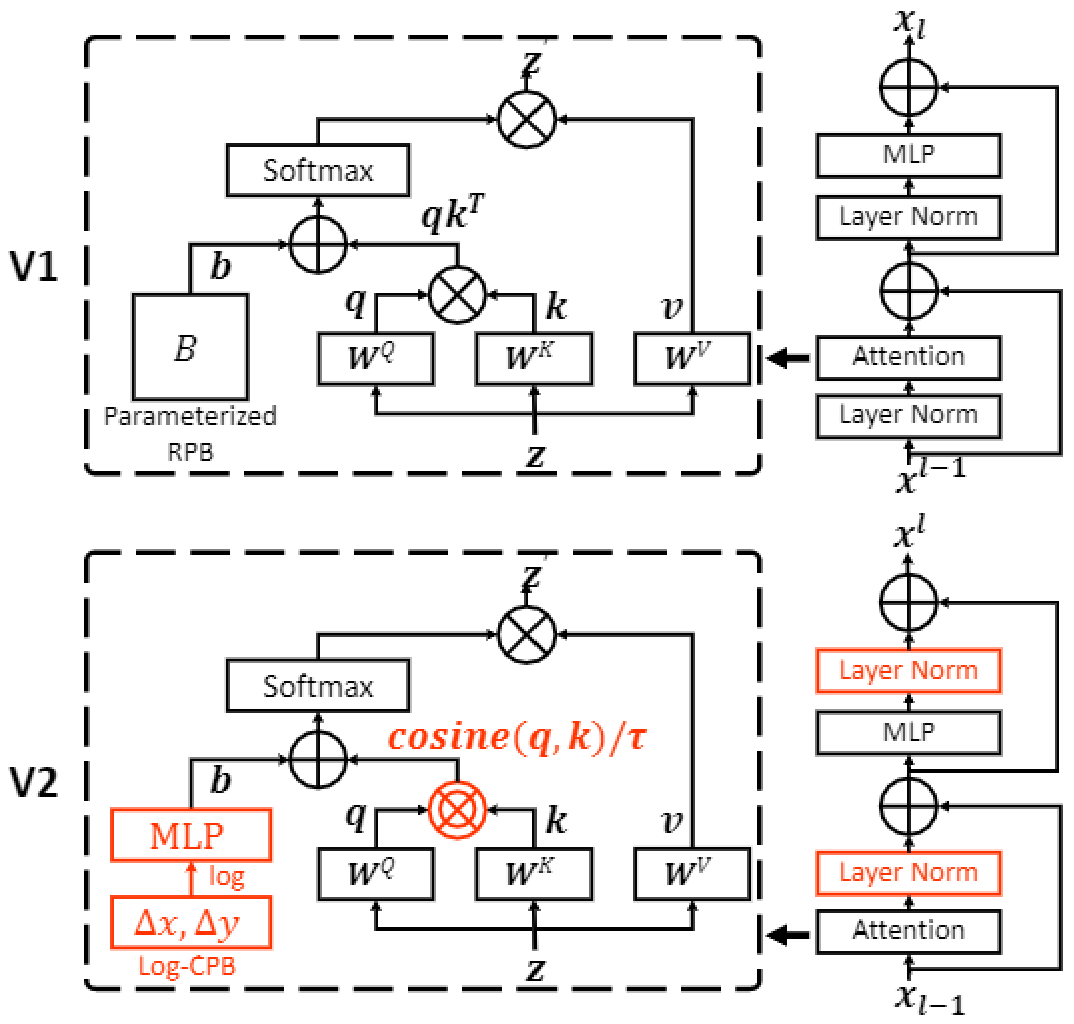
Method	mini-val		test-dev		#param.	FLOPs
	AP^{box}	AP^{mask}	AP^{box}	AP^{mask}		
RepPointsV2* [11]	-	-	52.1	-	-	-
GCNet* [6]	51.8	44.7	52.3	45.4	-	1041G
RelationNet++* [12]	-	-	52.7	-	-	-
SpineNet-190 [20]	52.6	-	52.8	-	164M	1885G
ResNeSt-200* [75]	52.5	-	53.3	47.1	-	-
EfficientDet-D7 [58]	54.4	-	55.1	-	77M	410G
DetectoRS* [45]	-	-	55.7	48.5	-	-
YOLOv4 P7* [3]	-	-	55.8	-	-	-
Copy-paste [25]	55.9	47.2	56.0	47.4	185M	1440G
X101-64 (HTC++)	52.3	46.0	-	-	155M	1033G
Swin-B (HTC++)	56.4	49.1	-	-	160M	1043G
Swin-L (HTC++)	57.1	49.5	57.7	50.2	284M	1470G
Swin-L (HTC++)*	**58.0**	**50.4**	**58.7**	**51.1**	284M	-

ADE20K Method	Backbone	val mIoU	test score	#param.	FLOPs	FPS
DANet [22]	ResNet-101	45.2	-	69M	1119G	15.2
DLab.v3+ [10]	ResNet-101	44.1	-	63M	1021G	16.0
ACNet [23]	ResNet-101	45.9	38.5	-		
DNL [68]	ResNet-101	46.0	56.2	69M	1249G	14.8
OCRNet [70]	ResNet-101	45.3	56.0	56M	923G	19.3
UperNet [66]	ResNet-101	44.9	-	86M	1029G	20.1
OCRNet [70]	HRNet-w48	45.7	-	71M	664G	12.5
DLab.v3+ [10]	ResNeSt-101	46.9	55.1	66M	1051G	11.9
DLab.v3+ [10]	ResNeSt-200	48.4	-	88M	1381G	8.1
SETR [78]	T-Large‡	50.3	61.7	308M	-	-
UperNet	DeiT-S†	44.0	-	52M	1099G	16.2
UperNet	Swin-T	46.1	-	60M	945G	18.5
UperNet	Swin-S	49.3	-	81M	1038G	15.2
UperNet	Swin-B‡	51.6	-	121M	1841G	8.7
UperNet	Swin-L‡	**53.5**	**62.8**	234M	3230G	6.2

Swin as a backbone architecture
- COCO object detection
- ADE20K semantic segmentation

SwinV2

- Swin Transformer V2: Scaling Up Capacity and Resolution
- A res-post-norm to replace the previous pre-norm configuration
- A scaled cosine attention to replace the original dot product attention
- A log-spaced continuous relative position bias approach to replace the previous parameterized approach
- Self-supervised pre-training method, SimMIM, to reduce the needs of vast labeled images
- Several tricks for memory efficiency
- Up to 3B parameters
- Up to 1,536x1,536 image resolution



[Liu et al., 2022] [Xie et al., 2021]

SwinV2 results

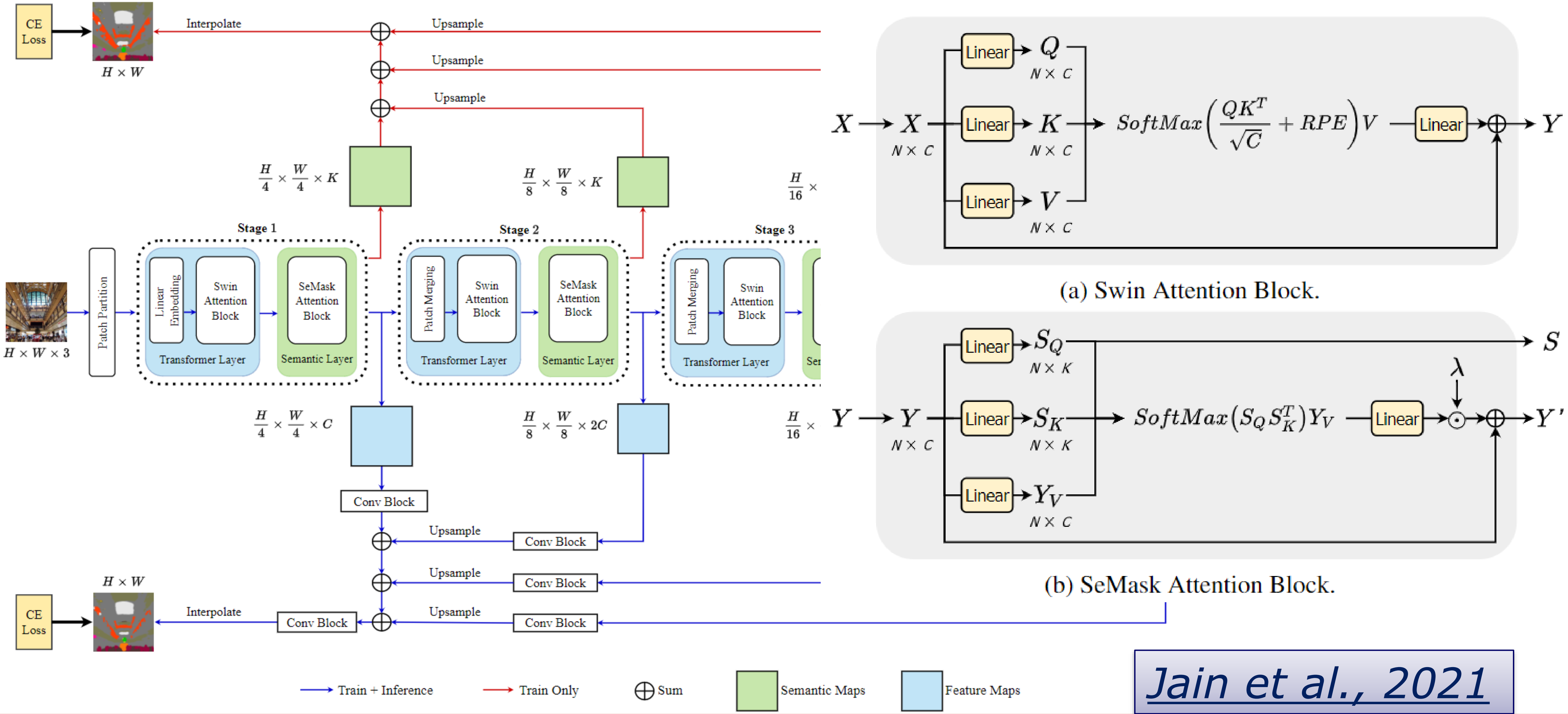
Method	param	pre-train images	pre-train length (#im)	pre-train im size	pre-train time	fine-tune im size	ImageNet-1K-V1 top-1 acc	ImaegNet-1K-V2 top-1 acc
SwinV1-B	88M	IN-22K-14M	1.3B	224^2	$<30^\dagger$	384^2	86.4	76.58
SwinV1-L	197M	IN-22K-14M	1.3B	224^2	$<10^\dagger$	384^2	87.3	77.46
ViT-G [80]	1.8B	JFT-3B	164B	224^2	>30k	518^2	90.45	83.33
V-MoE [56]	14.7B*	JFT-3B	-	224^2	16.8k	518^2	90.35	-
CoAtNet-7 [17]	2.44B	JFT-3B	-	224^2	20.1k	512^2	**90.88**	-
SwinV2-B	88M	IN-22K-14M	1.3B	192^2	$<30^\dagger$	384^2	87.1	78.08
SwinV2-L	197M	IN-22K-14M	1.3B	192^2	$<20^\dagger$	384^2	87.7	78.31
SwinV2-G	3.0B	IN-22K-ext-70M	3.5B	192^2	$<0.5k^\dagger$	640^2	90.17	**84.00**

Method	train I(W) size	test I(W) size	mIoU
SwinV1-L [46]	640(7)	640(7)	53.5*
Focal-L [75]	640(40)	640(40)	55.4*
CSwin-L [21]	640(40)	640(40)	55.7*
MaskFormer [13]	640(7)	640(7)	55.6*
FaPN [33]	640(7)	640(7)	56.7*
BEiT [4]	640(40)	640(40)	58.4*
SwinV2-L (UperNet)	640(40)	640(40)	55.9*
SwinV2-G (UperNet)	640(40)	640(40)	59.1
		896 (56)	59.3
		896 (56)	**59.9***

Method	train I(W) size	test I(W) size	mini-val (AP) box	mini-val (AP) mask	test-dev (AP) box	test-dev (AP) mask
CopyPaste [25]	1280(-)	1280(-)	57.0	48.9	57.3	49.1
SwinV1-L [46]	800(7)	ms(7)	58.0	50.4	58.7	51.1
YOLOR [66]	1280(-)	1280(-)	-	-	57.3	-
CBNet [43]	1400(7)	ms(7)	59.6	51.8	60.1	52.3
DyHead [16]	1200(-)	ms(-)	60.3	-	60.6	-
SoftTeacher [74]	1280(12)	ms(12)	60.7	52.5	61.3	53.0
SwinV2-L (HTC++)	1536(32)	1100(32)	58.8	51.1	-	-
		1100 (48)	58.9	51.2	-	-
		ms (48)	60.2	52.1	60.8	52.7
SwinV2-G (HTC++)	1536(32)	1100(32)	61.7	53.3	-	-
		1100 (48)	61.9	53.4	-	-
		ms (48)	**62.5**	**53.7**	**63.1**	**54.4**

SeMask

- Semantically Masked Transformers for Semantic Segmentation

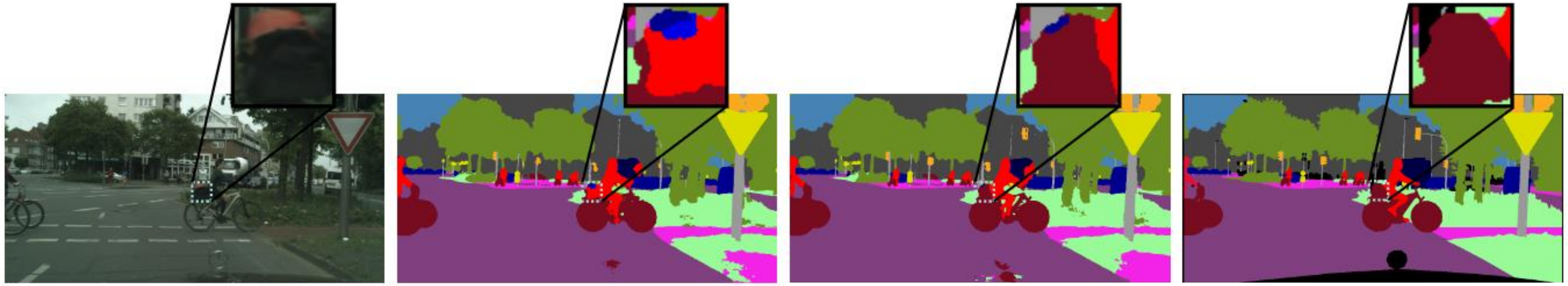


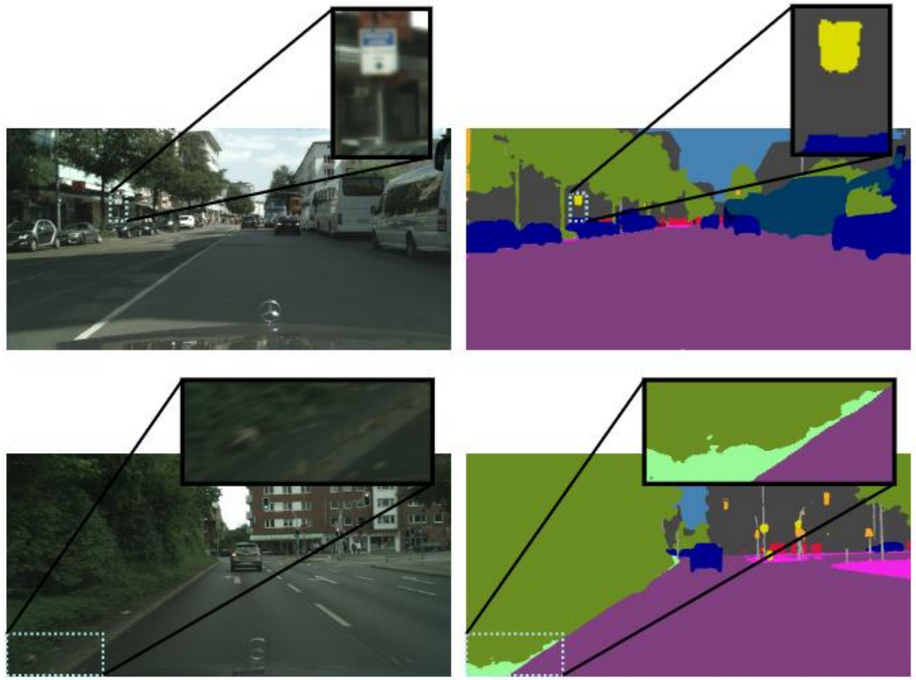
(a) Swin Attention Block.

(b) SeMask Attention Block.

[Jain et al., 2021](#)

SeMask results

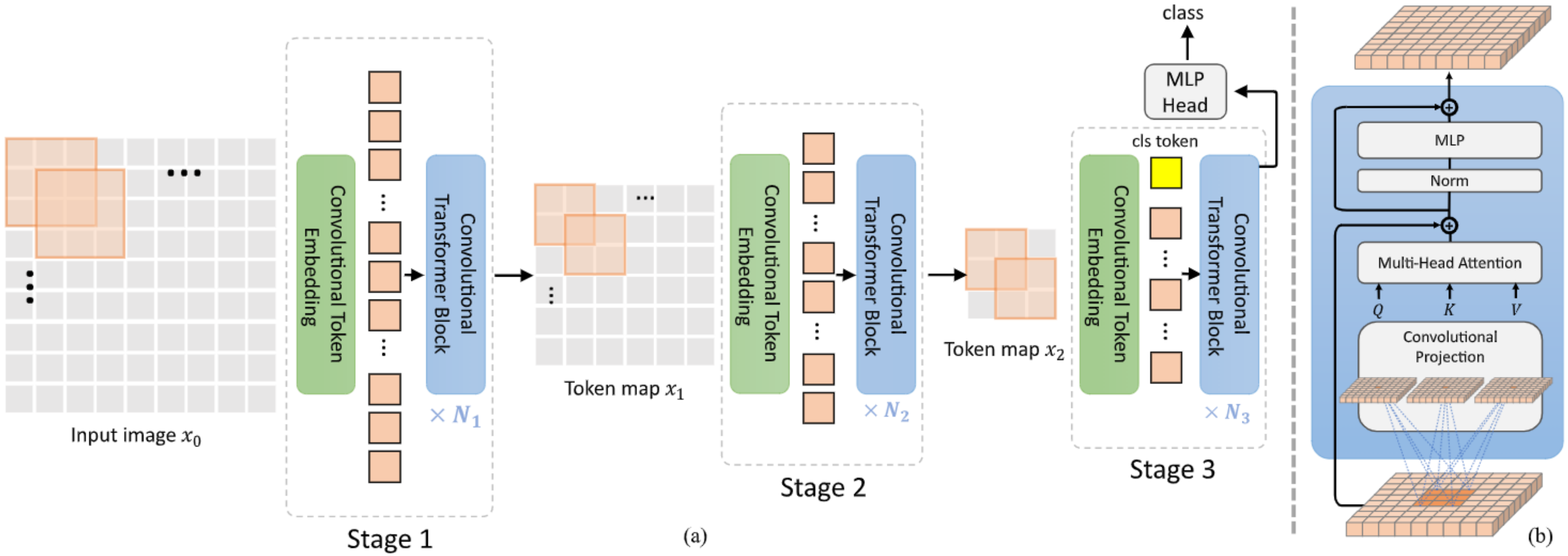




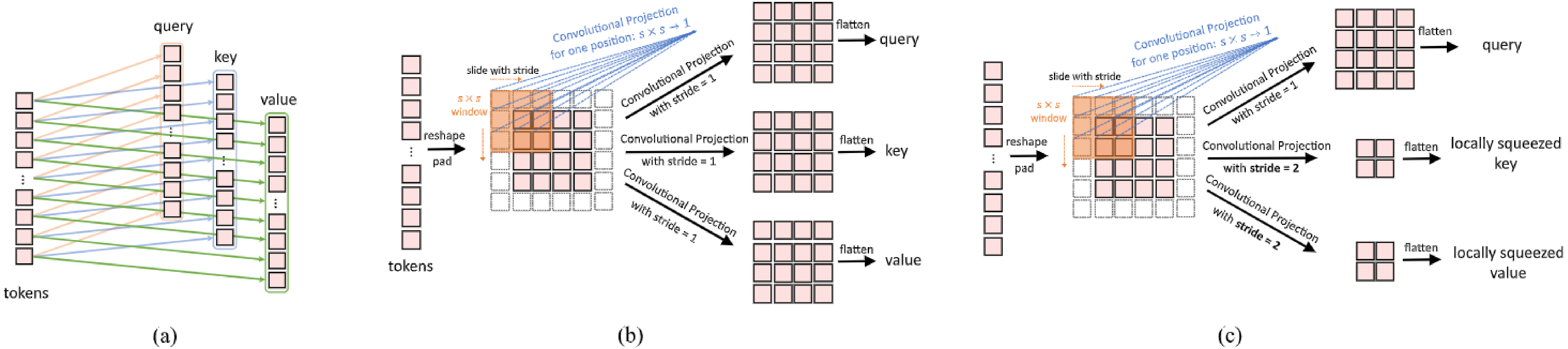
Method	Backbone	mIoU (%)	MS mIoU (%)
CNN Backbones			
PSANet [55]	ResNet-101	77.94	79.05
DeepLabV3+ [7]	Xception-71	-	79.55
CCNet [25]	ResNet-101	80.50	81.30
Transformer Backbones			
Seg-L-Mask/16 [44]	ViT-L/16†	79.10	81.30
Swin-L FPN [37]	Swin-L†	78.03	79.53
MaskFormer [9]	ResNet-101	78.50	80.30
Mask2Former [8]	Swin-L†	83.30	84.30
HRNetV2-OCR+PSA [35]	HRNetV2-W48†	-	**86.95**
SeMask-L FPN *(Ours)*	SeMask Swin-L†	78.53	80.39
SeMask-L Mask2Former *(Ours)*	SeMask Swin-L†	**83.97**	**84.98**

(a) Image (b) Swin-T FPN (c) Ours (d) Ground Truth

CvT: Introducing Convolutions to Vision Transformers



Wu et al., 2021



CvT

Method Type	Network	#Param. (M)	image size	FLOPs (G)	ImageNet top-1 (%)	Real top-1 (%)	V2 top-1 (%)
Convolutional Networks	ResNet-50 [15]	25	224^2	4.1	76.2	82.5	63.3
	ResNet-101 [15]	45	224^2	7.9	77.4	83.7	65.7
	ResNet-152 [15]	60	224^2	11	78.3	84.1	67.0
Transformers	ViT-B/16 [11]	86	384^2	55.5	77.9	83.6	–
	ViT-L/16 [11]	307	384^2	191.1	76.5	82.2	–
	DeiT-S [30][arxiv 2020]	22	224^2	4.6	79.8	85.7	68.5
	DeiT-B [30][arxiv 2020]	86	224^2	17.6	81.8	86.7	71.5
	PVT-Small [34][arxiv 2021]	25	224^2	3.8	79.8	–	–
	PVT-Medium [34][arxiv 2021]	44	224^2	6.7	81.2	–	–
	PVT-Large [34][arxiv 2021]	61	224^2	9.8	81.7	–	–
	T2T-ViT$_t$-14 [41][arxiv 2021]	22	224^2	6.1	80.7	–	–
	T2T-ViT$_t$-19 [41][arxiv 2021]	39	224^2	9.8	81.4	–	–
	T2T-ViT$_t$-24 [41][arxiv 2021]	64	224^2	15.0	82.2	–	–
	TNT-S [14][arxiv 2021]	24	224^2	5.2	81.3	–	–
	TNT-B [14][arxiv 2021]	66	224^2	14.1	82.8	–	–
Convolutional Transformers	**Ours:** CvT-13	20	224^2	4.5	81.6	86.7	70.4
	Ours: CvT-21	32	224^2	7.1	82.5	87.2	71.3
	Ours: CvT-13$_{\uparrow 384}$	20	384^2	16.3	83.0	87.9	71.9
	Ours: CvT-21$_{\uparrow 384}$	32	384^2	24.9	**83.3**	**87.7**	**71.9**
	Ours: CvT-13-NAS	18	224^2	4.1	82.2	87.5	71.3
Convolution Networks$_{22k}$	BiT-M$_{\uparrow 480}$ [18]	928	480^2	837	85.4	–	–
Transformers$_{22k}$	ViT-B/16$_{\uparrow 384}$ [11]	86	384^2	55.5	84.0	88.4	–
	ViT-L/16$_{\uparrow 384}$ [11]	307	384^2	191.1	85.2	88.4	–
	ViT-H/16$_{\uparrow 384}$ [11]	632	384^2	–	85.1	88.7	–
Convolutional Transformers$_{22k}$	**Ours:** CvT-13$_{\uparrow 384}$	20	384^2	16	83.3	88.7	72.9
	Ours: CvT-21$_{\uparrow 384}$	32	384^2	25	84.9	89.8	75.6
	Ours: CvT-W24$_{\uparrow 384}$	277	384^2	193.2	**87.7**	**90.6**	**78.8**

CoAtNet

- CoAtNet: Marrying Convolution and Attention for All Data Sizes
- Depthwise convolution merged into attention layers with simple relative attention
- Stacking convolutional and attention layers

Dai et al., 2021

$$y_i = \sum_{j \in \mathcal{L}(i)} w_{i-j} \odot x_j \quad \text{(depthwise convolution)}$$

$$y_i = \sum_{j \in \mathcal{G}} \underbrace{\frac{\exp\left(x_i^\top x_j\right)}{\sum_{k \in \mathcal{G}} \exp\left(x_i^\top x_k\right)}}_{A_{i,j}} x_j \quad \text{(self-attention)}$$

Properties	Convolution	Self-Attention
Translation Equivariance	✓	
Input-adaptive Weighting		✓
Global Receptive Field		✓

$$y_i^{\text{pre}} = \sum_{j \in \mathcal{G}} \frac{\exp\left(x_i^\top x_j + w_{i-j}\right)}{\sum_{k \in \mathcal{G}} \exp\left(x_i^\top x_k + w_{i-k}\right)} x_j$$

- generalization capability:

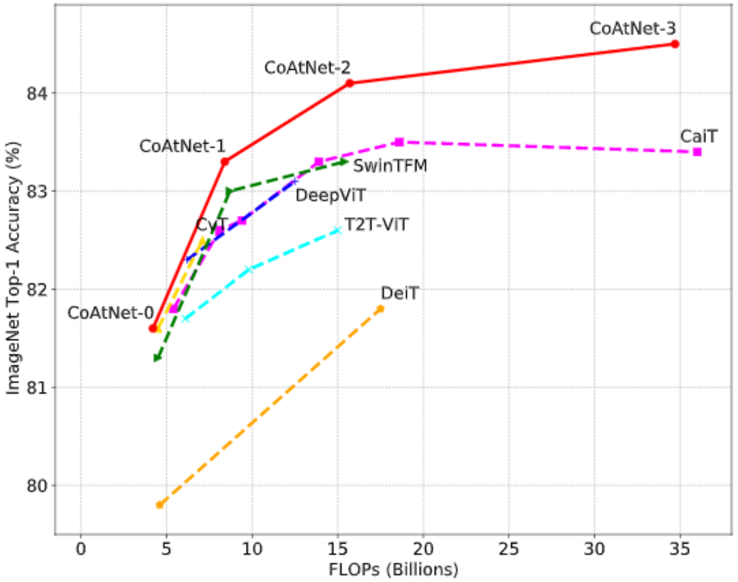
$$\text{C-C-C-C} \approx \text{C-C-C-T} \geq \text{C-C-T-T} > \text{C-T-T-T} \gg \text{ViT}_{\text{REL}}$$

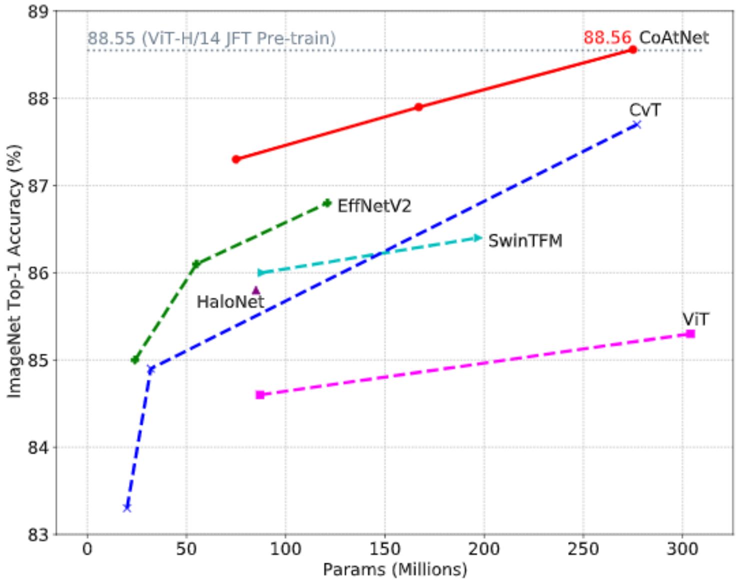
- model capacity:

$$\text{C-C-T-T} \approx \text{C-T-T-T} > \text{ViT}_{\text{REL}} > \text{C-C-C-T} > \text{C-C-C-C}$$

Metric	C-C-T-T	C-T-T-T
Pre-training Precision@1 (JFT)	34.40	34.36
Transfer Accuracy 224x224	**82.39**	81.78
Transfer Accuracy 384x384	**84.23**	84.02

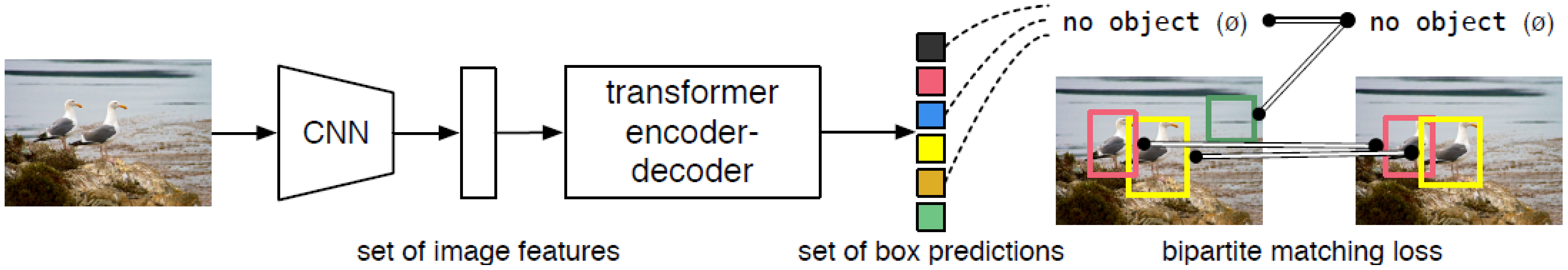
CoAtNet results

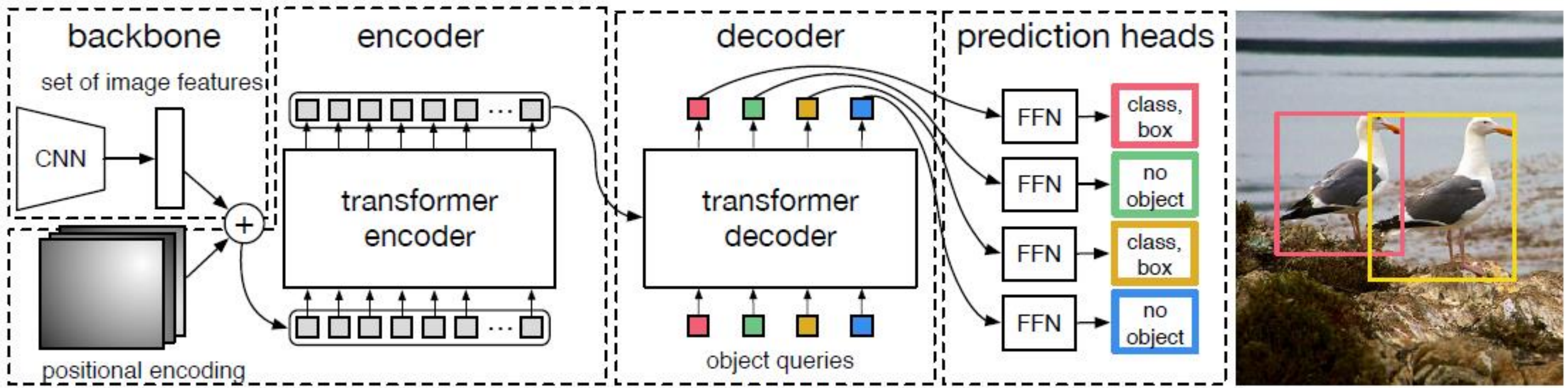




Models	Eval Size	#Params	#FLOPs	TPUv3-core-days	Top-1 Accuracy
ResNet + ViT-L/16	384^2	330M	-	-	87.12
ViT-L/16	512^2	307M	364B	0.68K	87.76
ViT-H/14	518^2	632M	1021B	2.5K	88.55
NFNet-F4+	512^2	527M	367B	1.86K	89.2
CoAtNet-3†	384^2	168M	114B	0.58K	88.52
CoAtNet-3†	512^2	168M	214B	0.58K	88.81
CoAtNet-4	512^2	275M	361B	0.95K	89.11
CoAtNet-5	512^2	688M	812B	1.82K	89.77
ViT-G/14	518^2	1.84B	5160B	>30K◇	90.45
CoAtNet-6	512^2	1.47B	1521B	6.6K	90.45
CoAtNet-7	512^2	2.44B	2586B	20.1K	**90.88**

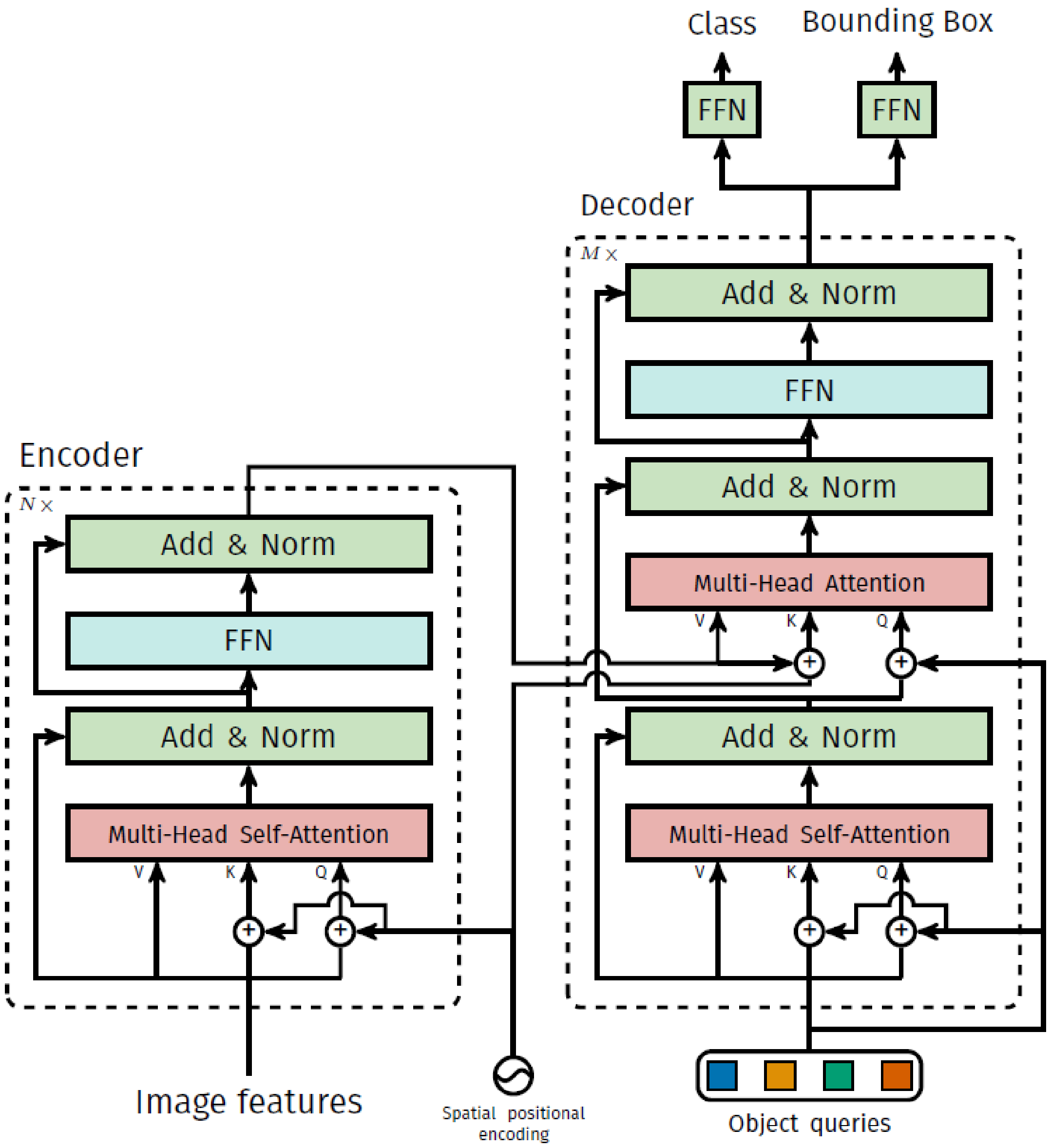
DETR - End-to-End Object Detection with Transformers





[Carion et al., 2020]

DETR transformer architecture

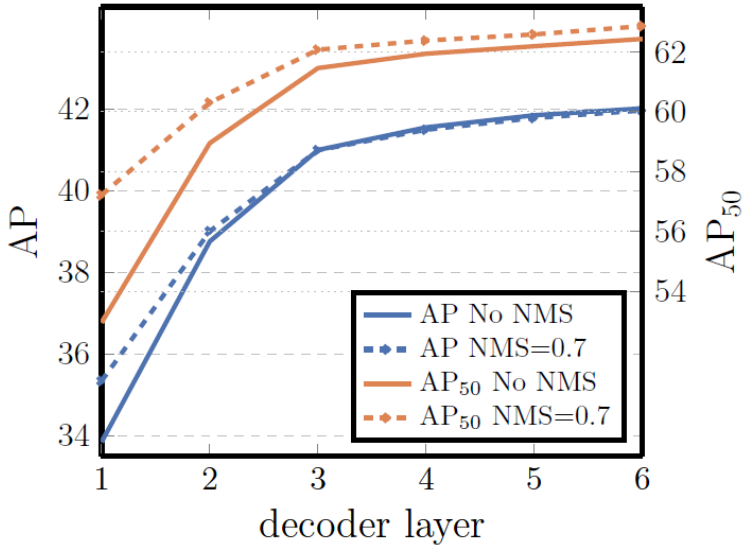


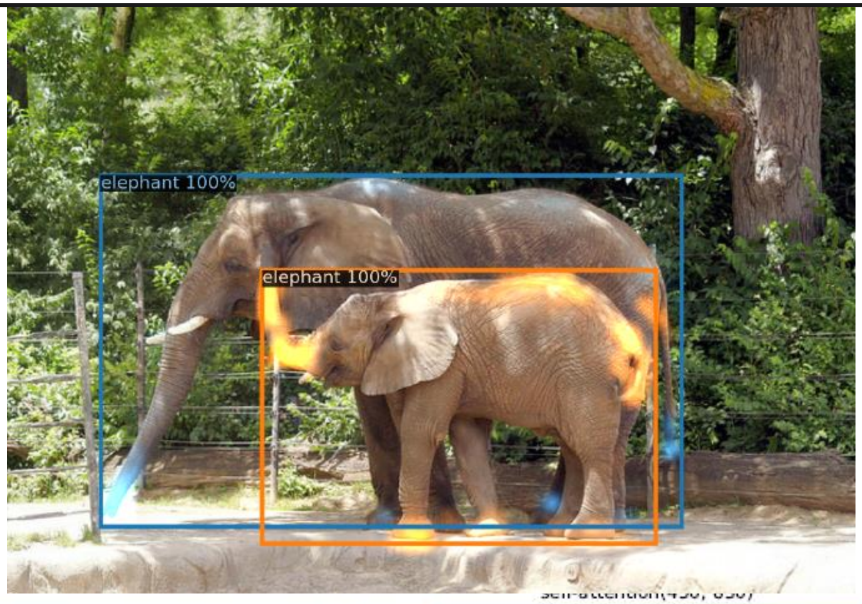
DETR detection results

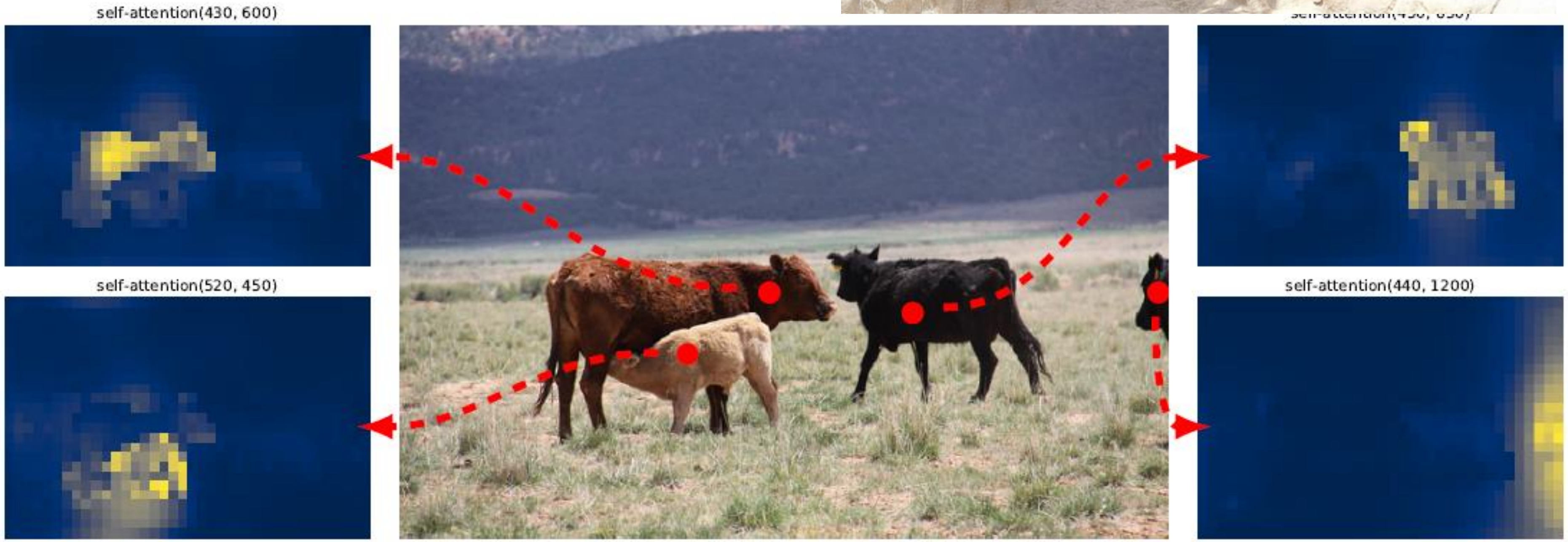
Model	GFLOPS/FPS	#params	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L
Faster RCNN-DC5	320/16	166M	39.0	60.5	42.3	21.4	43.5	52.5
Faster RCNN-FPN	180/26	42M	40.2	61.0	43.8	24.2	43.5	52.0
Faster RCNN-R101-FPN	246/20	60M	42.0	62.5	45.9	25.2	45.6	54.6
Faster RCNN-DC5+	320/16	166M	41.1	61.4	44.3	22.9	45.9	55.0
Faster RCNN-FPN+	180/26	42M	42.0	62.1	45.5	26.6	45.4	53.4
Faster RCNN-R101-FPN+	246/20	60M	44.0	63.9	**47.8**	**27.2**	48.1	56.0
DETR	86/28	41M	42.0	62.4	44.2	20.5	45.8	61.1
DETR-DC5	187/12	41M	43.3	63.1	45.9	22.5	47.3	61.1
DETR-R101	152/20	60M	43.5	63.8	46.4	21.9	48.0	61.8
DETR-DC5-R101	253/10	60M	**44.9**	**64.7**	47.7	23.7	**49.5**	**62.3**

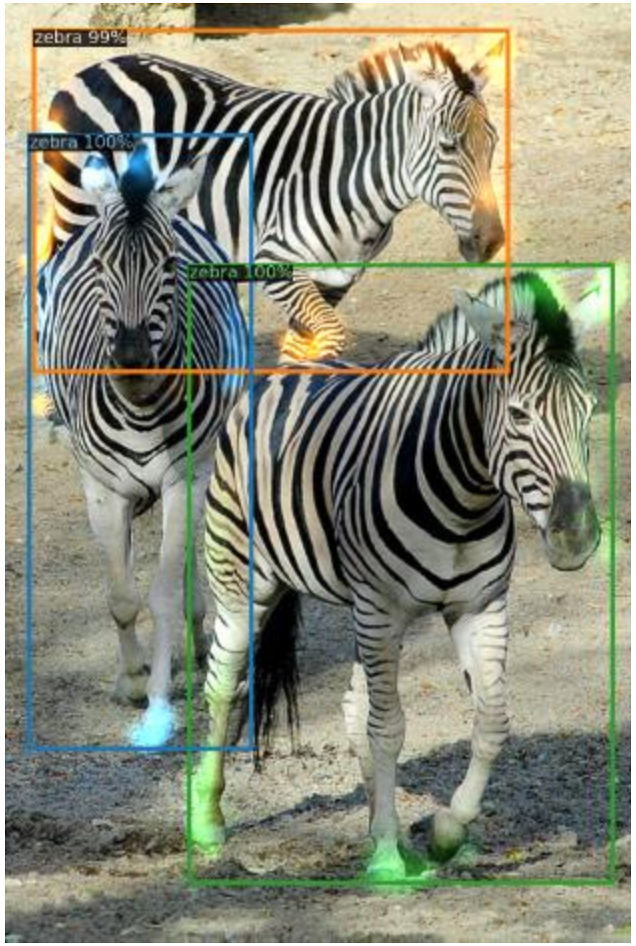
#layers	GFLOPS/FPS	#params	AP	AP_{50}	AP_S	AP_M	AP_L
0	76/28	33.4M	36.7	57.4	16.8	39.6	54.2
3	81/25	37.4M	40.1	60.6	18.5	43.8	58.6
6	86/23	41.3M	40.6	61.6	19.9	44.3	60.2
12	95/20	49.2M	41.6	62.1	19.8	44.9	61.9

DETR detection

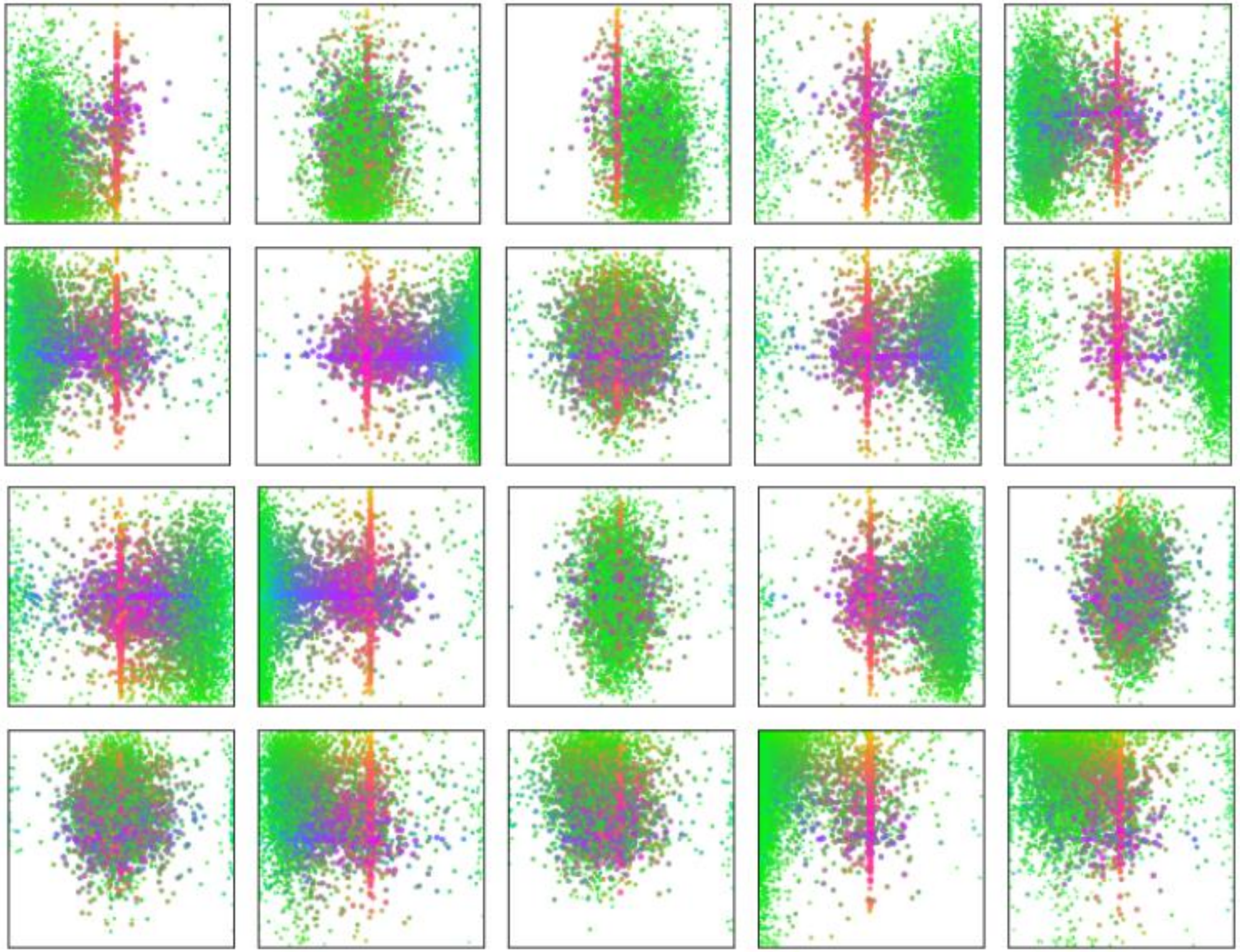






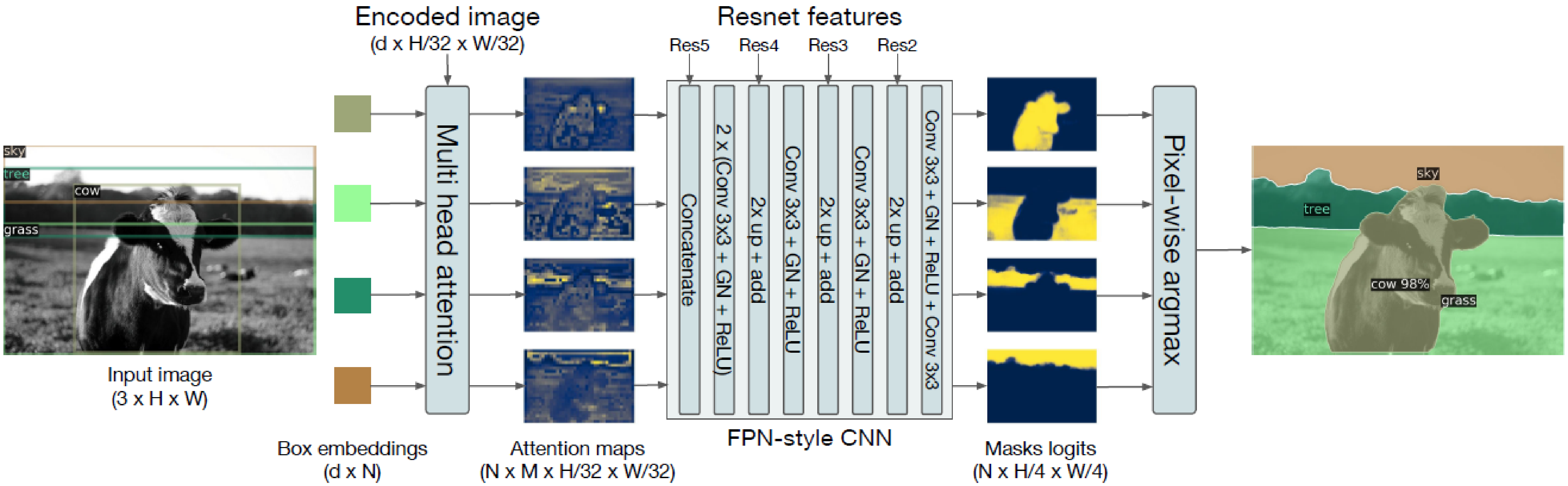


DETR box prediction



DETR panoptic segmentation

- Panoptic segmentation head

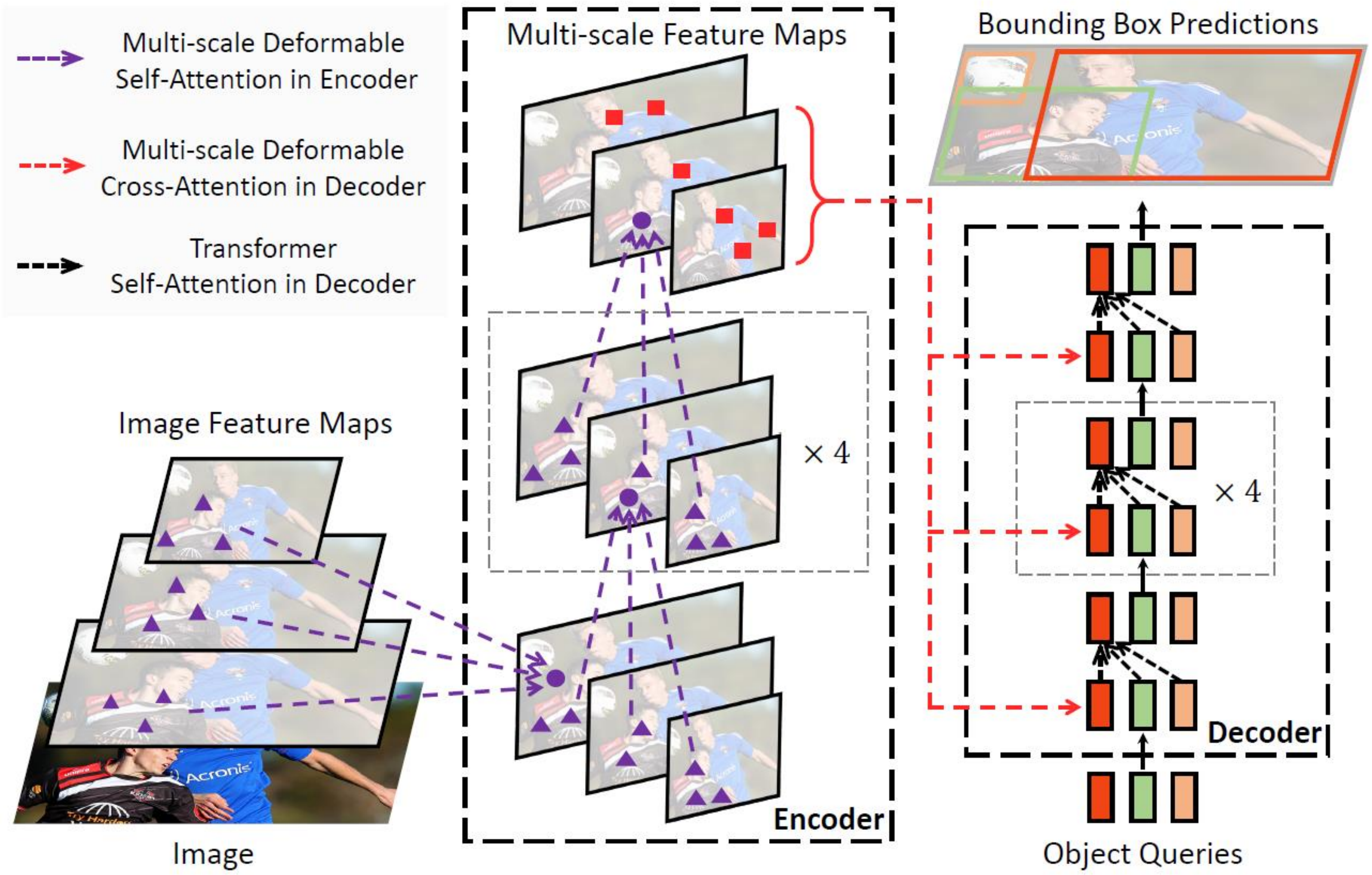


DETR panoptic segmentation results

Model	Backbone	PQ	SQ	RQ	PQ^{th}	SQ^{th}	RQ^{th}	PQ^{st}	SQ^{st}	RQ^{st}	AP
PanopticFPN++	R50	42.4	79.3	51.6	49.2	82.4	58.8	32.3	74.8	40.6	37.7
UPSnet	R50	42.5	78.0	52.5	48.6	79.4	59.6	33.4	75.9	41.7	34.3
UPSnet-M	R50	43.0	79.1	52.8	48.9	79.7	59.7	34.1	78.2	42.3	34.3
PanopticFPN++	R101	44.1	79.5	53.3	**51.0**	**83.2**	60.6	33.6	74.0	42.1	**39.7**
DETR	R50	43.4	79.3	53.8	48.2	79.8	59.5	36.3	78.5	45.3	31.1
DETR-DC5	R50	44.6	79.8	55.0	49.4	80.5	60.6	**37.3**	**78.7**	**46.5**	31.9
DETR-R101	R101	**45.1**	**79.9**	**55.5**	50.5	80.9	**61.7**	37.0	78.5	46.0	33.0

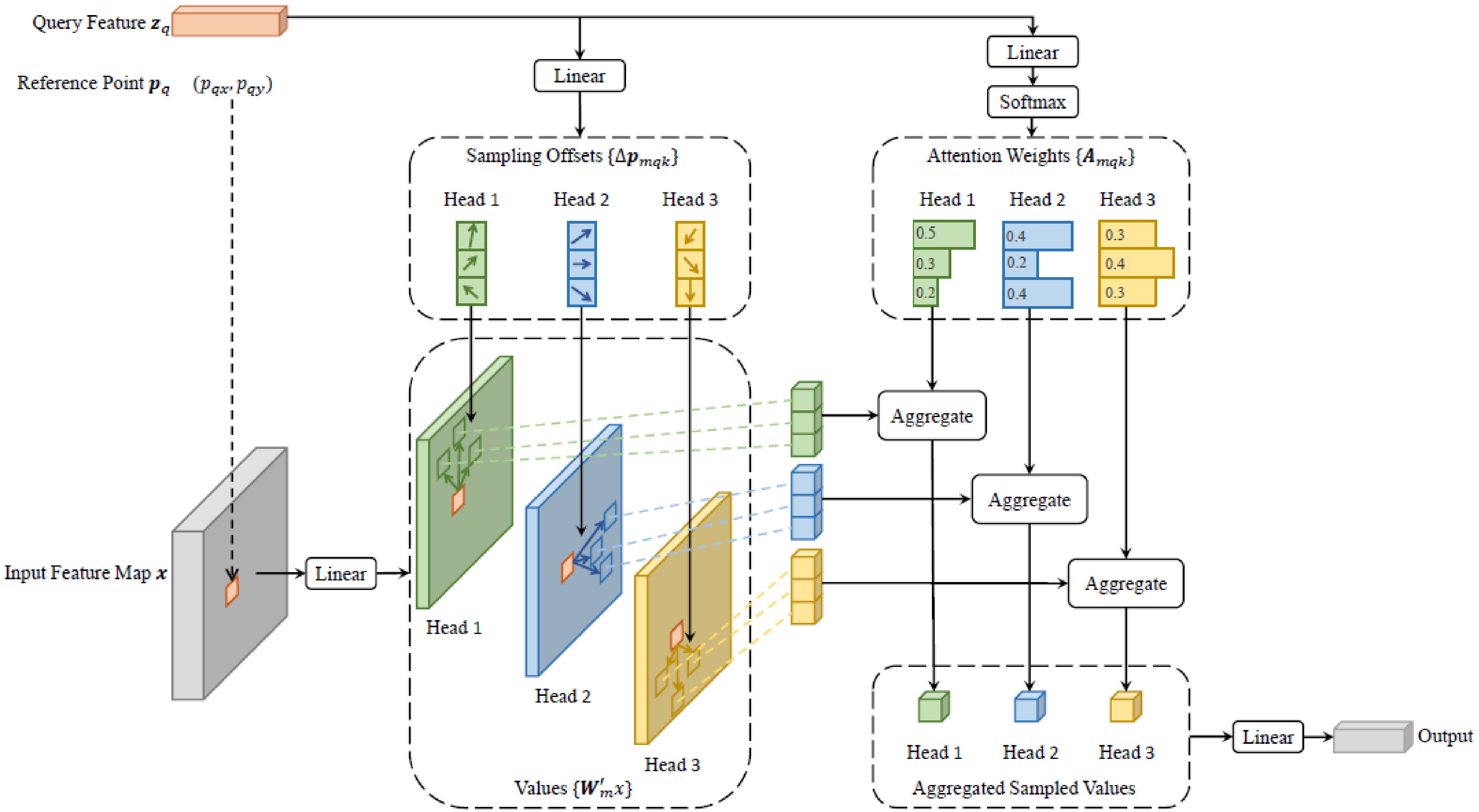
Deformable DETR

- Deformable transformers for end-to-end object detection



[Zhu et al., 2020]

Deformable DETR

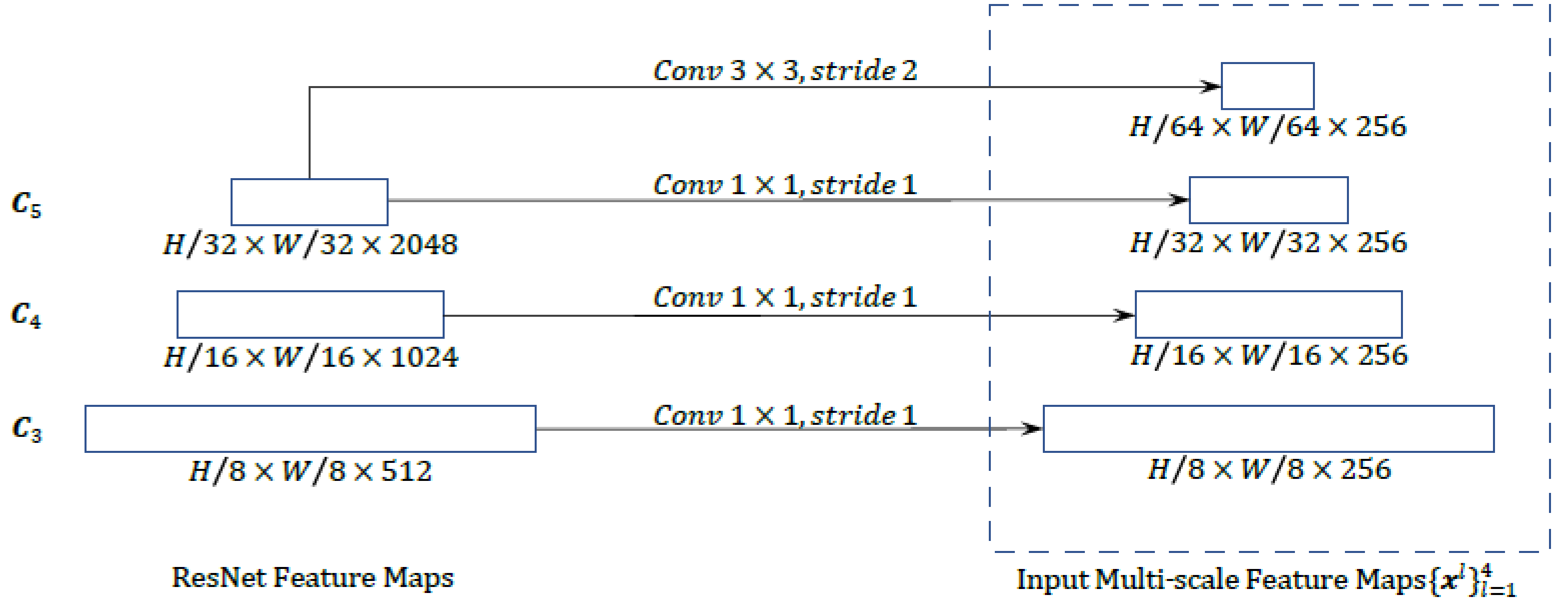


Multiscale deformable attention

$$\text{MultiHeadAttn}(\boldsymbol{z}_q, \boldsymbol{x}) = \sum_{m=1}^{M} \boldsymbol{W}_m \Big[\sum_{k \in \Omega_k} A_{mqk} \cdot \boldsymbol{W}'_m \boldsymbol{x}_k \Big]$$

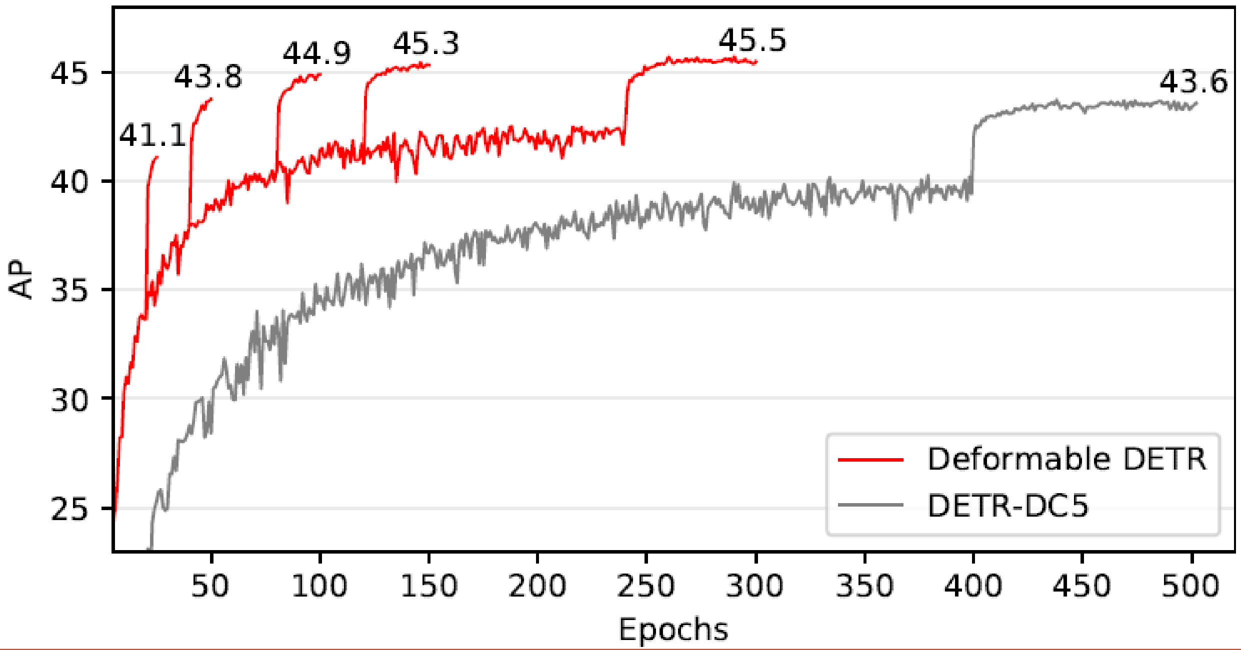
$$\text{DeformAttn}(\boldsymbol{z}_q, \boldsymbol{p}_q, \boldsymbol{x}) = \sum_{m=1}^{M} \boldsymbol{W}_m \Big[\sum_{k=1}^{K} A_{mqk} \cdot \boldsymbol{W}'_m \boldsymbol{x}(\boldsymbol{p}_q + \Delta \boldsymbol{p}_{mqk}) \Big]$$

$$\text{MSDeformAttn}(\boldsymbol{z}_q, \hat{\boldsymbol{p}}_q, \{\boldsymbol{x}^l\}_{l=1}^{L}) = \sum_{m=1}^{M} \boldsymbol{W}_m \Big[\sum_{l=1}^{L} \sum_{k=1}^{K} A_{mlqk} \cdot \boldsymbol{W}'_m \boldsymbol{x}^l (\phi_l(\hat{\boldsymbol{p}}_q) + \Delta \boldsymbol{p}_{mlqk}) \Big]$$



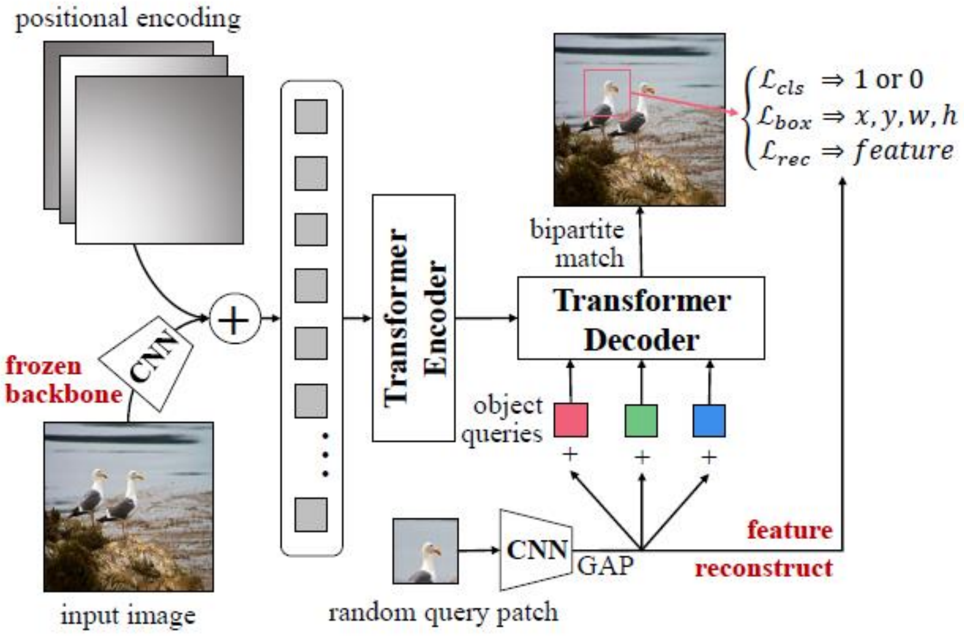
Deformable DETR results

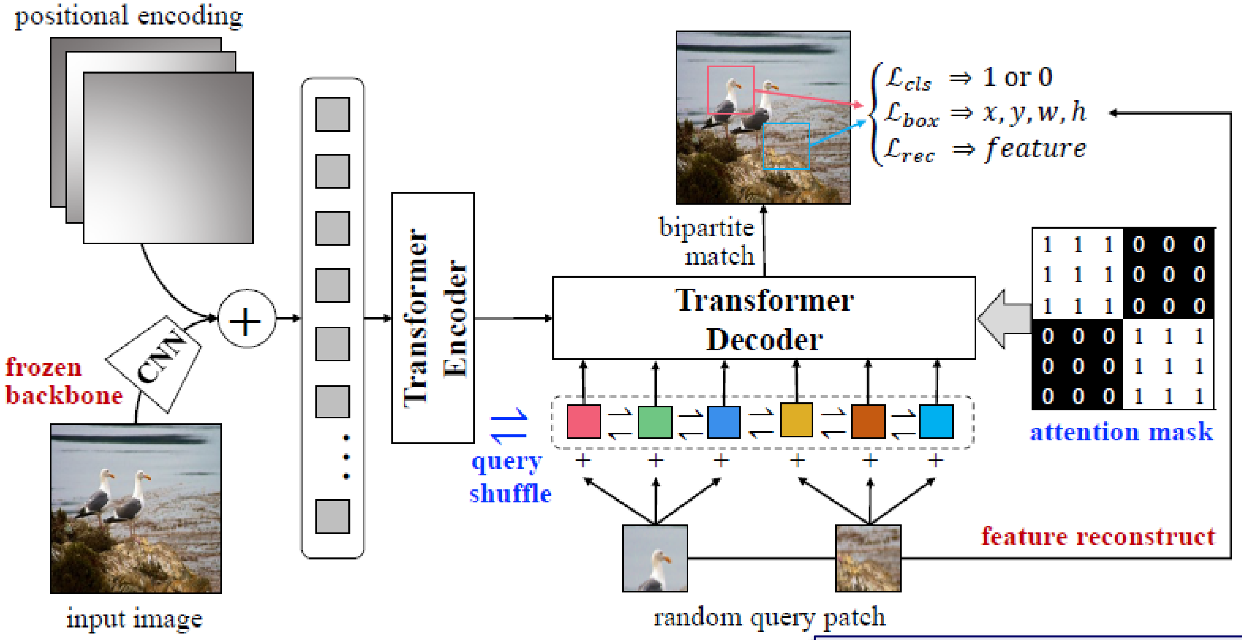
Method	Epochs	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L	params	FLOPs	Training GPU hours	Inference FPS
Faster R-CNN + FPN	109	42.0	62.1	45.5	26.6	45.4	53.4	42M	180G	380	26
DETR	500	42.0	62.4	44.2	20.5	45.8	61.1	41M	86G	2000	28
DETR-DC5	500	43.3	63.1	45.9	22.5	47.3	61.1	41M	187G	7000	12
DETR-DC5	50	35.3	55.7	36.8	15.2	37.5	53.6	41M	187G	700	12
DETR-DC5$^+$	50	36.2	57.0	37.4	16.3	39.2	53.9	41M	187G	700	12
Deformable DETR	50	43.8	62.6	47.7	26.4	47.1	58.0	40M	173G	325	19
+ iterative bounding box refinement	50	45.4	64.7	49.0	26.8	48.3	61.7	40M	173G	325	19
++ two-stage Deformable DETR	50	46.2	65.2	50.0	28.8	49.2	61.7	40M	173G	340	19



UP-DETR

- UP-DETR: Unsupervised Pre-training for Object Detection with Transformer
  - Unsupervised pretraining on a large-scale dataset
    - Detect randomly cropped query patches
  - Supervised fine-tunning as in DETR
- Single-query and multiply-query patches for unsupervised pretraining

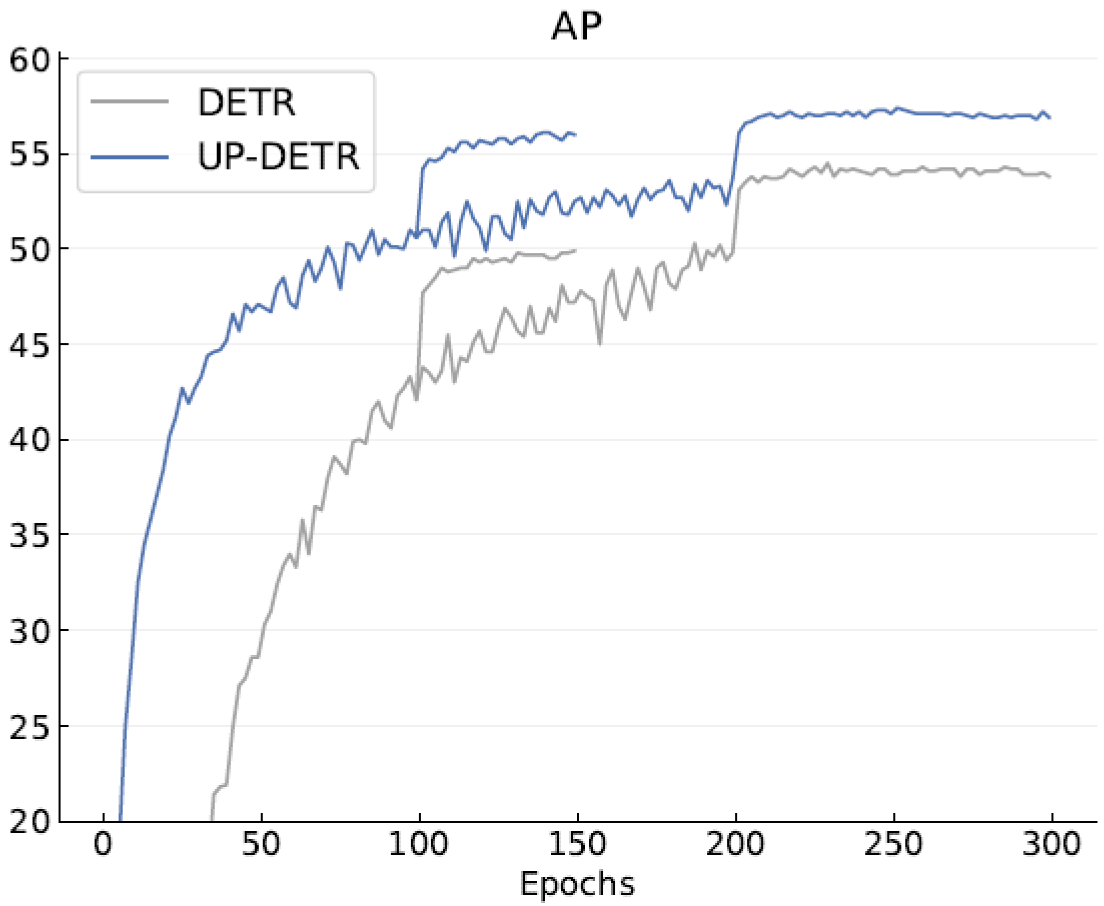




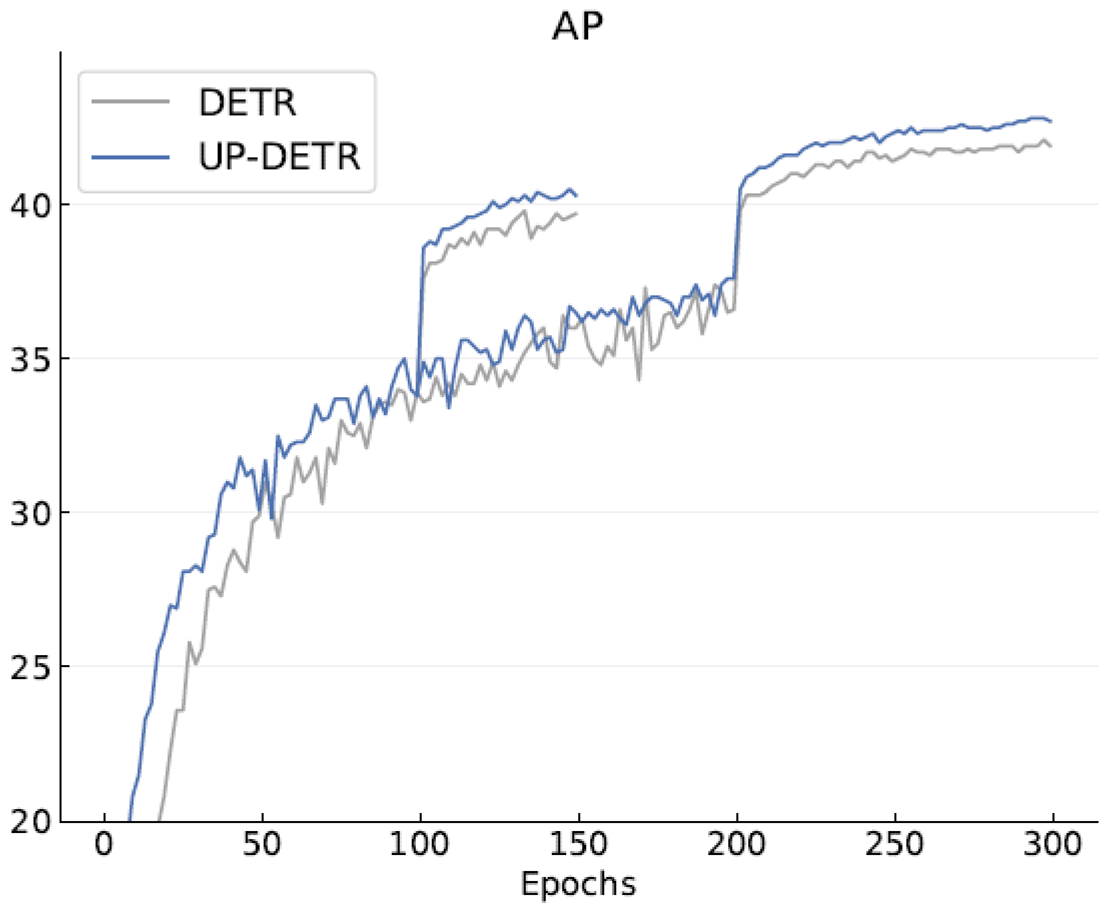
Dai et al., 2020

UP-DETR results

- Pre-training helps!



AP learning curves of VOC



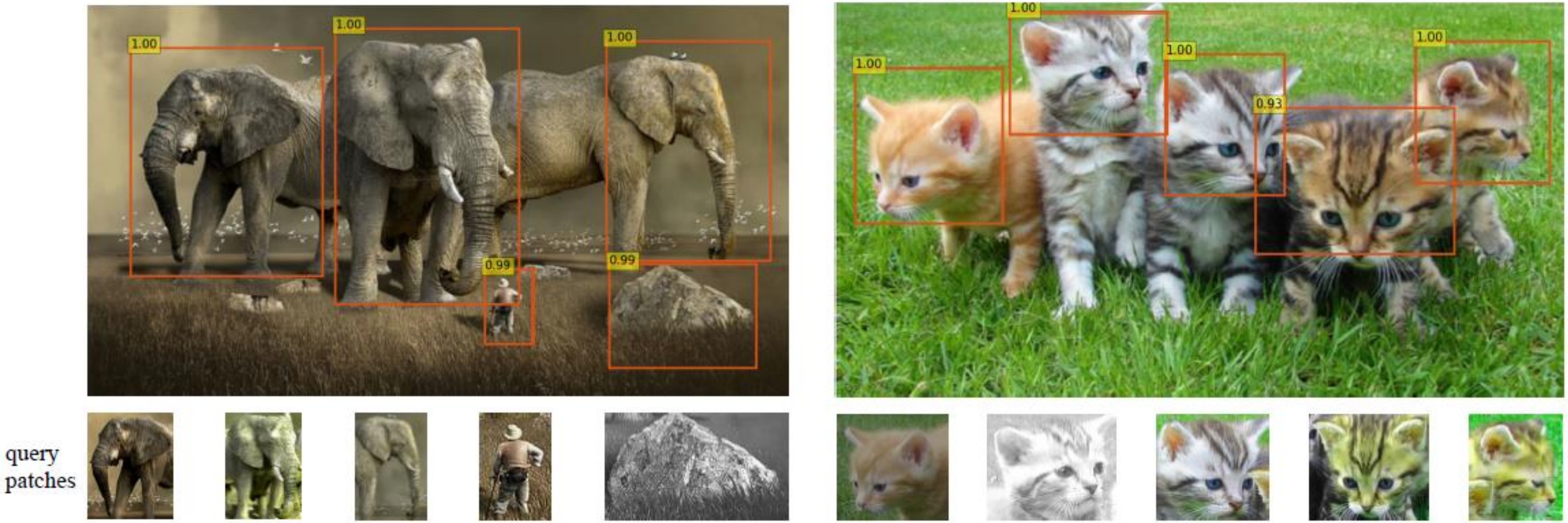
AP learning curves of COCO

UP-DETR results

Model	Backbone	Epochs	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L
Faster R-CNN † [26]	R101-FPN	-	36.2	59.1	39.0	18.2	39.0	48.2
Mask R-CNN † [18]	R101-FPN	-	38.2	60.3	41.7	20.1	41.1	50.2
Grid R-CNN † [31]	R101-FPN	-	41.5	60.9	44.5	23.3	44.9	53.1
Double-head R-CNN [40]	R101-FPN	-	41.9	62.4	45.9	23.9	45.2	55.8
RetinaNet † [27]	R101-FPN	-	39.1	59.1	42.3	21.8	42.7	50.2
FCOS † [38]	R101-FPN	-	41.5	60.7	45.0	24.4	44.8	51.6
DETR [5]	R50	500	42.0	62.4	44.2	20.5	45.8	61.1
Faster R-CNN	R50-FPN	3×	40.2	**61.0**	**43.8**	**24.2**	43.5	52.0
DETR (Supervised CNN)	R50	150	39.5	60.3	41.4	17.5	43.0	59.1
DETR (SwAV CNN) [7]	R50	150	39.7	60.3	41.7	18.5	43.8	57.5
UP-DETR	R50	150	**40.5** (+0.8)	60.8	42.6	19.0	**44.4**	**60.0**
Faster R-CNN	R50-FPN	9×	42.0	62.1	**45.5**	**26.6**	45.4	53.4
DETR (Supervised CNN)	R50	300	40.8	61.2	42.9	20.1	44.5	60.3
DETR (SwAV CNN) [7]	R50	300	42.1	**63.1**	44.5	19.7	46.3	60.9
UP-DETR	R50	300	**42.8** (+0.7)	63.0	45.3	20.8	**47.1**	**61.7**

UP-DETR results

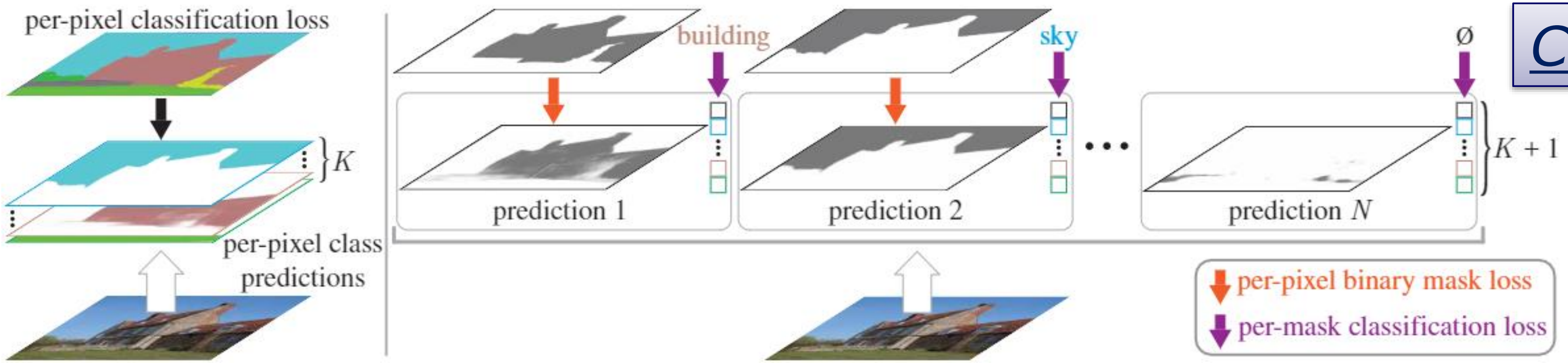
- Unsupervised one-shot detection
- Deep-learning-based template matching

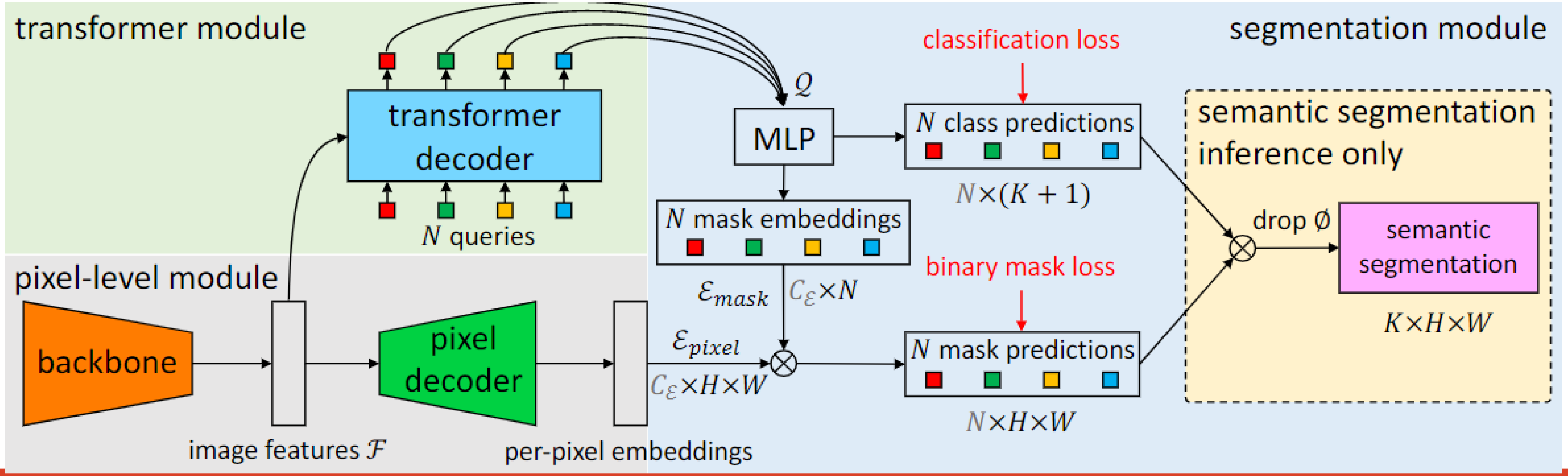


MaskFormer

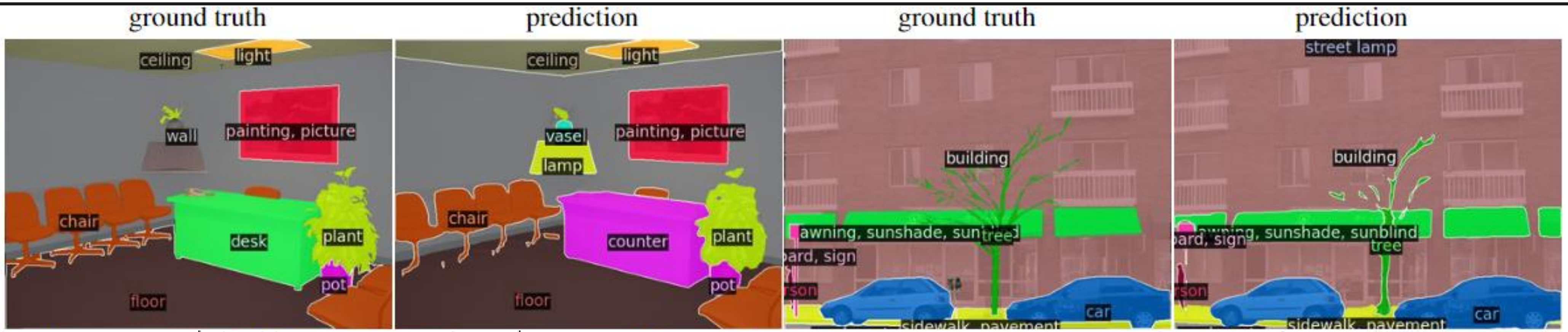
- Per-Pixel Classification is Not All You Need for Semantic Segmentation

[Cheng et al., 2021]

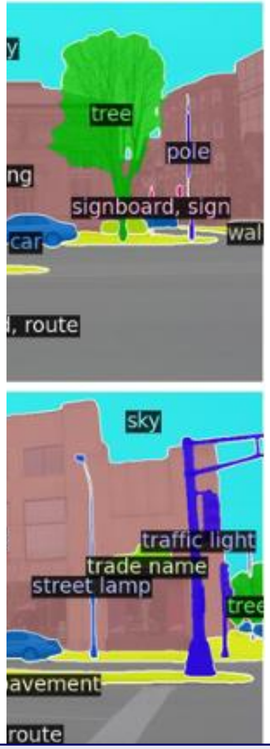




MaskFormer results



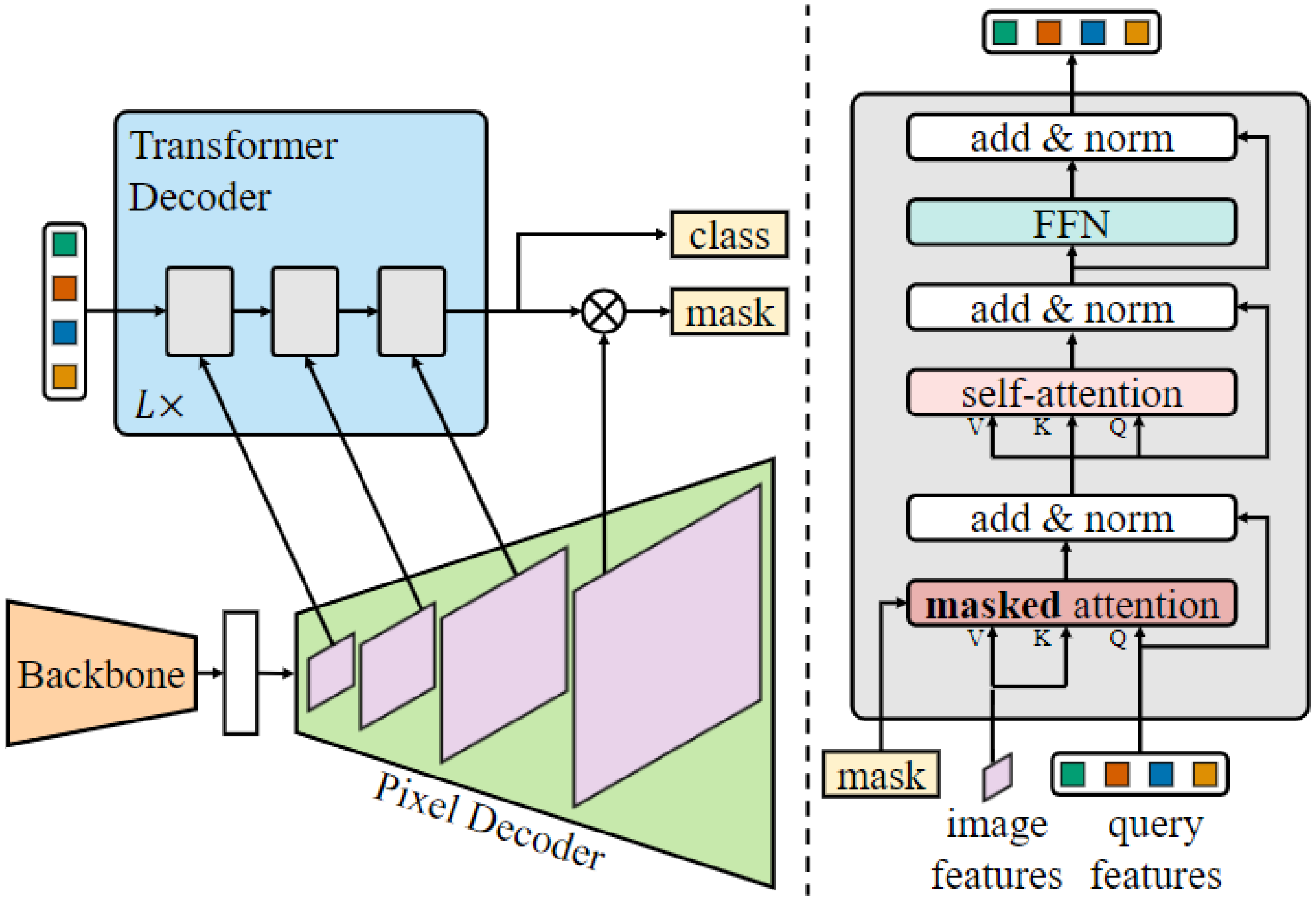
	method	backbone	crop size	mIoU (s.s.)	mIoU (m.s.)	#params.	FLOPs	fps
CNN backbones	OCRNet [50]	R101c	520 × 520	-	45.3	-	-	-
	DeepLabV3+ [9]	R50c	512 × 512	44.0	44.9	44M	177G	21.0
		R101c	512 × 512	45.5	46.4	63M	255G	14.2
	MaskFormer (ours)	R50	512 × 512	44.5 ±0.5	46.7 ±0.6	41M	53G	24.5
		R101	512 × 512	45.5 ±0.5	47.2 ±0.2	60M	73G	19.5
		R101c	512 × 512	**46.0** ±0.1	**48.1** ±0.2	60M	80G	19.0
Transformer backbones	SETR [53]	ViT-L†	512 × 512	-	50.3	308M	-	-
	Swin-UperNet [29, 49]	Swin-T	512 × 512	-	46.1	60M	236G	18.5
		Swin-S	512 × 512	-	49.3	81M	259G	15.2
		Swin-B†	640 × 640	-	51.6	121M	471G	8.7
		Swin-L†	640 × 640	-	53.5	234M	647G	6.2
	MaskFormer (ours)	Swin-T	512 × 512	46.7 ±0.7	48.8 ±0.6	42M	55G	22.1
		Swin-S	512 × 512	49.8 ±0.4	51.0 ±0.4	63M	79G	19.6
		Swin-B	640 × 640	51.1 ±0.2	52.3 ±0.4	102M	195G	12.6
		Swin-B†	640 × 640	52.7 ±0.4	53.9 ±0.2	102M	195G	12.6
		Swin-L†	640 × 640	**54.1** ±0.2	**55.6** ±0.1	212M	375G	7.9



Zhou et al., 2017

Mask2Former

- Masked-attention Mask Transformer for Universal Image Segmentation



$$\mathbf{X}_l = \text{softmax}(\mathbf{Q}_l\mathbf{K}_l^{\mathrm{T}})\mathbf{V}_l + \mathbf{X}_{l-1}$$

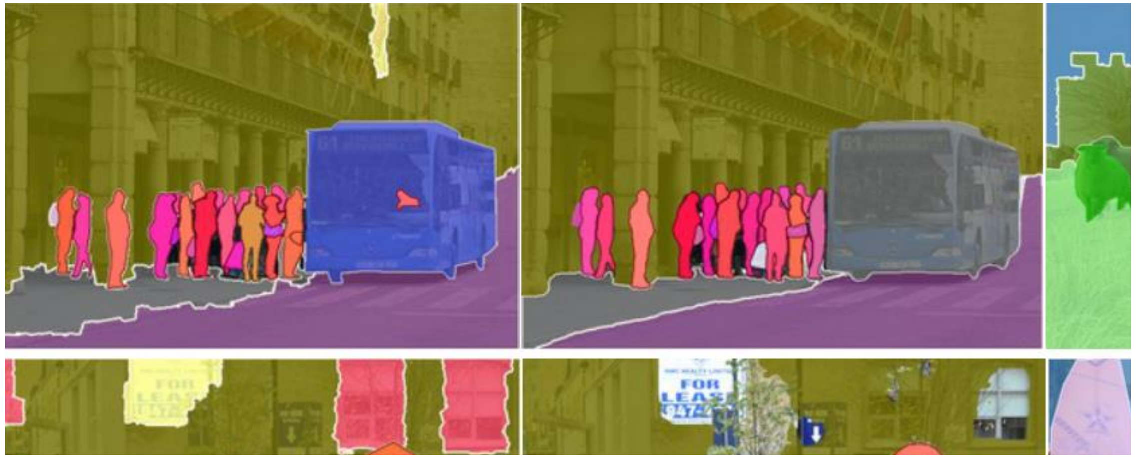
$$\downarrow$$

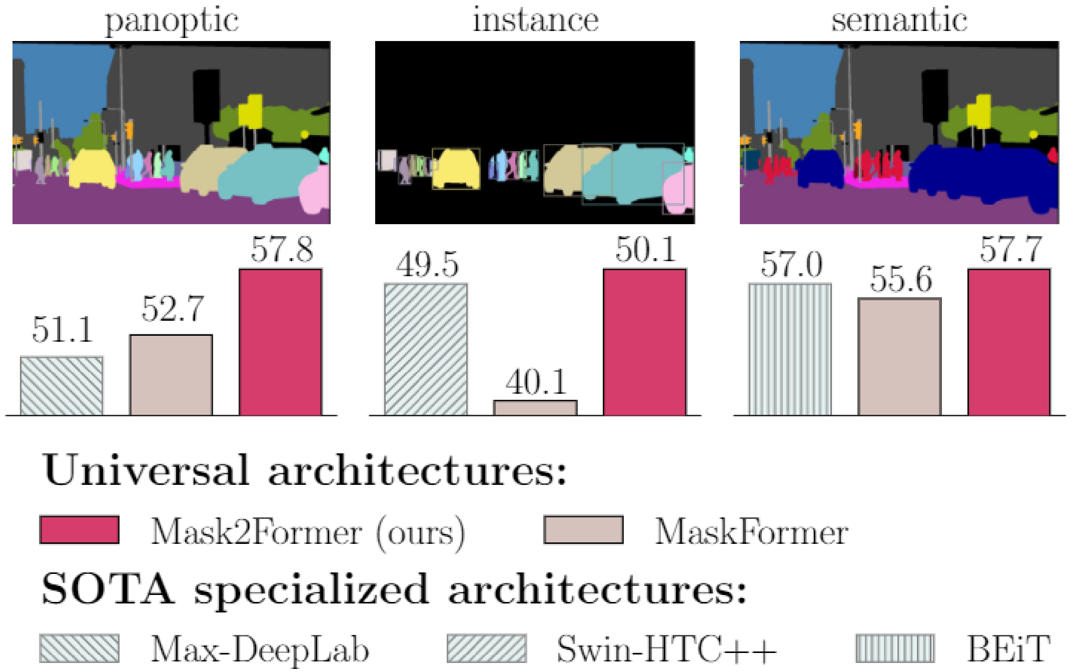
$$\mathbf{X}_l = \text{softmax}(\mathcal{M}_{l-1} + \mathbf{Q}_l\mathbf{K}_l^{\mathrm{T}})\mathbf{V}_l + \mathbf{X}_{l-1}$$

$$\mathcal{M}_{l-1}(x,y) = \begin{cases} 0 & \text{if } \mathbf{M}_{l-1}(x,y) = 1 \\ -\infty & \text{otherwise} \end{cases}$$

Cheng et al., 2021b

Mask2Former results





method	backbone	query type	epochs	PQ	PQ^{Th}	PQ^{St}	AP^{Th}_{pan}	$mIoU_{pan}$	#params.	FLOPs	fps
DETR [5]	R50	100 queries	500+25	43.4	48.2	36.3	31.1	-	-	-	-
MaskFormer [14]	R50	100 queries	300	46.5	51.0	39.8	33.0	57.8	45M	181G	17.6
Mask2Former (ours)	R50	100 queries	50	**51.9**	**57.7**	**43.0**	**41.7**	**61.7**	44M	226G	8.6
DETR [5]	R101	100 queries	500+25	45.1	50.5	37.0	33.0	-	-	-	-
MaskFormer [14]	R101	100 queries	300	47.6	52.5	40.3	34.1	59.3	64M	248G	14.0
Mask2Former (ours)	R101	100 queries	50	**52.6**	**58.5**	**43.7**	**42.6**	**62.4**	63M	293G	7.2
Max-DeepLab [52]	Max-L	128 queries	216	51.1	57.0	42.2	-	-	451M	3692G	-
MaskFormer [14]	Swin-L†	100 queries	300	52.7	58.5	44.0	40.1	64.8	212M	792G	5.2
K-Net [62]	Swin-L†	100 queries	36	54.6	60.2	46.0	-	-	-	-	-
Mask2Former (ours)	Swin-L†	*200 queries*	*100*	**57.8**	**64.2**	**48.1**	**48.6**	**67.4**	216M	868G	4.0

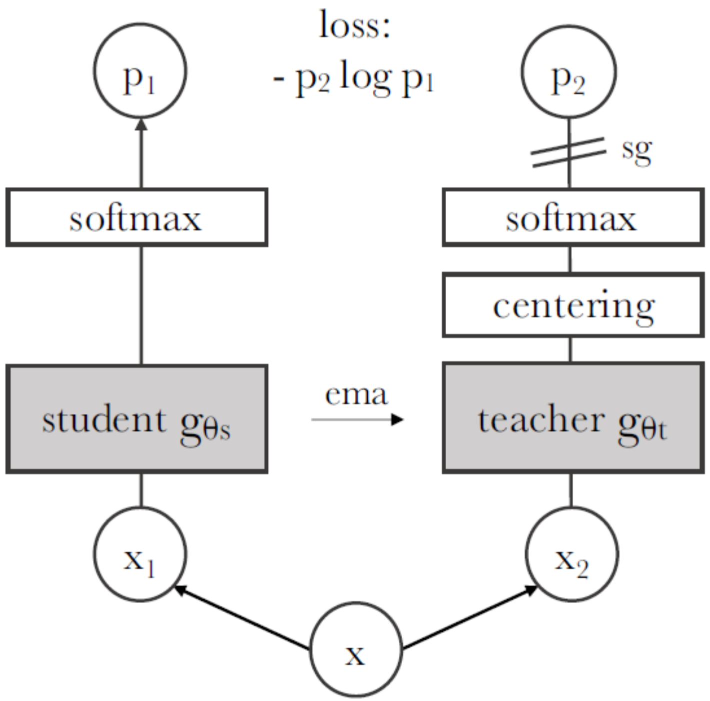
Lin et al., 2014

DINO

- Emerging Properties in Self-Supervised Vision Transformers
- Self <u>di</u>stillation with <u>no</u> labels

[Carionet et al., 2021]

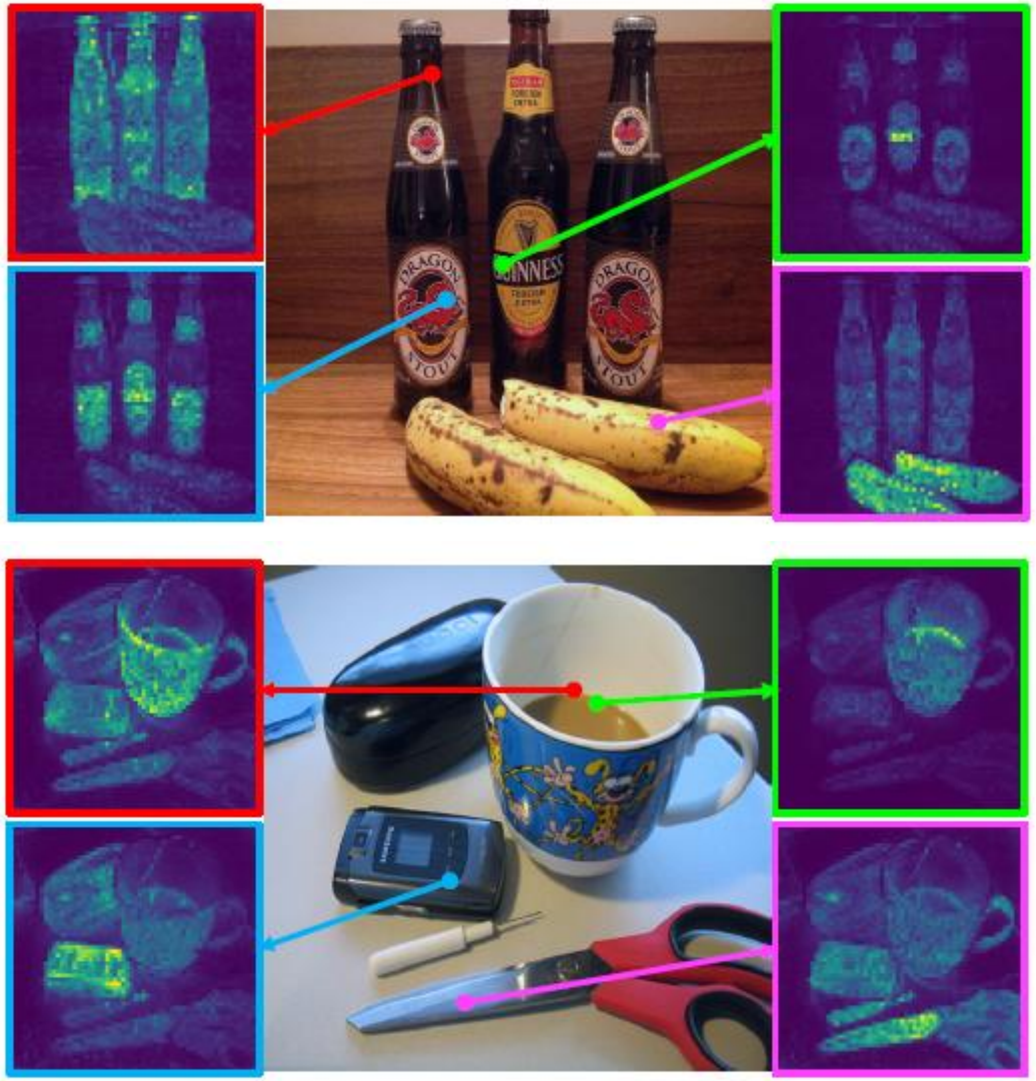
DINO

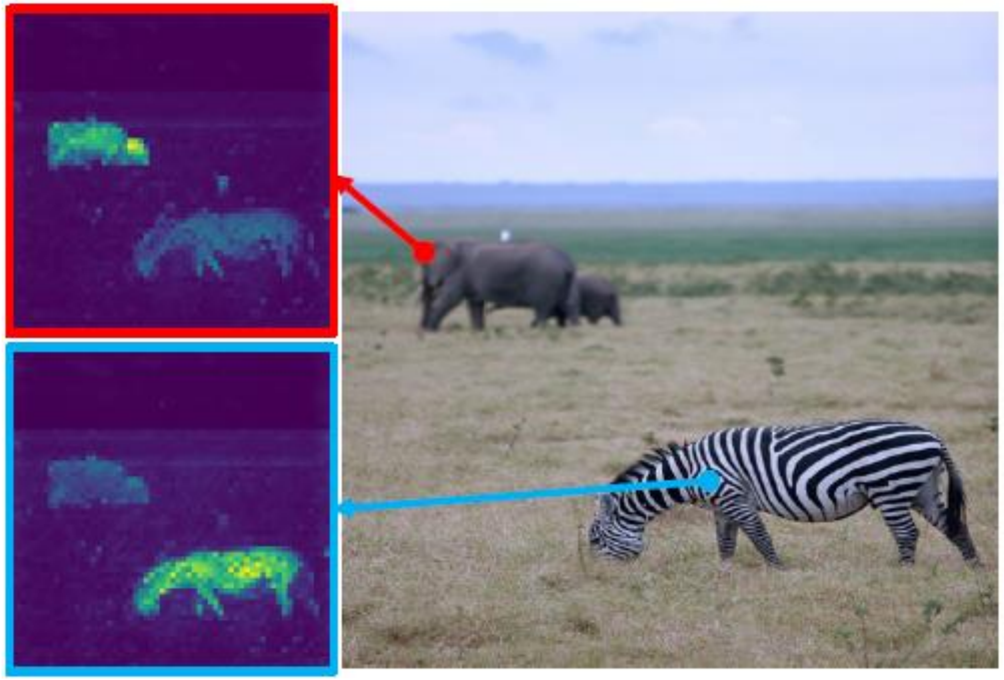


Algorithm 1 DINO PyTorch pseudocode w/o multi-crop.

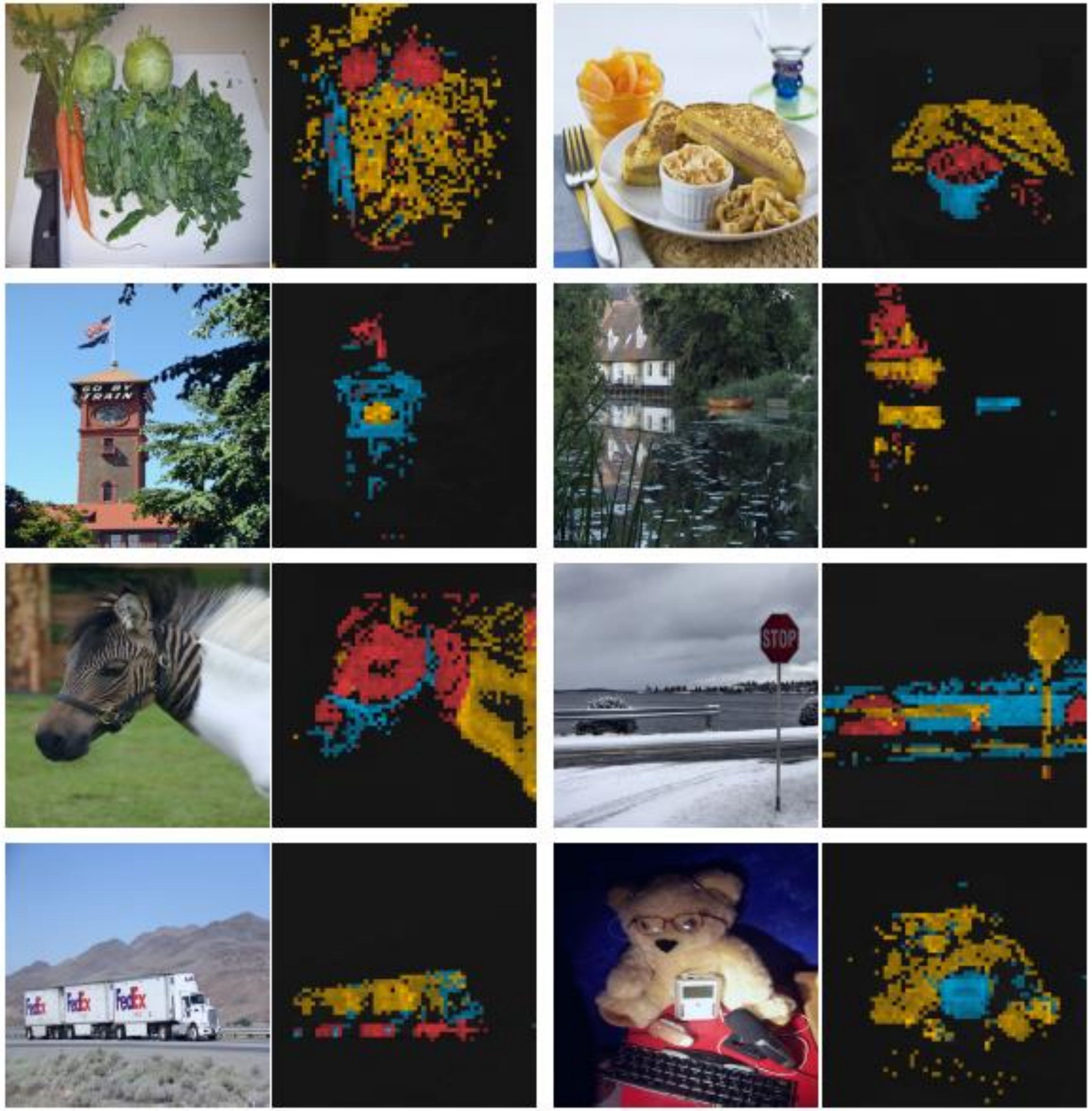
```
# gs, gt: student and teacher networks
# C: center (K)
# tps, tpt: student and teacher temperatures
# l, m: network and center momentum rates
gt.params = gs.params
for x in loader: # load a minibatch x with n samples
    x1, x2 = augment(x), augment(x) # random views

    s1, s2 = gs(x1), gs(x2) # student output n-by-K
    t1, t2 = gt(x1), gt(x2) # teacher output n-by-K

    loss = H(t1, s2)/2 + H(t2, s1)/2
    loss.backward() # back-propagate

    # student, teacher and center updates
    update(gs) # SGD
    gt.params = l*gt.params + (1-l)*gs.params
    C = m*C + (1-m)*cat([t1, t2]).mean(dim=0)

def H(t, s):
    t = t.detach() # stop gradient
    s = softmax(s / tps, dim=1)
    t = softmax((t - C) / tpt, dim=1) # center + sharpen
    return - (t * log(s)).sum(dim=1).mean()
```

DINO self-attention





DINO segmentation results



	Random	Supervised	DINO
DeiT-S/16	22.0	27.3	45.9
DeiT-S/8	21.8	23.7	44.7

DINO experimental results

- Linear and k-NN classification on ImageNet

Method	Arch.	Param.	im/s	Linear	k-NN
Supervised	RN50	23	1237	79.3	79.3
SCLR [12]	RN50	23	1237	69.1	60.7
MoCov2 [14]	RN50	23	1237	71.1	61.9
InfoMin [64]	RN50	23	1237	73.0	65.3
BarlowT [78]	RN50	23	1237	73.2	66.0
OBoW [25]	RN50	23	1237	73.8	61.9
BYOL [28]	RN50	23	1237	74.4	64.8
DCv2 [10]	RN50	23	1237	75.2	67.1
SwAV [10]	RN50	23	1237	**75.3**	65.7
DINO	RN50	23	1237	**75.3**	**67.5**
Supervised	DeiT-S	21	1007	79.8	79.8
BYOL* [28]	DeiT-S	21	1007	71.4	66.6
MoCov2* [14]	DeiT-S	21	1007	72.7	64.4
SwAV* [10]	DeiT-S	21	1007	73.5	66.3
DINO	DeiT-S	21	1007	**77.0**	**74.5**

Comparison across architectures					
SCLR [12]	RN50w4	375	117	76.8	69.3
SwAV [10]	RN50w2	93	384	77.3	67.3
BYOL [28]	RN50w2	93	384	77.4	–
DINO	ViT-B/16	85	312	78.2	76.1
SwAV [10]	RN50w5	586	76	78.5	67.1
BYOL [28]	RN50w4	375	117	78.6	–
BYOL [28]	RN200w2	250	123	79.6	73.9
DINO	DeiT-S/8	21	180	79.7	**78.3**
SCLRv2 [13]	RN152w3+SK	794	46	79.8	73.1
DINO	ViT-B/8	85	63	**80.1**	77.4

DINO experimental results

- DAVIS 2017 Video object segmentation

Method	Data	Arch.	$(\mathcal{J}\&\mathcal{F})_m$	$\mathcal{J}_m$	$\mathcal{F}_m$
Supervised					
ImageNet	INet	DeiT-S/8	66.0	63.9	68.1
STM [46]	I/D/Y	RN50	81.8	79.2	84.3
Self-supervised					
CT [68]	VLOG	RN50	48.7	46.4	50.0
MAST [38]	YT-VOS	RN18	65.5	63.3	67.6
STC [35]	Kinetics	RN18	67.6	64.8	70.2
DINO	INet	DeiT-S/16	61.8	60.2	63.4
DINO	INet	ViT-B/16	62.3	60.7	63.9
DINO	INet	DeiT-S/8	**69.9**	**66.6**	**73.1**
DINO	INet	ViT-B/8	**71.4**	**67.9**	**74.9**

- Transfer learning by fine-tuning pre-trained models on different datasets

	Cifar_{10}	Cifar_{100}	INat_{18}	INat_{19}	Flwrs	Cars	INet
DeiT-S/16							
Sup. [66]	**99.0**	89.5	70.7	76.6	98.2	92.1	79.9
DINO	**99.0**	**90.5**	**72.0**	**78.2**	**98.5**	**93.0**	**81.5**
ViT-B/16							
Sup. [66]	99.0	90.8	**73.2**	77.7	98.4	92.1	81.8
DINO	**99.1**	**91.7**	72.6	**78.6**	**98.8**	**93.0**	**82.8**

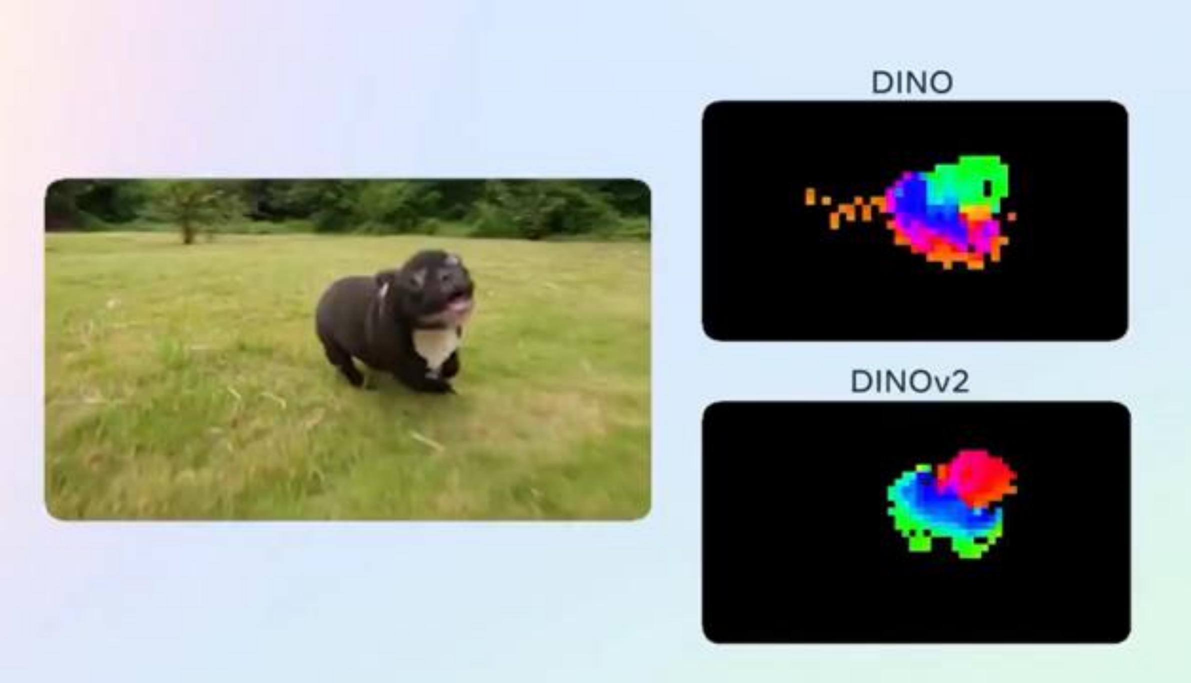
DINO

DINOv2

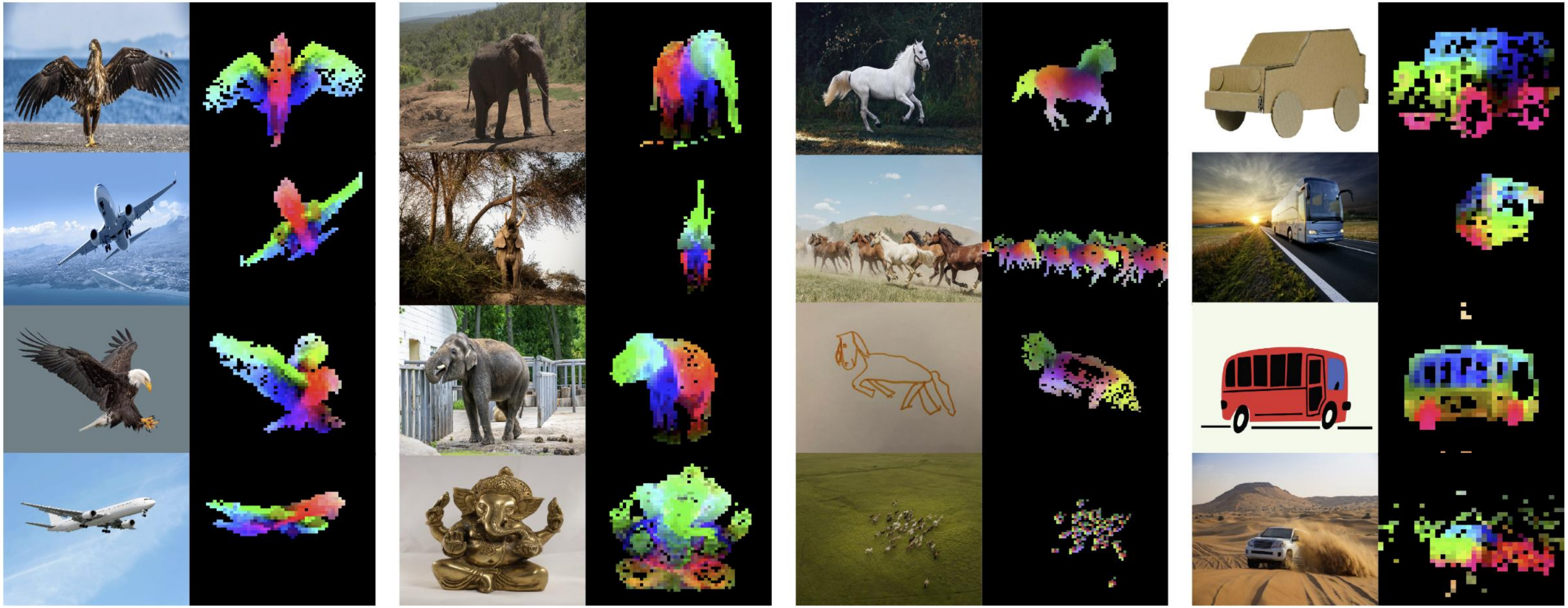
- DINOv2: Learning Robust Visual Features without Supervision
- Discriminative self-supervised learning
- No fine-tunning – general multipurpose backbone
- Foundation model
- Multipupose backbone -high-performance features to be used
  - classification, segmantation, image retrieval, depth estimation

Oquab et al., 2023

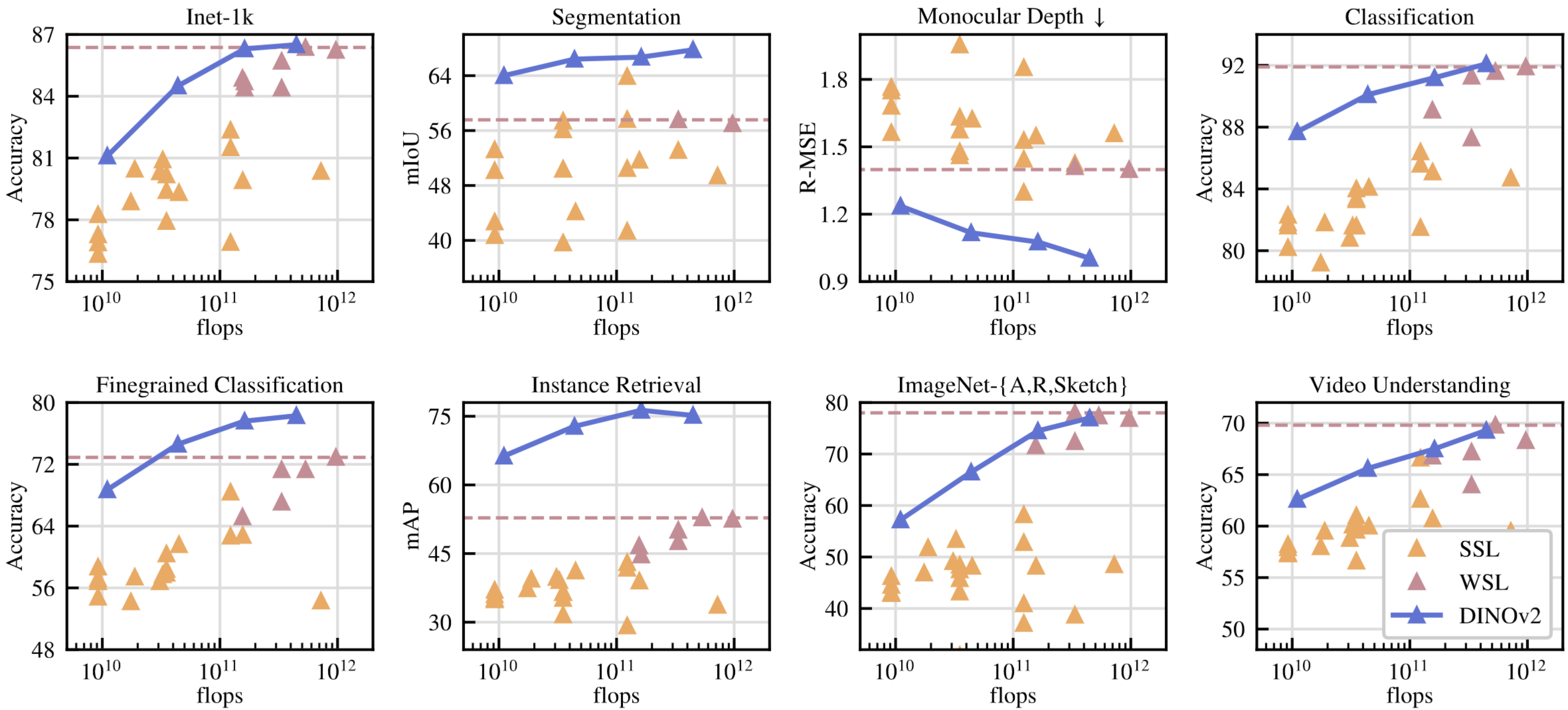
- Automatic pipeline to build a dedicated, diverse, and curated image dataset
- ViT model with 1B parameters
  - distill it into a series of smaller models
- Accelerating and stabilizing the training at scale
- 2×faster and require 3×less memory than similar self-supervised methods
- SOTA results



DINOv2



DINOv2


Inet-1k
Segmentation
Monocular Depth ↓
Classification
Finegrained Classification
Instance Retrieval
ImageNet-{A,R,Sketch}
Video Understanding
Accuracy
mIoU
R-MSE
mAP
flops
SSL
WSL
DINOv2

Tasks and design choices

Khan et al., 2021

Task	Method	Design Highlights (focus on differences with the standard form)	Input Data Type	Label Type	Loss
Image Classification	ViT [11]	Directly adopted NLP Transformer Encoder for images, Mechanism to linearly embed image patches with positional embedding suitable for the Encoder.	2D Image	Class labels	Cross-entropy
	DeiT [12]	Transformer as s student while CNN as a teacher, Distillation tokens to produce estimated labels from teacher, Attention between class and distillation tokens.	2D Image	Class labels	Cross-entropy, Distillation loss based on KL-divergence
	CLIP [81]	Jointly train image and text encoders on image-text pairs, to maximize similarity of valid pairs and minimize otherwise	2D Images & texts	Image-text pairs	Symmetric cross-entropy
Object Detection	DETR [13]	Linear projection layer to reduce CNN feature dimension, Spatial positional embedding added to each multi-head self-attention layer of both encoder and decoder. Object queries (output positional encoding) added to each multi-head self-attention layer of decoder.	2D Image	Class labels	Hungarian loss based on bipartite matching between predicted and ground truths
	D-DETR [14]	Deformable Transformer consists of deformable attention layers to introduce sparse priors in Transformers, Multi-scale attention module.	2D Image	Class labels	Hungarian loss
Low Shot Learning	CT [25]	Self-supervised pretraining, Query-aligned class prototypes that provide spatial correspondence between the support-set images and query image.	2D Image	Pretraining without labels and few-shot learning with Class labels	Normalized Cross-entropy
Image Colorization	ColTran [24]	Conditional Row/column multi-head attention layers, Progressive multi-scale colorization scheme.	2D Image	2D Image	Negative log-likelihood of the images
Action Recognition	ST-TR [164]	Spatial and Temporal self-attention to operates on graph data such as joints in skeletons.	Skeleton	Action Classes	Cross-entropy

Tasks and design choices

Khan et al., 2021

Task	Method	Design Highlights (focus on differences with the standard form)	Input Data Type	Label Type	Loss
Super-resolution	TTSR [16]	Texture enhancing Transformer module, Relevance embeddings to compute the relevance between the low-resolution and reference image.	2D Image	2D Image	Reconstruction loss, Perceptual loss defined on pretrained VGG19 features.
Multi-Model Learning	Oscar [36]	Transformer layer to jointly process triplet representation of image-text [words, tags, features], Masked tokens to represent text data.	2D Image	Captions, Class labels, Object tags	Negative log-likelihood of masked tokens, Contrastive binary cross-entropy
3D Classification/Segmentation	PT [173]	Point Transformer block, Transition down block to reduce cardinality of the point set, Transition up for dense prediction tasks.	CAD models, 3D object part segmentation	Object and shape categories	Cross-entropy
3D Mesh Reconstruction	METRO [37]	Progressive dimensionality reduction across Transformer layers, Positional Encoding with 3D joint and 3D vertex coordinates, Masked vertex/joint modeling.	2D Image	3D Mesh + Human Pose	L_1 loss on mesh vertices and joints in 3D and 2D projection.
Vision and Language Navigation	Chen *et al.* [149]	Uni-modal encoders on language and map inputs followed by a cross-modal transformer, Trajectory position encodings in the map encoder.	Instruction text + RGBD panorama + Topological Environment Map	Navigation Plan	Cross-entropy over nodes and `[stop]` action
Referring Image Segmentation	CMSA [15]	Multimodal feature, Cross-modal self-attention on multiple levels and their fusion using learned gates.	2D Image + Language expression	Segmentation mask	Binary cross-entropy loss
Video Classification	Lee *et al.* [134]	Operates on real-valued audio-visual signals instead of tokens, Contrastive learning for pre-training, End-to-end multimodal transformer learning.	Audio-Visual	Activity labels	Contrastive InfoNCE loss and Binary cross-entropy

Advantages and limitations

Khan et al., 2021

Task	Method	Metric	Dataset	Performance	Highlights	Limitations
Image Classification	ViT [11] ICLR'21	Top-1 Acc.	ImageNet	88.55	**a)** First application of Transformer (global self-attention) directly on image patches, **b)** Convolution-free network architecture, **c)** Outperforms CNN models such as ResNet.	**a)** Requires training on large-scale data *e.g.*, 300-Million images, **b)** Requires careful transfer learning to the new task, **c)** Requires large model with 632-Million parameters to achieve SOTA results.
	DeiT [12] arXiv'20	Top-1 Acc.	ImageNet	83.10	**a)** Successfully trains Transformer on ImageNet only, **b)** Introduces attention-based distillation method. **c)** Produces competitive performance with small (86-Million parameters) Transformers.	**a)** Requires access to pretrained CNN based teacher model thus performance depends on the quality of the teacher model.
Low-Shot Learning	CT [25] NeurIPS'20	Top-1 Acc.	ImageNet COCO	62.25 60.35	**a)** Self-supervised pre-training mechanism that does not need manual labels, **b)** Dynamic inference using Transformer achieving stat-of-the-art results.	Proposed algorithm is limited in its capacity to perform on datasets that lack spatial details such as texture.
Object Detection	DETR [13] ECCV'20	AP	COCO	44.9	**a)** Use of Transformer allows end-to-end training pipeline for object detection, **b)** Removes the need for hand-crafted post-processing steps.	**a)** Performs poorly on small objects, **b)** Requires long training time to converge.
	D-DETR [14] ICLR'21	AP	COCO	43.8	**a)** Achieves better performance on small objects than DETR [13], **b)** Faster convergence than DETR [13]	Obtain SOTA results with 52.3 AP but with two stage detector design and test time augmentations.
Image Colorization	ColTran [24] ICLR'21	FID	ImageNet	19.71	**a)** First successful application of Transformer to image colorization, **b)** Achieves SOTA FID score.	**a)** Lacks end-to-end training, **b)** limited to images of size 256×256.
Action Recognition	ST-TR [164] arXiv'20	Top-1 Acc.	NTU 60/120	94.0/84.7	**a)** Successfully applies Transformer to model relations between body joints both in spatial and temporal domain, **b)** Achieves SOTA results.	Proposed Transformers do not process joints directly rather operate on features extracted by a CNN, thus the overall model is based on hand-crafted design.

Advantages and limitations

Khan et al., 2021

Task	Method	Metric	Dataset	Performance	Highlights	Limitations
Super-Resolution	TTSR [16] CVPR'20	PSNR/ SSIM	CUFED5 Sun80 Urban100 Manga109	27.1 / 0.8 30.0 / 0.81 25.9 / 0.78 30.1 / 0.91	**a)** Achieves state-of-the-art super-resolution by using attention, **b)** Novel Transformer inspired architectures that can process multi-scale features.	**a)** Proposed Transformer does not process images directly but features extracted by a convolution based network, **b)** Model with large number of trainable parameters, and **c)** Compute intensive.
Multi-Model Learning	ViLBERT [133] NeurIPS'19	Acc./ mAP ($R@1$)	VQA [135]/ Retrieval [181]	70.6/ 58.2	**a)** Proposed Transformer architecture can combine text and visual information to understand inter-task dependencies, **b)** Achieves pre-training on unlabelled dataset.	**a)** Requires large amount of data for pre-training, **b)** Requires fine tuning to the new task.
	Oscar [36] ECCV'20	Acc./ mAP ($R@1$)	VQA [182]/ COCO	80.37/57.5	**a)** Exploit novel supervisory signal via object tags to achieve text and image alignment, **b)** Achieves state-of-the-art results.	Requires extra supervision through pre-trained object detectors thus performance is dependent on the quality of object detectors.
	UNITER [35] ECCV'20	Acc./ Avg. ($R@1/5/10$)	VQA [135]/ Flickr30K [183]	72.47/83.72	Learns fine-grained relation alignment between text and images	Requires large multi-task datasets for Transformer training which lead to high computational cost.
3D Analysis	Point Transformer [173] arXiv'20	Top-1 Acc. IoU	ModelNet40 [175]	92.8 85.9	**a)** Transformer based attention capable to process unordered and unstructured point sets, **b)** Permutation invariant architecture.	**a)** Only moderate improvements over previous SOTA, **b)** Large number of trainable parameters around 6× higher than PointNet++ [184].
	METRO [37] arXiv'20	MPJPE PA-MPJPE MPVE	3DPW [178]	77.1 47.9 88.2	**a)** Does not depend on parametric mesh models so easily extendable to different objects, **b)** Achieves SOTA results using Transformers.	Dependent on hand-crafted network design.

Open problems and opportunities

- High computational cost
- Large data requirements
- Large memory requirements

- Vision tailored transformer designs
- Interpretability of transformers
- Hardware efficient designs

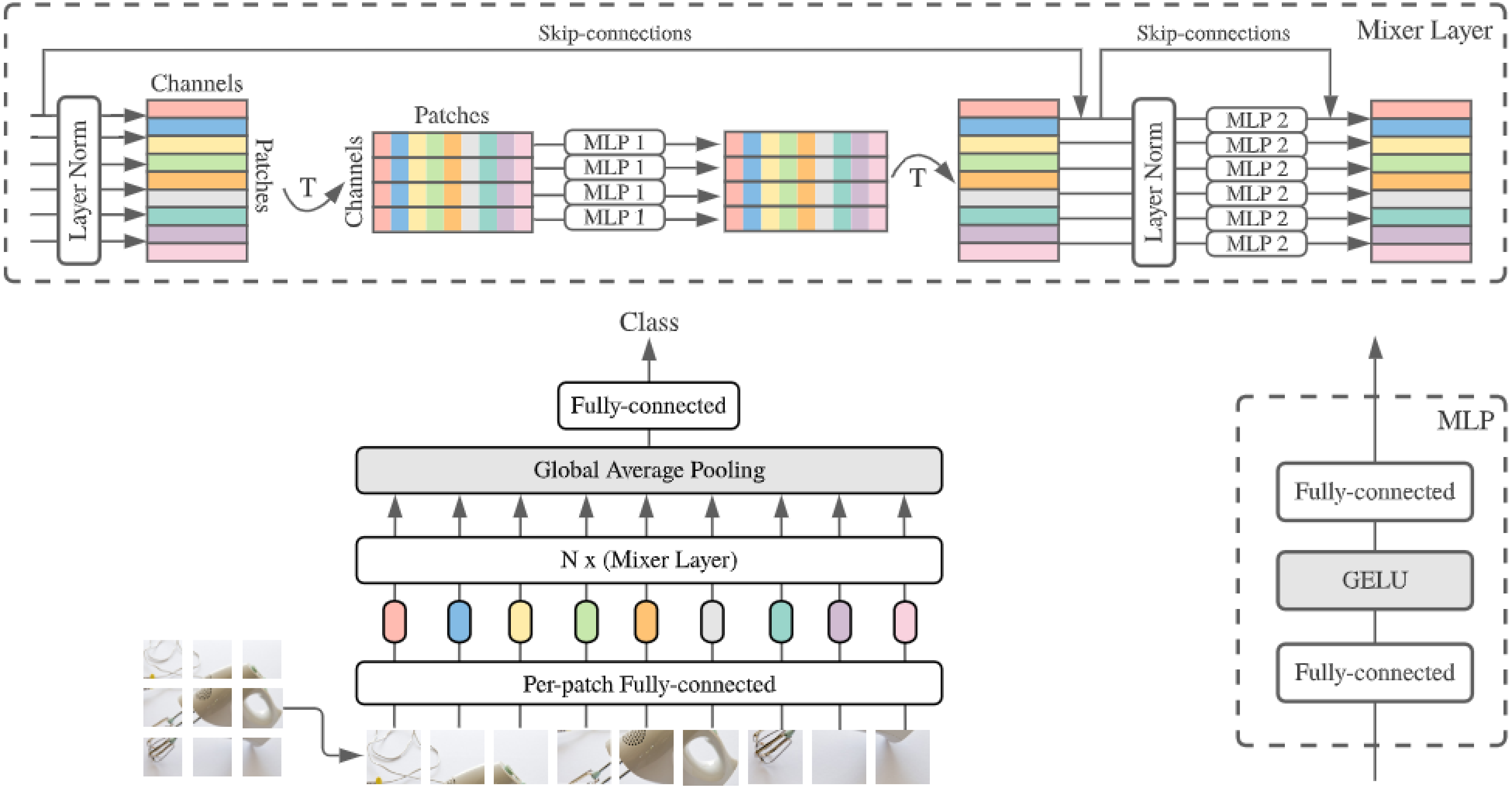
- Combinations with CNNs?
- Inductive bias?

- A plethora of papers published very recently
- SOTA results

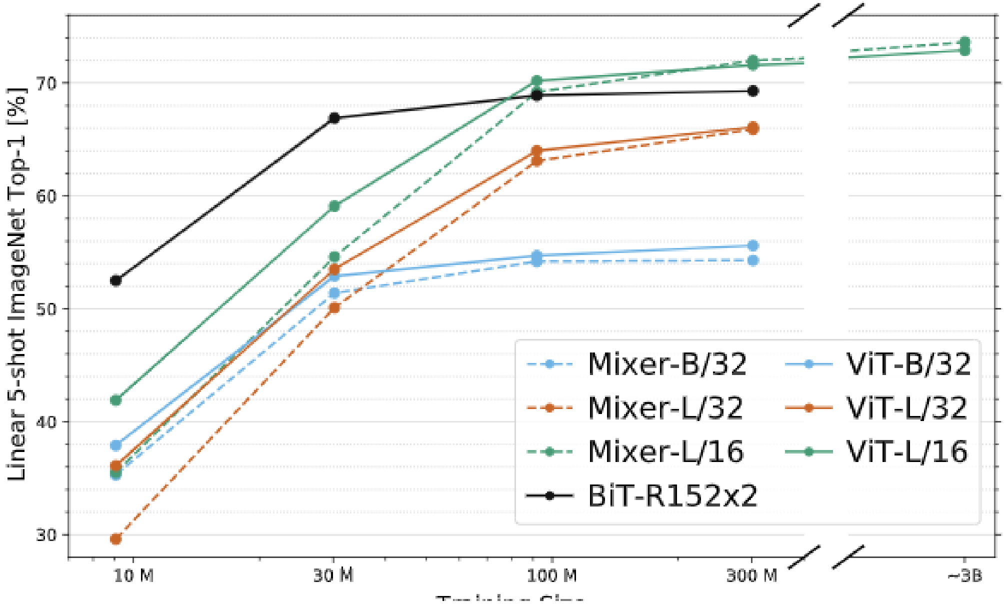
MLP-Mixer: An all-MLP Architecture for Vision

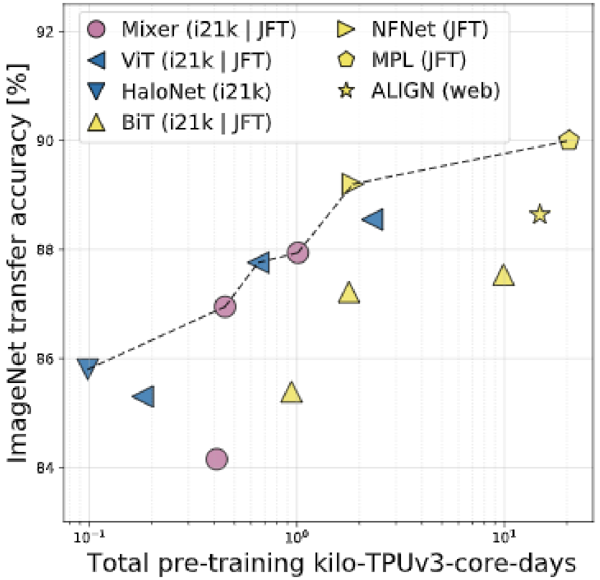
- No convolutions, no attention, only MLPs!

Tolstikhin et al., 2021



MLP-Mixer results



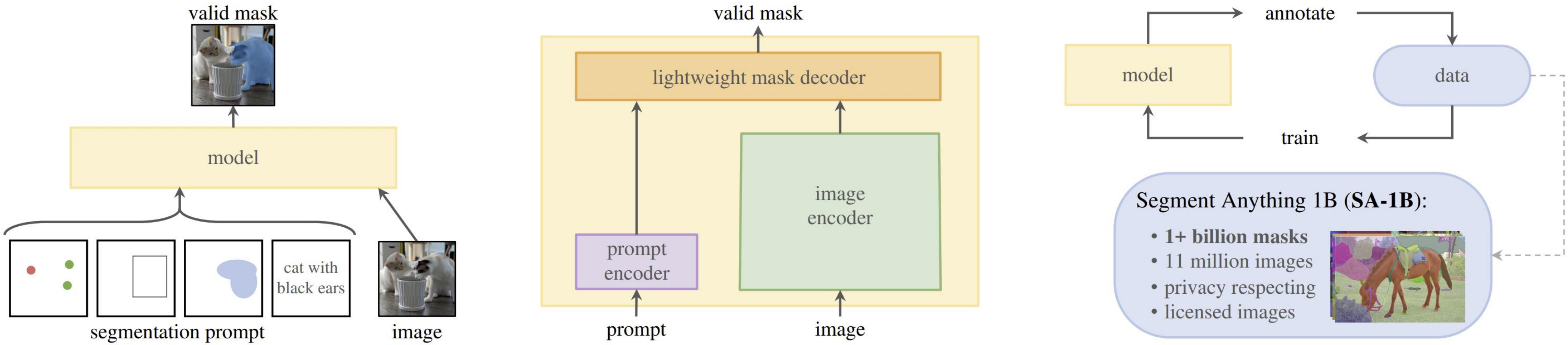


	ImNet top-1	ReaL top-1	Avg 5 top-1	VTAB-1k 19 tasks	Throughput img/sec/core	TPUv3 core-days
Pre-trained on ImageNet-21k (public)						
HaloNet [51]	85.8	—	—	—	120	0.10k
Mixer-L/16	84.15	87.86	93.91	74.95	105	0.41k
ViT-L/16 [14]	85.30	88.62	94.39	72.72	32	0.18k
BiT-R152x4 [22]	85.39	—	94.04	70.64	26	0.94k
Pre-trained on JFT-300M (proprietary)						
NFNet-F4+ [7]	89.2	—	—	—	46	1.86k
Mixer-H/14	87.94	90.18	95.71	75.33	40	1.01k
BiT-R152x4 [22]	87.54	90.54	95.33	76.29	26	9.90k
ViT-H/14 [14]	88.55	90.72	95.97	77.63	15	2.30k
Pre-trained on unlabelled or weakly labelled data (proprietary)						
MPL [34]	90.0	91.12	—	—	—	20.48k
ALIGN [21]	88.64	—	—	79.99	15	14.82k

Specification	S/32	S/16	B/32	B/16	L/32	L/16	H/14
Number of layers	8	8	12	12	24	24	32
Patch resolution $P \times P$	32×32	16×16	32×32	16×16	32×32	16×16	14×14
Hidden size C	512	512	768	768	1024	1024	1280
Sequence length S	49	196	49	196	49	196	256
MLP dimension D_C	2048	2048	3072	3072	4096	4096	5120
MLP dimension D_S	256	256	384	384	512	512	640
Parameters (M)	19	18	60	59	206	207	431

Segment Anything

- New task, model, and dataset for image segmentation
- Foundation model – Segment Anything Model (SAM)



(a) **Task**: promptable segmentation (b) **Model**: Segment Anything Model (**SAM**) (c) **Data**: data engine (top) & dataset (bottom)

[Kirillov et al., 2023]

SAM dataset

- 11M diverse, high-resolution, licensed, and privacy protecting images
- 1.1B high-quality segmentation masks, 99.1% of which generated fully automatically

SAM model

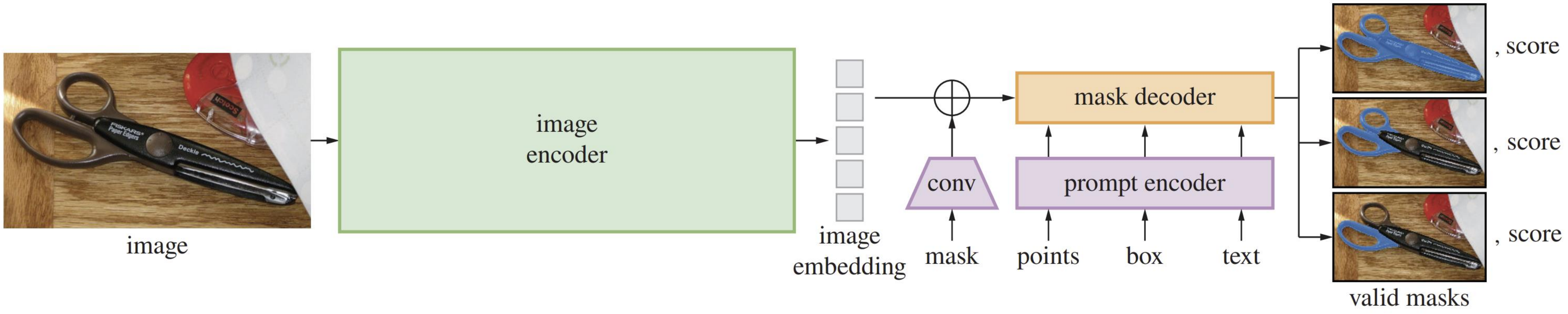


image
image encoder
image embedding
conv
mask
mask decoder
prompt encoder
points
box
text
, score
, score
, score
valid masks

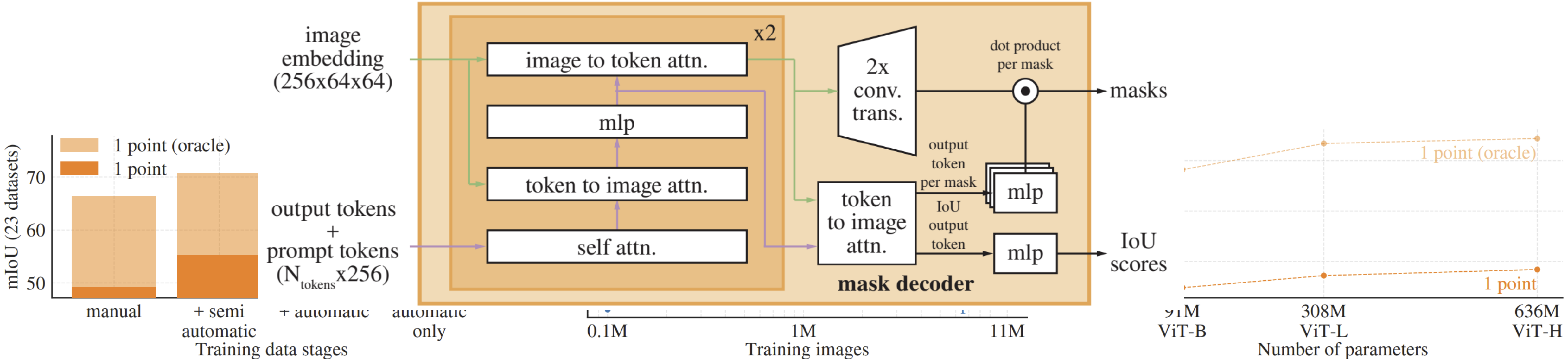
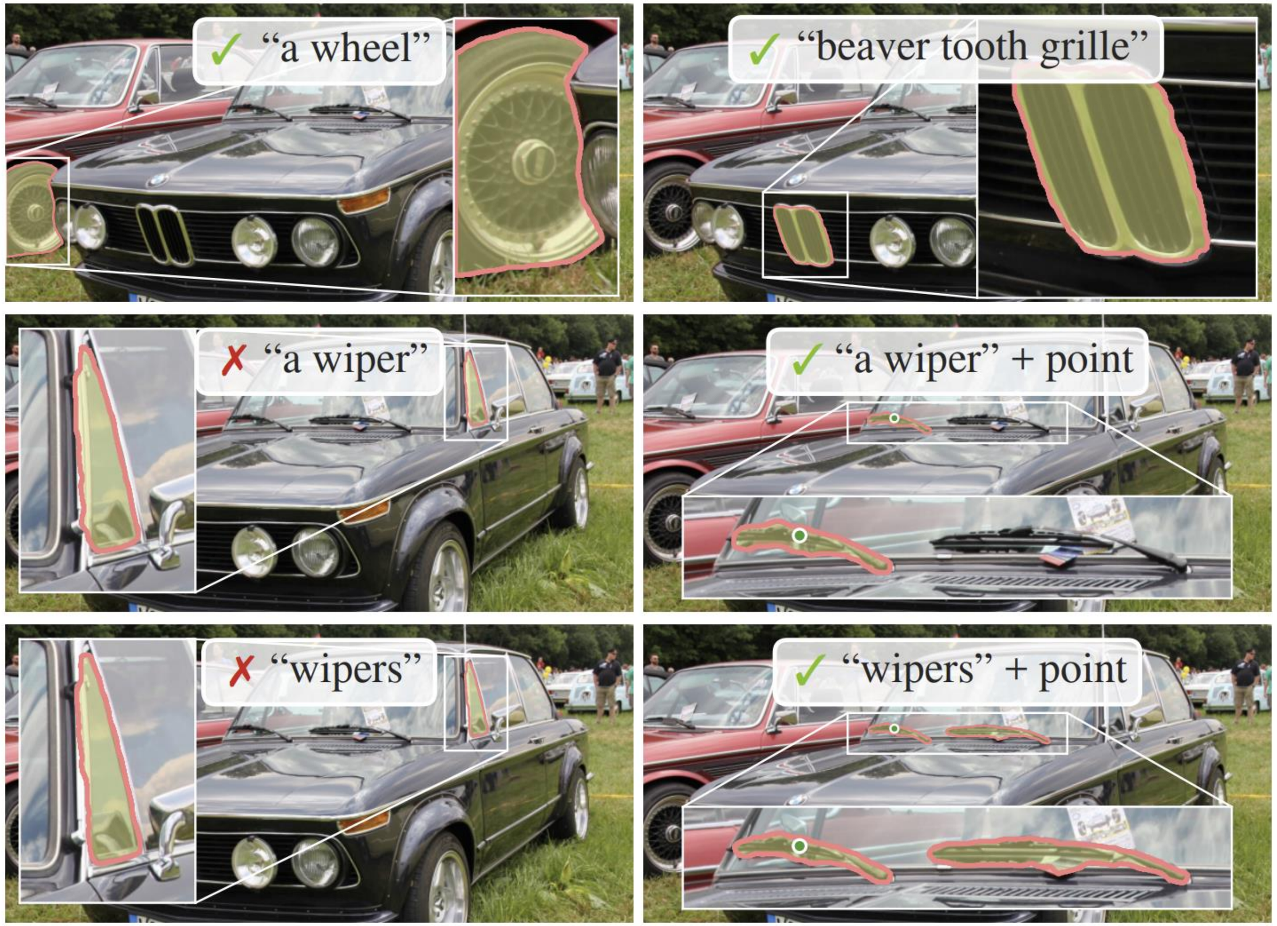


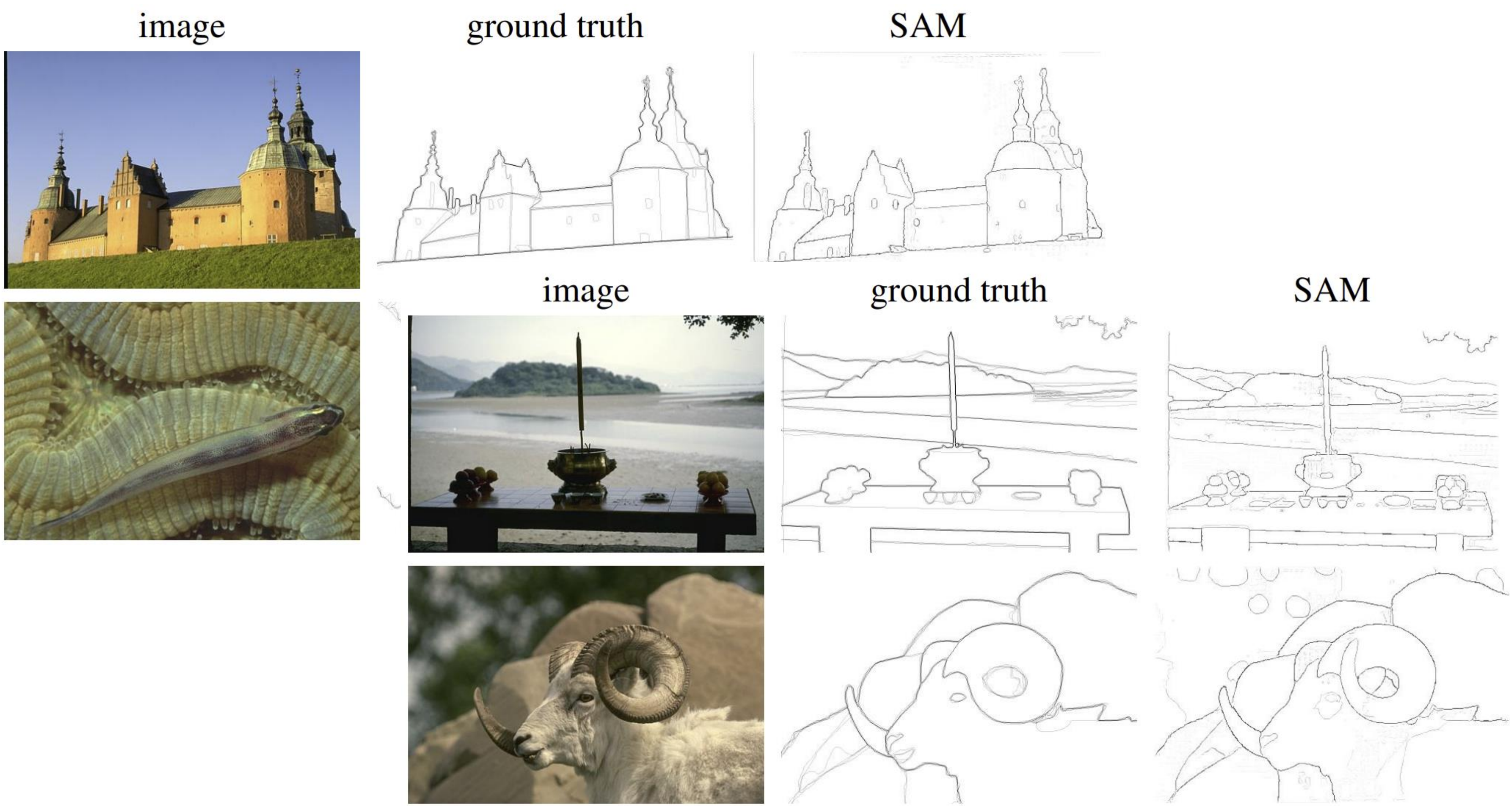
image embedding (256x64x64)
output tokens + prompt tokens (N_{tokens}x256)
x2
image to token attn.
mlp
token to image attn.
self attn.
2x conv. trans.
dot product per mask
masks
output token per mask
IoU output token
token to image attn.
mlp
mlp
IoU scores
mask decoder
mIoU (23 datasets)
1 point (oracle)
1 point
70
60
50
manual
+ semi automatic
only
Training data stages
0.1M
1M
11M
Training images
308M ViT-L
636M ViT-H
ViT-B
Number of parameters
1 point (oracle)
1 point

SAM prompting

- Mask
- Points
- Box
- Text



SAM - Zero-shot edge detection



SAM - Similarities of mask embeddings

SAM - Zero-shot instance segmentation