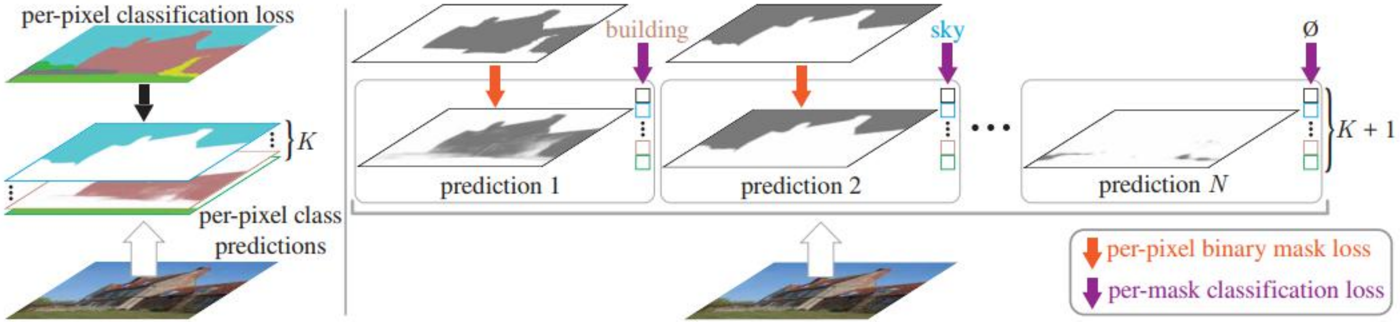
Masked-attention Mask Transformer for Universal Image Segmentation

Bowen Cheng[1,2*] Ishan Misra[1] Alexander G. Schwing[2] Alexander Kirillov[1] Rohit Girdhar[1]
[1]Facebook AI Research (FAIR) [2]University of Illinois at Urbana-Champaign (UIUC)

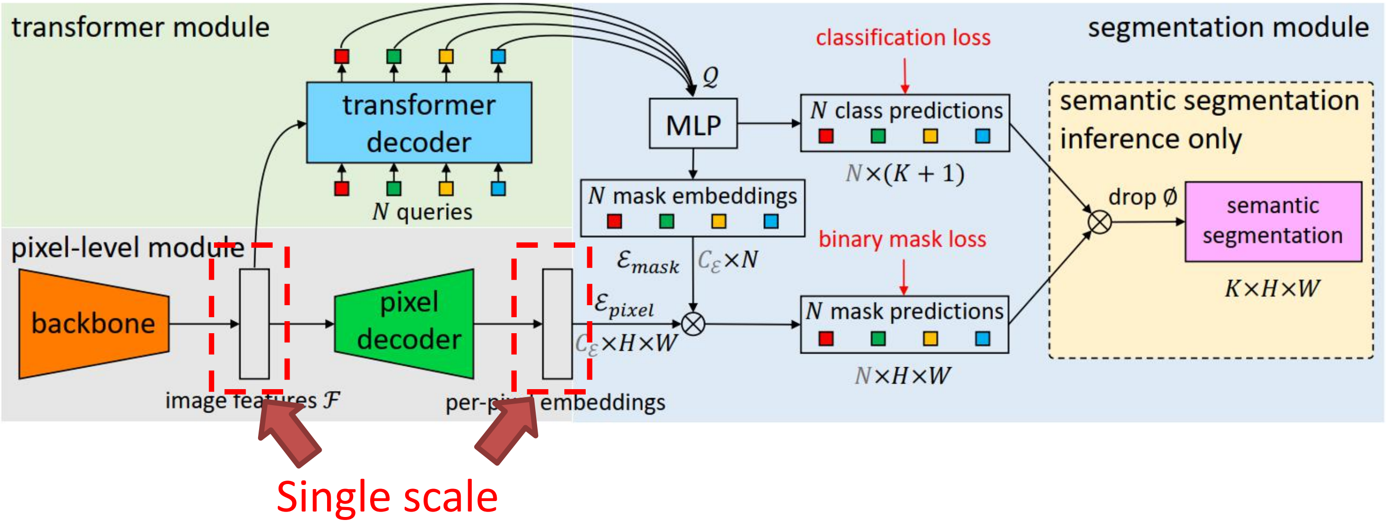
MaskFormer

- MaskFormer employs a Transformer decoder to compute a set of pairs, each consisting of a class prediction and a mask embedding vector.

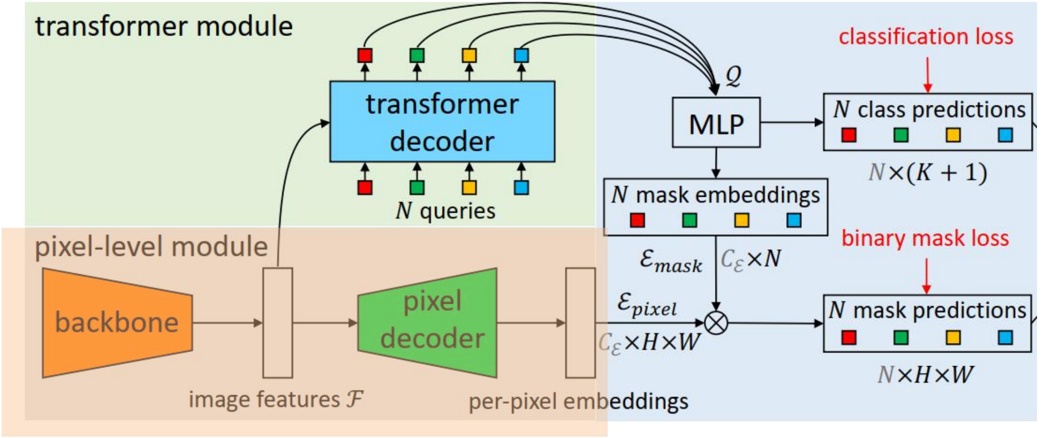


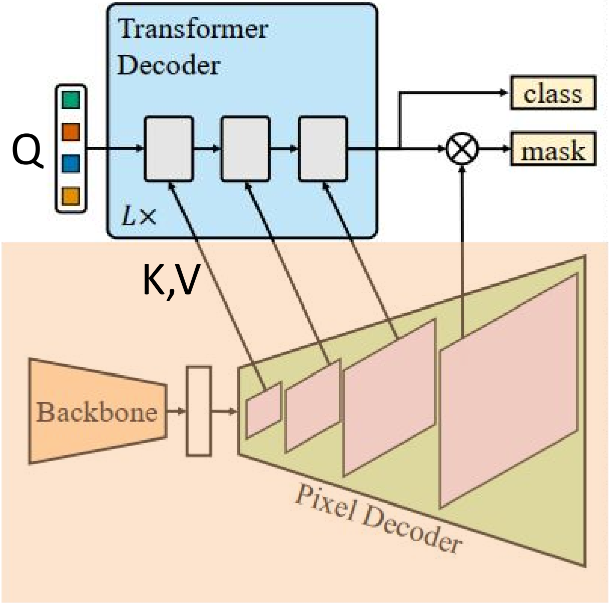
MaskFormer

- The model contains three modules :
- 1**) a pixel-level modul**e extracts per-pixel embeddings used to generate binary mask
- 2) **a transformer module**, computes *N* per-segment embeddings;
- 3) **a segmentation module**



Mask2Former overview

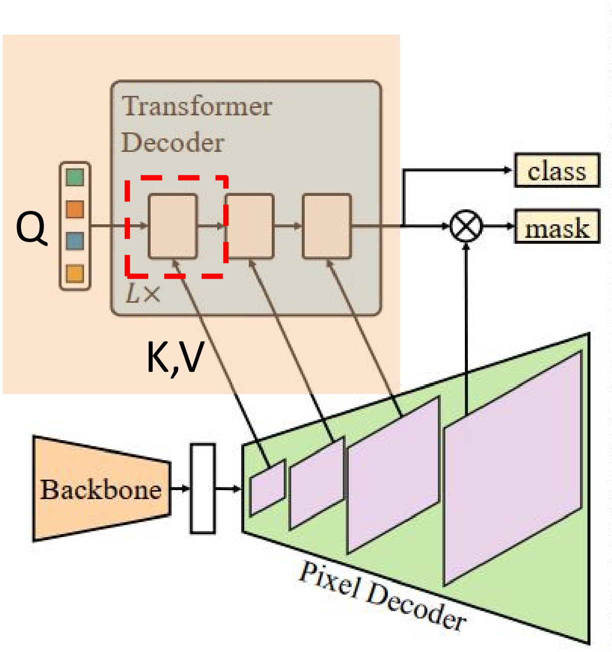


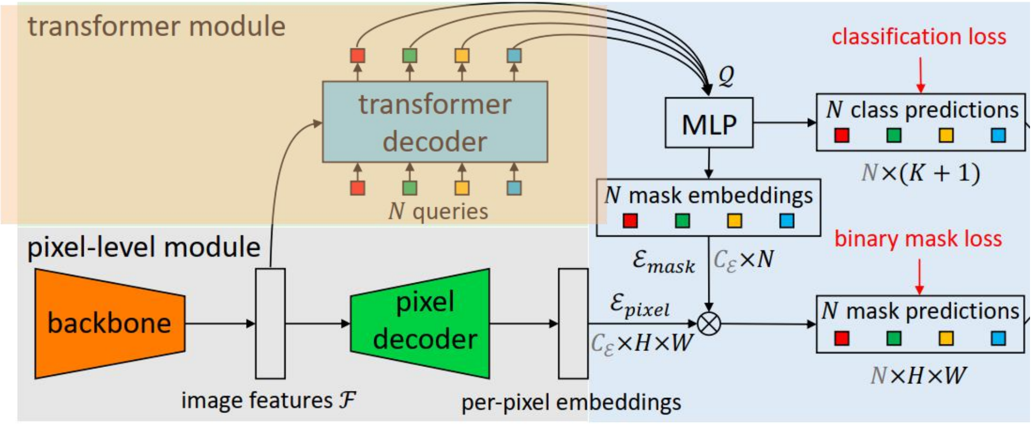


pixel decoder with resolution 1/32, 1/16 and 1/8

High-resolution features improve model performance, especially for small objects

Mask2Former overview





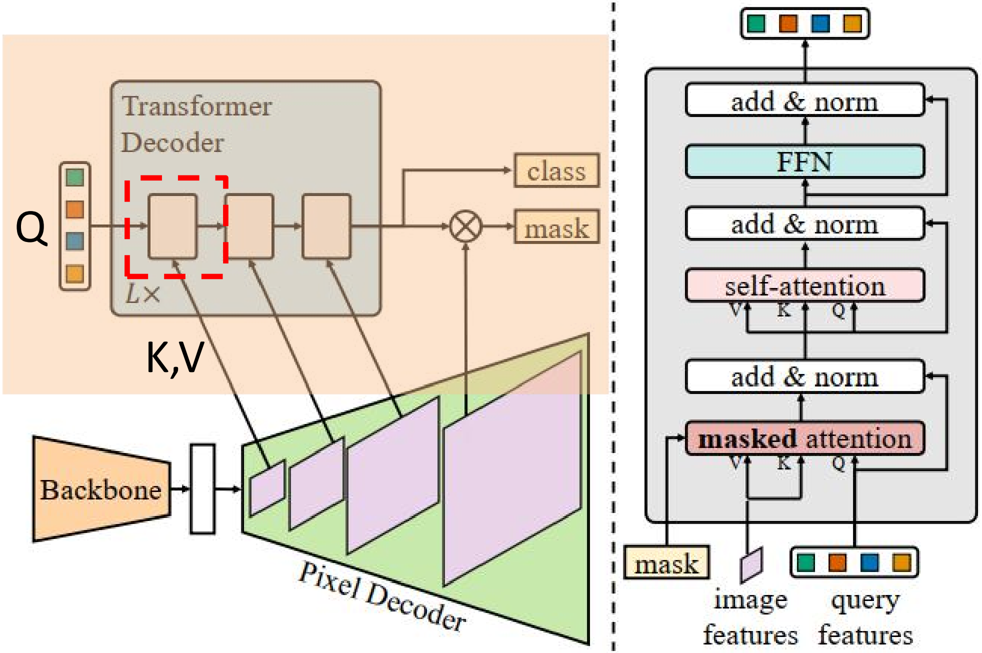
learnable positional embedding

$$e_{\text{pos}} \in \mathbb{R}^{H_l W_l \times C}$$

learnable scale-level embedding

$$e_{\text{lvl}} \in \mathbb{R}^{1 \times C}$$

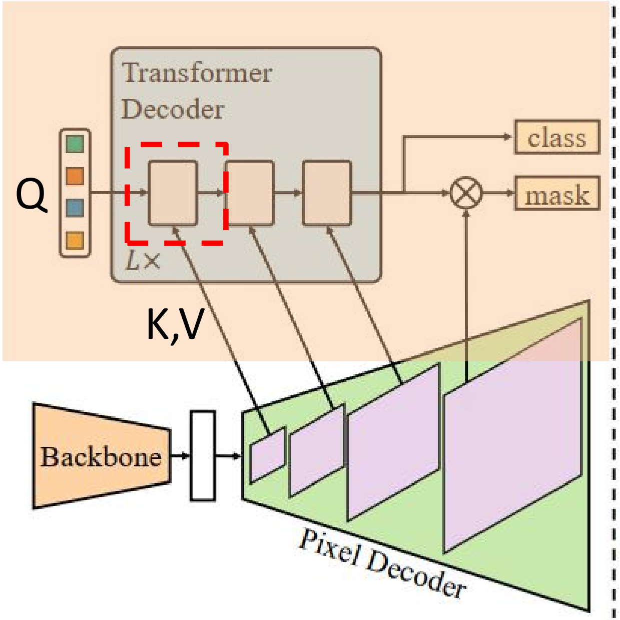
Mask2Former overview

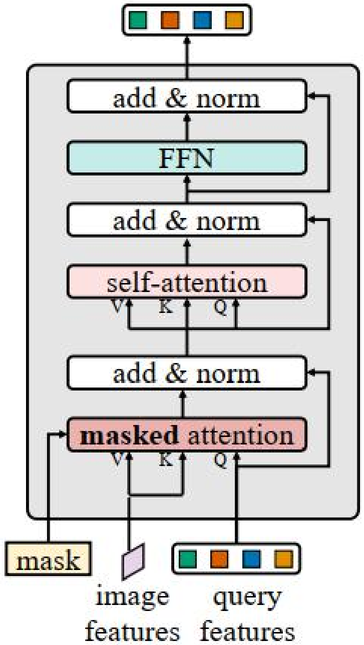


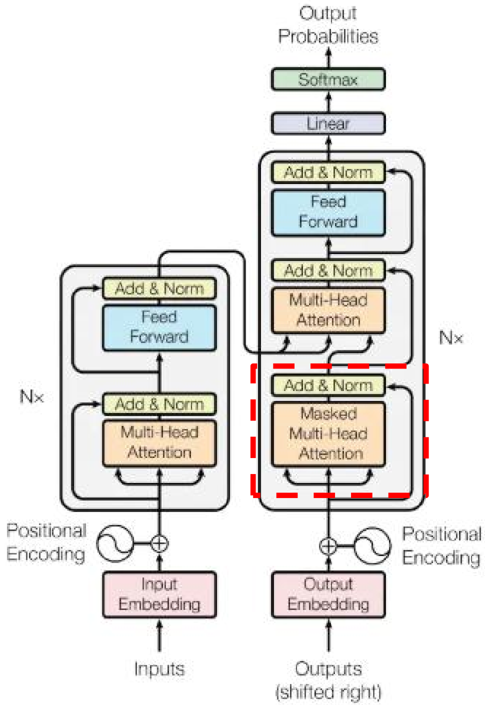
Deformable detr

6 MSDeformAttn layers

Mask2Former overview







switch the order of self and cross-attention

Masked attention

Standard cross-attention

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

$$\mathbf{Q}_l = f_Q(\mathbf{X}_{l-1}) \in \mathbb{R}^{N\times C}$$

$\mathbf{K}_l, \mathbf{V}_l \leftarrow f_K(\cdot)$ and $f_V(\cdot)$ with feature

query features (**X**0) are *zero initialized*

masked attention modulates

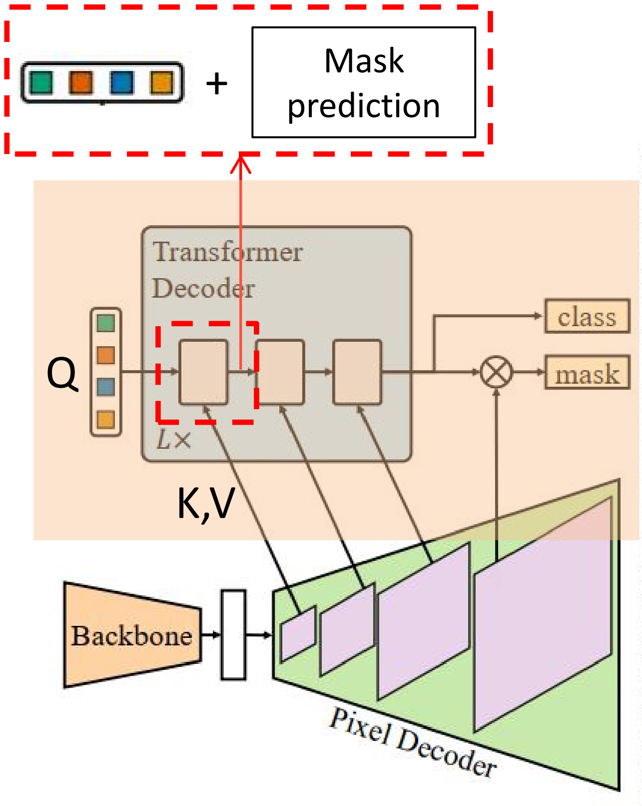
$$\mathbf{X}_l = \text{softmax}(\mathcal{M}_{l-1} + \mathbf{Q}_l\mathbf{K}_l^{\mathrm{T}})\mathbf{V}_l + \mathbf{X}_{l-1}.$$

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

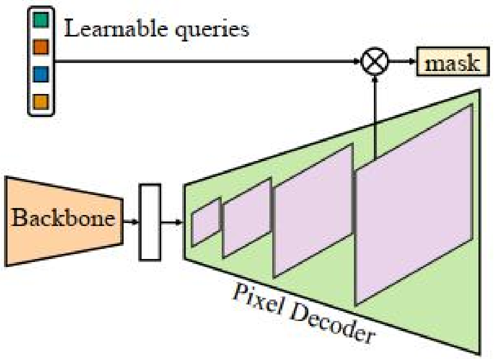
only attends within the foreground region of the predicted mask

M0 is the binary mask prediction obtained from **X**0

Mask2Former overview



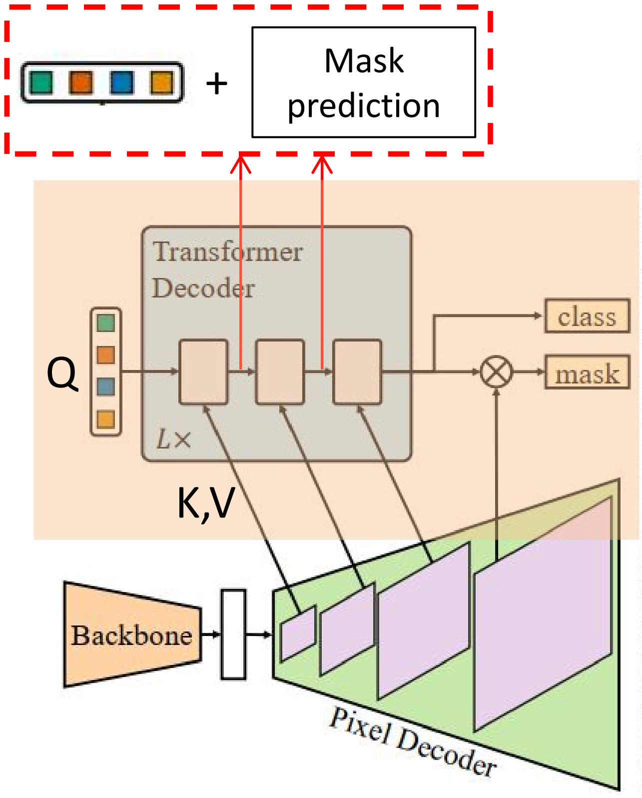
Mask prediction



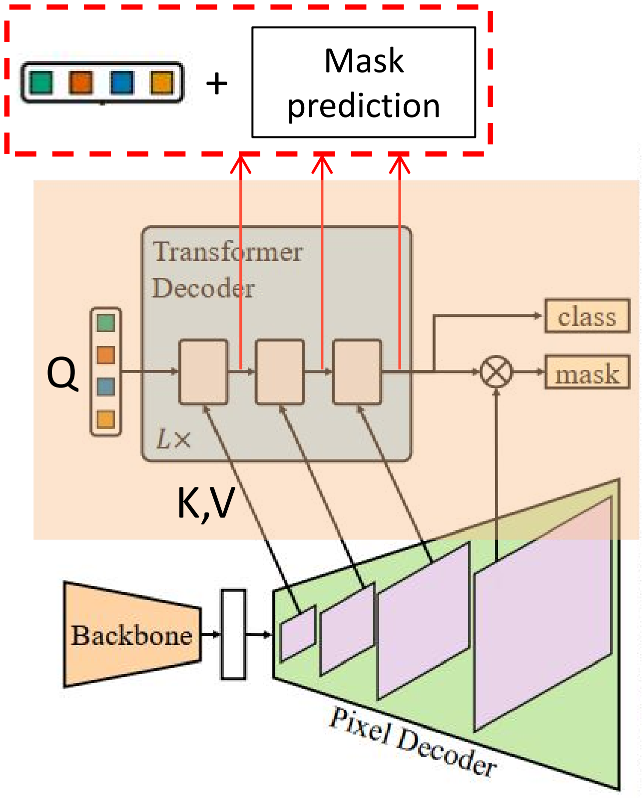
Than resized to the same resolution

these learnable query features function like a region proposal network and have the ability to generate mask proposals.

Mask2Former overview



Mask2Former overview



We repeat this 3-layer Transformer decoder L times

L = 3, 100 queries

Improving training efficiency

- trained with its mask loss calculated on *K* randomly sampled points instead of the whole mask (PointRend)

$$K = 12544, \textit{i.e.}, 112 \times 112 \text{ points.}$$

- mask loss with sampled points from 18GB to 6GB per image

matching loss： uniformly sampling
final loss ： importance sampling

Loss weights

- We use the binary cross-entropy loss (instead of focal) and the dice loss for our mask loss:

$$\mathcal{L}_{\text{mask}} = \lambda_{\text{ce}}\mathcal{L}_{\text{ce}} + \lambda_{\text{dice}}\mathcal{L}_{\text{dice}}.$$

$$\lambda_{\text{dice}} = 5.0. \quad \lambda_{\text{ce}} = 5.0$$

- The final loss is a combination of mask loss and classification loss

$$\mathcal{L}_{\text{mask}} + \lambda_{\text{cls}}\mathcal{L}_{\text{cls}}$$

$$\lambda_{\text{cls}} = 2.0$$

Panoptic segmentation on COCO panoptic `val2017` with 133 categories.

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

Instance segmentation on COCO `val2017` with 80 categories.

data augmentation
training strategies and model scaling

method	backbone	query type	epochs	AP	AP^S	AP^M	AP^L	$AP^{boundary}$	#params.	FLOPs	fps
MaskFormer [14]	R50	100 queries	300	34.0	16.4	37.8	54.2	23.0	45M	181G	19.2
Mask R-CNN [24]	R50	dense anchors	36	37.2	18.6	39.5	53.3	23.1	44M	201G	15.2
Mask R-CNN [18, 23, 24]	R50	dense anchors	400	42.5	**23.8**	45.0	60.0	28.0	46M	358G	10.3
Mask2Former (ours)	R50	100 queries	50	**43.7**	23.4	**47.2**	**64.8**	**30.6**	44M	226G	9.7
Mask R-CNN [24]	R101	dense anchors	36	38.6	19.5	41.3	55.3	24.5	63M	266G	10.8
Mask R-CNN [18, 23, 24]	R101	dense anchors	400	43.7	**24.6**	46.4	61.8	29.1	65M	423G	8.6
Mask2Former (ours)	R101	100 queries	50	**44.2**	23.8	**47.7**	**66.7**	**31.1**	63M	293G	7.8
QueryInst [20]	Swin-L†	300 queries	50	48.9	30.8	52.6	68.3	33.5	-	-	3.3
Swin-HTC++ [6, 36]	Swin-L†	dense anchors	72	49.5	**31.0**	52.4	67.2	34.1	284M	1470G	-
Mask2Former (ours)	Swin-L†	*200 queries*	*100*	**50.1**	29.9	**53.9**	**72.1**	**36.2**	216M	868G	4.0

Semantic segmentation on ADE20K `val`

method	backbone	crop size	mIoU (s.s.)	mIoU (m.s.)
MaskFormer [14]	R50	512	44.5	46.7
Mask2Former (ours)	R50	512	**47.2**	**49.2**
Swin-UperNet [36, 58]	Swin-T	512	-	46.1
MaskFormer [14]	Swin-T	512	46.7	48.8
Mask2Former (ours)	Swin-T	512	**47.7**	**49.6**
MaskFormer [14]	Swin-L†	640	54.1	55.6
FaPN-MaskFormer [14, 39]	Swin-L-FaPN†	640	55.2	56.7
BEiT-UperNet [2, 58]	BEiT-L†	640	-	57.0
Mask2Former (ours)	Swin-L†	640	56.1	57.3
	Swin-L-FaPN†	640	**56.4**	**57.7**

Table 3. **Semantic segmentation on ADE20K `val` with 150 categories.** Mask2Former consistently outperforms MaskFormer [14] by a large margin with different backbones (all Mask2Former models use MSDeformAttn [66] as pixel decoder, except Swin-L-FaPN uses FaPN [39]). Our best model outperforms the best specialized model, BEiT [2]. We report both single-scale (s.s.) and multi-scale (m.s.) inference results. Backbones pre-trained on ImageNet-22K are marked with †.

Ablation studies

	AP	PQ	mIoU	FLOPs
Mask2Former (ours)	**43.7**	**51.9**	**47.2**	226G
— masked attention	37.8 (-5.9)	47.1 (-4.8)	45.5 (-1.7)	213G
— high-resolution features	41.5 (-2.2)	50.2 (-1.7)	46.1 (-1.1)	218G

(a) Masked attention and high-resolution features (from efficient multi-scale strategy) lead to the most gains. More detailed ablations are in Table 4c and Table 4d. We remove one component at a time.

	AP	PQ	mIoU	FLOPs
cross-attention	37.8	47.1	45.5	213G
SMCA [22]	37.9	47.2	<u>46.6</u>	213G
mask pooling [62]	<u>43.1</u>	<u>51.5</u>	46.0	217G
masked attention	**43.7**	**51.9**	**47.2**	226G

(c) **Masked attention.** Our masked attention performs better than other variants of cross-attention across all tasks.

	AP	PQ	mIoU	FLOPs
single scale (1/32)	41.5	50.2	46.1	218G
single scale (1/16)	43.0	51.5	46.5	222G
single scale (1/8)	**44.0**	<u>51.8</u>	**47.4**	239G
naïve m.s. (3 scales)	**44.0**	**51.9**	46.3	247G
efficient m.s. (3 scales)	<u>43.7</u>	**51.9**	<u>47.2</u>	226G

(d) **Feature resolution.** High-resolution features (single scale 1/8) are important. Our efficient multi-scale (efficient m.s.) strategy effectively reduces the FLOPs.

Ablation studies

	AP	PQ	mIoU	FLOPs
FPN [33]	41.5	50.7	45.6	195G
Semantic FPN [27]	42.1	51.2	46.2	258G
FaPN [39]	42.4	51.8	46.8	-
BiFPN [47]	43.5	51.8	45.6	204G
MSDeformAttn [66]	**43.7**	**51.9**	**47.2**	226G

(e) **Pixel decoder.** MSDeformAttn [66] consistently performs the best across all tasks.

matching loss	training loss	AP (COCO)	PQ (COCO)	mIoU (ADE20K)	memory (COCO)
mask	mask	41.0	50.3	45.9	18G
	point	41.0	50.8	45.9	6G
point (ours)	mask	43.1	51.4	**47.3**	18G
	point (ours)	**43.7**	**51.9**	47.2	6G

Table 5. **Calculating loss with points *vs*. masks.** Training with point loss reduces training memory without influencing the performance. Matching with point loss further improves performance.

groundtruth groundtruth