Masked-attention Mask Transformer for Universal Image Segmentation (Mask2Former)

Presented by Louie Lu, Chongyi Zheng, Mingcheng Hu

Motivation

Motivation

- How to unify 3 image segmentation tasks in 1 architecture?
  - Semantic / Instance / Panoptic Segmentation

Motivation

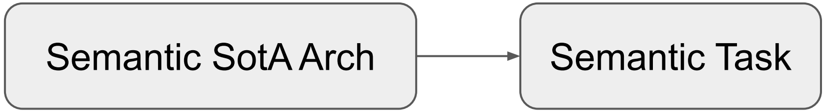
- How to unify 3 image segmentation tasks in 1 architecture?
    - Semantic / Instance / Panoptic Segmentation

Motivation

- How to unify 3 image segmentation tasks in 1 architecture?
  - Semantic / Instance / Panoptic Segmentation
- Why unify in one architecture?
  - State-of-the-arts (SotA) are 3 different specialized model in those tasks.
  - And, training **three (3)** specialized model takes **time** and **resources**!

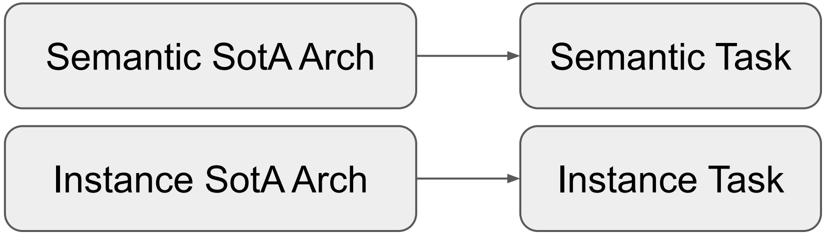
Motivation

- How to unify 3 image segmentation tasks in 1 architecture?
    - Semantic / Instance / Panoptic Segmentation
- Why unify in one architecture?
    - State-of-the-arts (SotA) are 3 different specialized model in those tasks.
    - And, training **three (3)** specialized model takes **time** and **resources**!



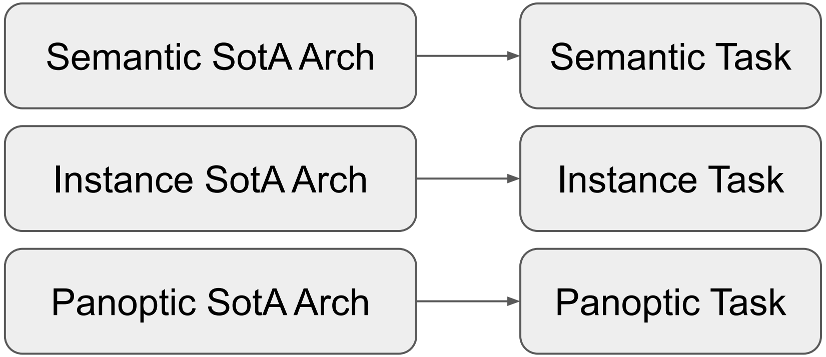
Motivation

- How to unify 3 image segmentation tasks in 1 architecture?
  - Semantic / Instance / Panoptic Segmentation
- Why unify in one architecture?
  - State-of-the-arts (SotA) are 3 different specialized model in those tasks.
  - And, training **three (3)** specialized model takes **time** and **resources**!



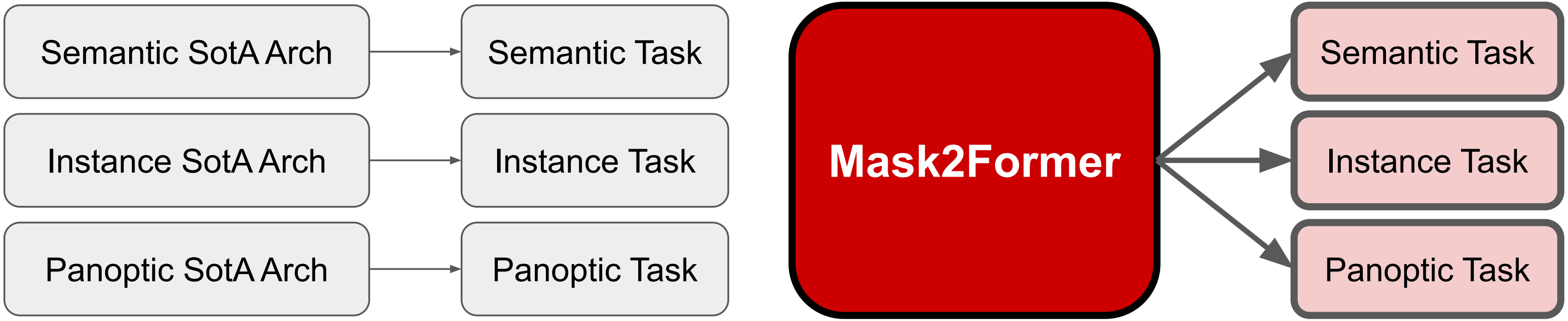
Motivation

- How to unify 3 image segmentation tasks in 1 architecture?
  - Semantic / Instance / Panoptic Segmentation
- Why unify in one architecture?
  - State-of-the-arts (SotA) are 3 different specialized model in those tasks.
  - And, training **three (3)** specialized model takes **time** and **resources**!



Motivation

- How to unify 3 image segmentation tasks in 1 architecture?
  - Semantic / Instance / Panoptic Segmentation
- Why unify in one architecture?
  - State-of-the-arts (SotA) are 3 different specialized model in those tasks.
  - And, training **three (3)** specialized model takes **time** and **resources**!
  - We want to **train once, use it anywhere!**

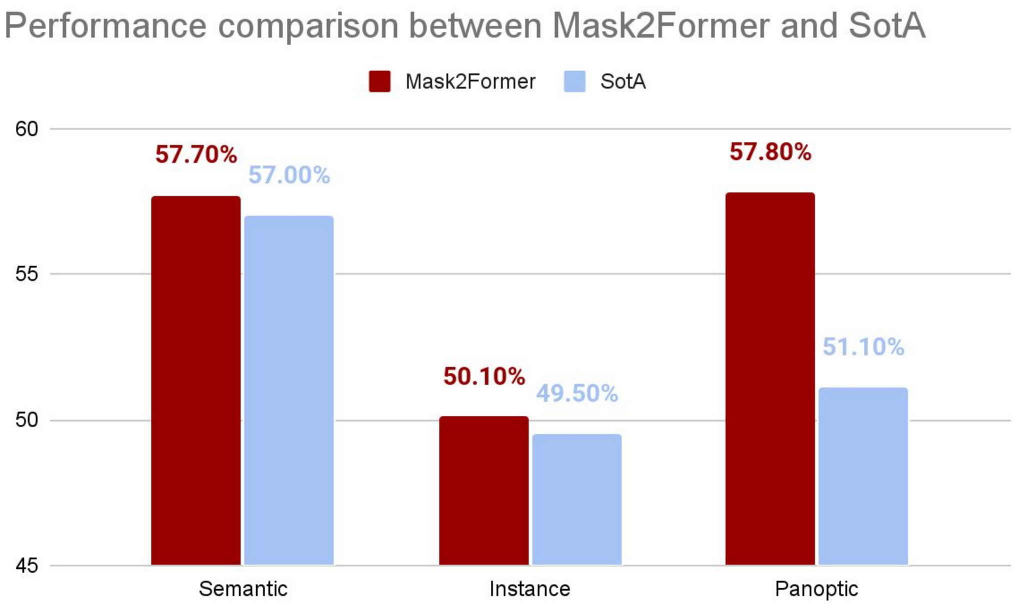


Motivation

- How to unify 3 image segmentation tasks in 1 architecture?
  - Semantic / Instance / Panoptic Segmentation
- Why unify in one architecture?
  - State-of-the-arts (SotA) are 3 different specialized model in those tasks.
  - And, training **three (3)** specialized model takes **time** and **resources**!
  - We want to **train once, use it anywhere!**
- This architecture should replace specialized models
  - Performance of this unified model should surpass specialized models!

Motivation

- This architecture should replace specialized models
  - Performance of this unified model should surpass specialized models!



Background

Thing and Stuff

- **Thing**
  - Objects with well-defined shape (e.g. bikes)
- **Stuff**
  - Amorphous background regions (e.g. wall, building)

Definition reference:
COCO-Stuff: Thing and Stuff Classes in Context

Bike rack in front of Carroll Hall. By Louie Lu, all rights reserved.

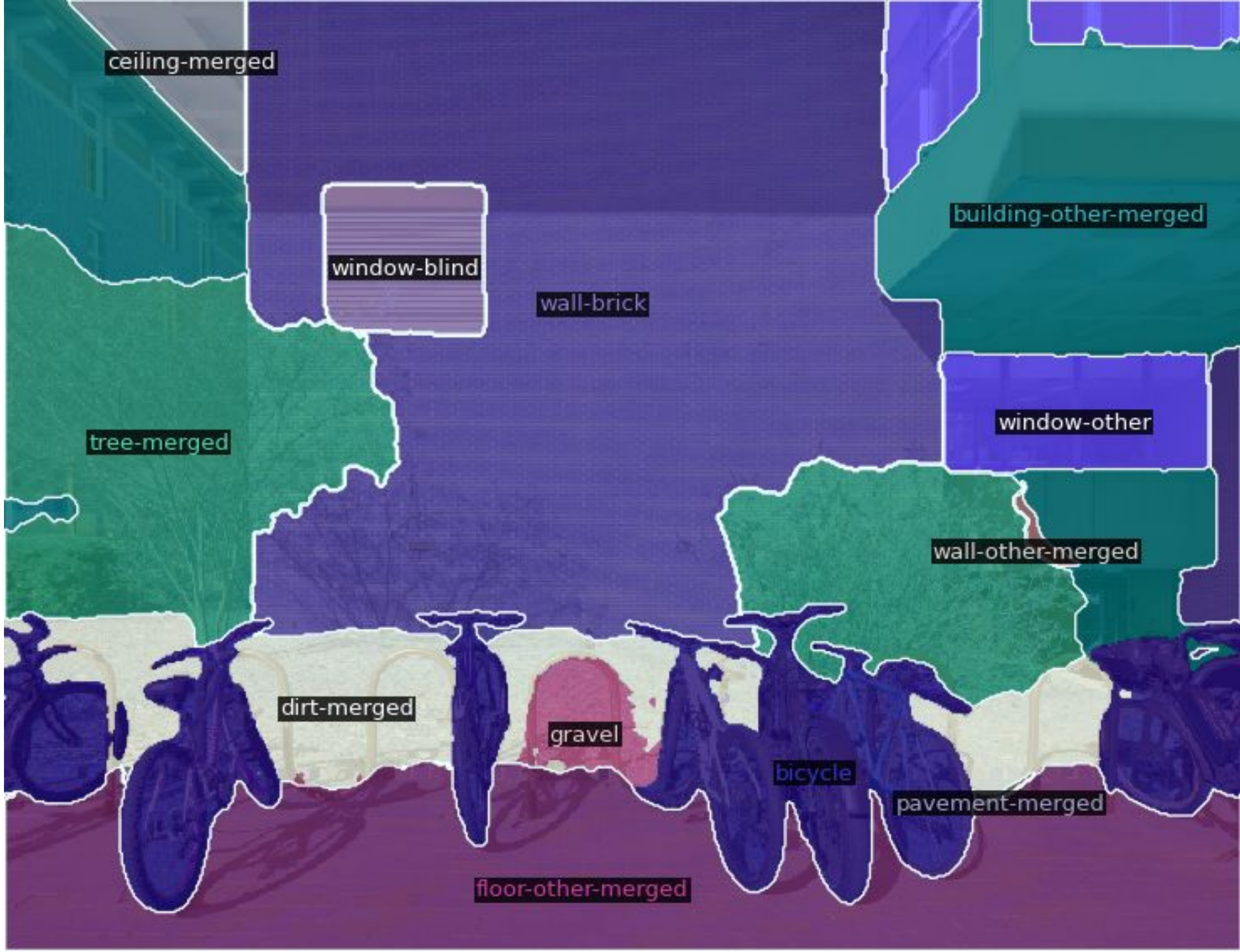
Semantic / Instance / Panoptic Segmentation

- **Quick poll**, raise your hand if you think you know the difference between:
    - Semantic
    - Instance
    - Panoptic

Bike rack in front of Carroll Hall. By Louie Lu, all rights reserved.

Semantic

- **Per-pixel classification problem**
- Group same classes
  - i.e. Only one (1) bicycle in the prediction result



Bike rack in front of Carroll Hall, applied semantic prediction with Mask2Former. By Louie Lu, all rights reserved.

Instance

- **Mask classification problem**
- Unique mask
  - i.e. Different bicycle will have its own mask.

Bike rack in front of Carroll Hall, applied instance prediction with Mask2Former. By Louie Lu,

Panoptic

- **Unify semantic & instance tasks**
- Unique mask both on:
  - Things: bike…
  - Stuff: wall, tree…

Bike rack in front of Carroll Hall, applied panoptic prediction with Mask2Former. By Louie Lu,

Goal of Mask2Former

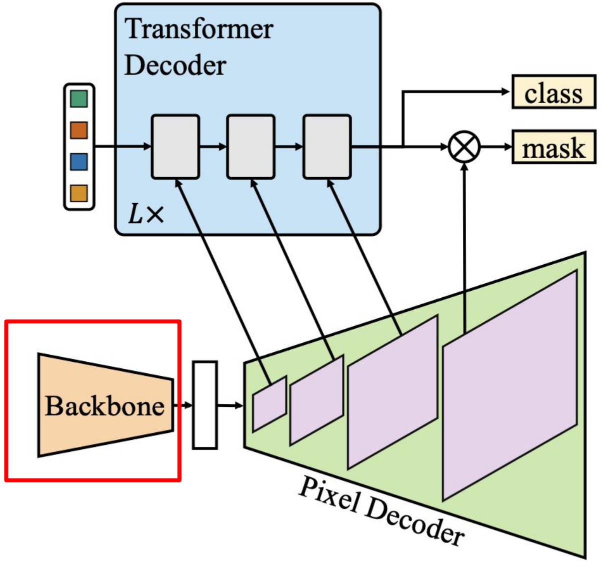
- One Model Rule Them All
- Performance surpass specialized models

Bike rack in front of Carroll Hall, with Mask2Former (semantic, instance, panoptic). By Louie Lu, all rights reserved.

Introducing Mask2Former

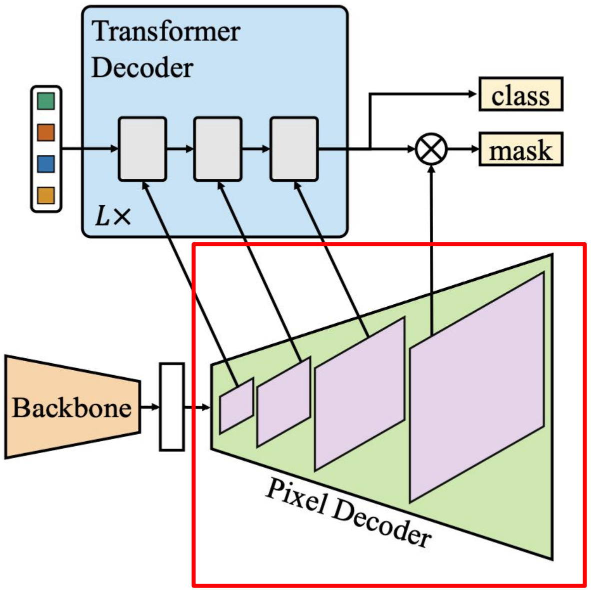
Architecture

- **Backbone**
  extract low-resolution features using a vision model, e.g. Swin Transformer



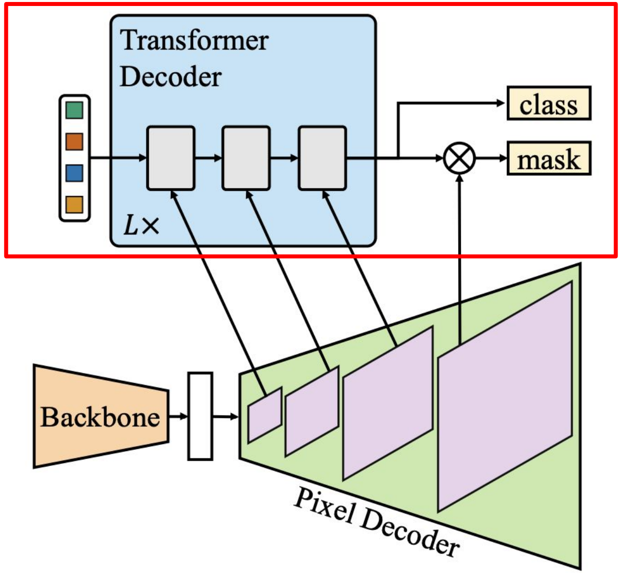
Architecture

- **Backbone**
  extract low-resolution features using a vision model, e.g. Swin Transformer
- **Pixel Decoder**
  Gradually upsamples low-resolution feature to generate high-resolution per-pixel embeddings



Architecture

- **Backbone**
  extract low-resolution features using a vision model, e.g. Swin Transformer
- **Pixel Decoder**
  Gradually upsamples low-resolution feature to generate high-resolution per-pixel embeddings
- **Transformer Decoder**
  Operates on image features to process object queries
  Includes ***masked attention***



Architecture

Standard cross-attention

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

Masked attention

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

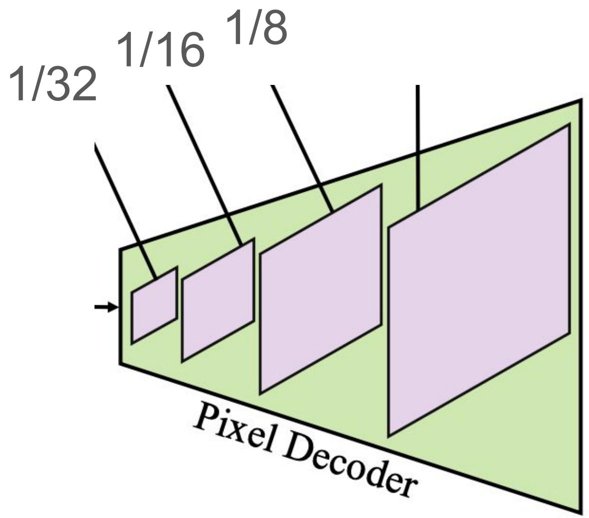
$$\boldsymbol{\mathcal{M}}_{l-1}(x, y) = \begin{cases} 0 & \text{if } \mathbf{M}_{l-1}(x, y) = 1 \\ -\infty & \text{otherwise} \end{cases}$$

is the attention mask at feature location (x, y)

Architecture

Balance computation and performance

Use **feature pyramid** to introduce high-resolution features



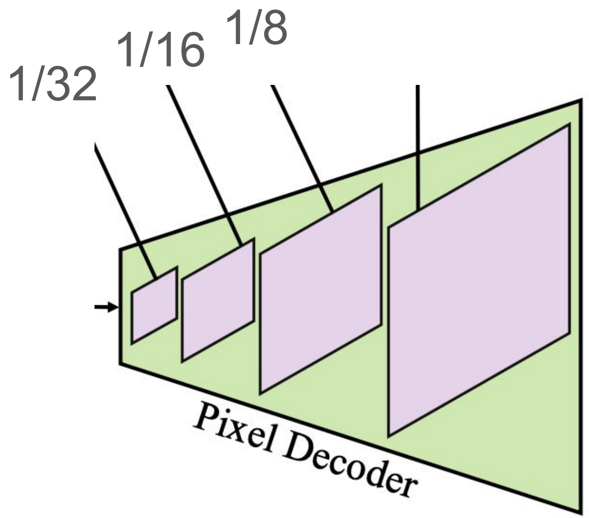
Architecture

Balance computation and performance

Use **feature pyramid** to introduce high-resolution features

Each resolution

- sinusoidal positional embedding $e_{\text{pos}} \in \mathbb{R}^{H_l W_l \times C}$
- learnable scale-level embedding $e_{\text{lvl}} \in \mathbb{R}^{1 \times C}$



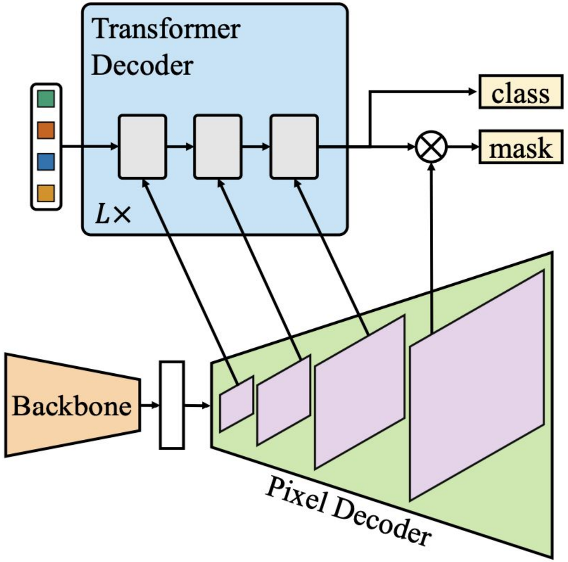
Architecture

Balance computation and performance

Use **feature pyramid** to introduce high-resolution features

Each resolution

- sinusoidal positional embedding $e_{\text{pos}} \in \mathbb{R}^{H_l W_l \times C}$
- learnable scale-level embedding $e_{\text{lvl}} \in \mathbb{R}^{1 \times C}$

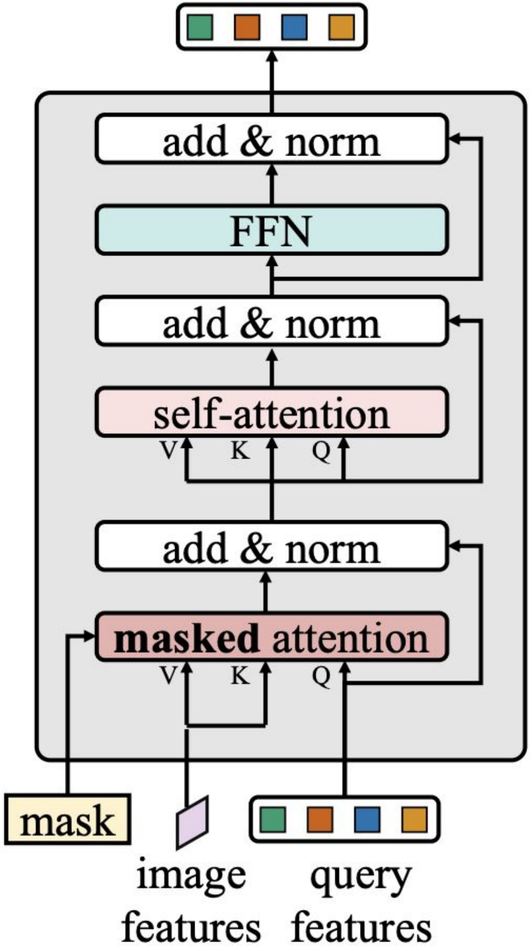


Optimization

Masked attention (modified cross-attention first), then self-attention layer

Make query features learnable

Remove dropout



Optimization

Mask loss

- **Matching loss**: sample same K points for prediction and ground truth masks

Optimization

Mask loss

- **Matching loss**: sample same K points for prediction and ground truth masks
- **Final loss**: sample different K points for different prediction/ground truth pairs using *importance sampling*

Optimization

Mask loss

- **Matching loss**: sample same K points for prediction and ground truth masks
- **Final loss**: sample different K points for different prediction/ground truth pairs using *importance sampling*

Training memory reduced from 18GB to 6GB per image

Experiments

Datasets

- **COCO**
- **ADE20K**
- **Cityscapes**
- **Mapillary Vistas**

Instance segmentation on the COCO dataset

Semantic segmentation on the ADE20K dataset Second and fourth columns are predictions

Evaluation Metrics

Task	Evaluate on	Metric
Panoptic Segmentation	Things and Stuff	$PQ = \frac{\sum_{(p,g)\in TP} IOU(p,g)}{\lvert TP\rvert + \frac{1}{2}\lvert FP\rvert + \frac{1}{2}\lvert FN\rvert} = \underbrace{\frac{\sum_{(p,g)\in TP} IOU(p,g)}{\lvert TP\rvert}}_{\text{Segmentation quality}} \cdot \underbrace{\frac{\lvert TP\rvert}{\lvert TP\rvert + \frac{1}{2}\lvert FP\rvert + \frac{1}{2}\lvert FN\rvert}}_{\text{Recognition Quality}}$ AP^{Th}_{pan} $mIOU_{pan}$
Semantic Segmentation	Things and Stuff	$mIOU$ mean IOU
Instance Segmentation	Things	AP average precision

Implementation Details

- Pixel decoder: We use advanced multiscale deformable attention Transformer (MSDeformAttn) with 6 layers as our default pixel decoder, applied to feature maps with resolution 1/8, 1/16 and 1/32
- Transformer decoder: Proposed previously with L=3 (9 layers in total)
- Loss weights: Combination of mask loss and classification loss

$\mathcal{L}_{total} = \mathcal{L}_{mask} + 2\mathcal{L}_{cls}$, where $\mathcal{L}_{mask} = 5\mathcal{L}_{ce} + 5\mathcal{L}_{dice}$

Training Settings

- Follow the updated Mask R-CNN baseline settings for COCO dataset
- Use AdamW optimizer and step learning rate schedule
- Learning rate multiplier of 0.1 is applied to both CNN and Transformer backbone
- Use initial learning rate of 0.0001 and a weight decay of 0.05 for all backbones
- Decay the learning rate at 0.9 and 0.95 fraction of the total number of training steps by a factor of 10
- Data augmentation: Large-scale jittering(LSJ) augmentation with a random scale sampled from range 0.1 to 2.0 followed by a fixed size crop to 1024*1024

Panoptic segmentation Result(Evaluate on COCO)

- It outperform MaskFormer by 5.1 PQ and concurrent work K-Net by 3.2PQ
- It also achieves higher performance on average precision and mean IoU compared with DETR and MaskFormer

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 Result(Evaluate on COCO)

- With **ResNet** backbone, It outperforms a strong Mask-R-CNN also using LSJ augmentation while requiring 8* fewer training iterations
- With **Swin-L** backbone, it outperforms state-of-the-art HTC++, suggesting that the predictions have a better boundary quality thanks to high-resolution mask predictions
- However, there still remains room for improvement on small objects

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

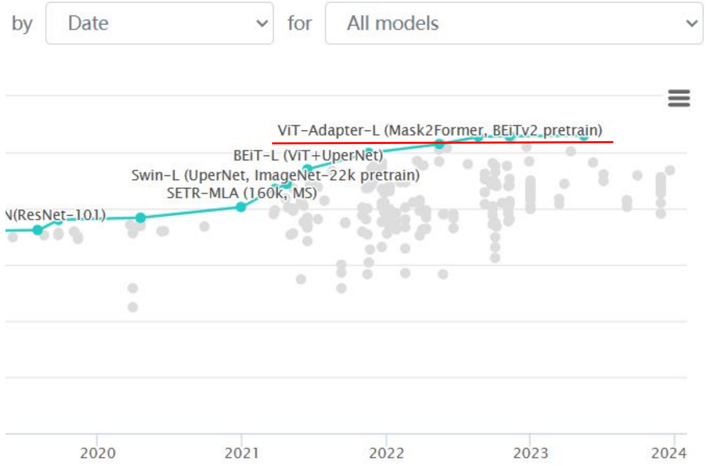
Semantic segmentation Result(Evaluate on ADE20K)

- Outperforms MaskFormer across different backbones
- Set a new state-of-the-art 57.7 mIoU

Semantic Segmentation on ADE20K

Leaderboard   Dataset

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**



Ablation Study

- Transformer decoder:
  - Masked attention leads to biggest improvement
  - Using high resolution features are also important
  - Additional optimization improvements further improve the performance without extra computation

	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

	AP	PQ	mIoU	FLOPs
Mask2Former (ours)	**43.7**	**51.9**	**47.2**	226G
— learnable query features	42.9 (-0.8)	51.2 (-0.7)	45.4 (-1.8)	226G
— cross-attention first	43.2 (-0.5)	51.6 (-0.3)	46.3 (-0.9)	226G
— remove dropout	43.0 (-0.7)	51.3 (-0.6)	47.2 (-0.0)	226G
— all 3 components above	42.3 (-1.4)	50.8 (-1.1)	46.3 (-0.9)	226G

Ablation Study (Continued)

- **Masked attention**: While existing cross-attention variants (such as mask pooling from K-Net) may improve on a specific task, masked attention performs the best on all three tasks
- **Feature resolution**: Mask2Former benefits from using high-resolution features, while the additional computation may be reduced through multi-scale strategy

- **Pixel decoder**:
  - Weighted Bi-directional Feature Pyramid Network performs better on instance-level segmentation
  - Feature-aligned Pyramid Network works better for semantic segmentation
  - MSDeformaAttn consistently performs the best across all tasks and thus is selected as default

	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

Ablation Study (Continued)

Calculating the final training loss with sampled points reduces training memory by 3* without affecting the performance

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

Limitations

- As suggested above, Mask2Former struggles with segmenting small objects and is unable to fully leverage multi-scale features
- Mask2Former still needs to be trained on different tasks, though having the same architecture

	PQ	AP	mIoU	PQ	AP	mIoU	PQ	AP	mIoU
panoptic	51.9	41.7	**61.7**	39.7	**26.5**	46.1	62.1	37.3	77.5
instance	-	**43.7**	-	-	26.4	-	-	**37.4**	-
semantic	-	-	61.5	-	-	**47.2**	-	-	**79.4**

(a) COCO (b) ADE20K (c) Cityscapes

Conclusion

Conclusion - Mask2Former

- One Model Rule Them All
- Performance surpass specialized models