arXiv:2112.01527v3 [cs.CV] 15 Jun 2022

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)

https://bowenc0221.github.io/mask2former

Abstract

Image segmentation groups pixels with different semantics, e.g.*, category or instance membership. Each choice of semantics defines a task. While only the semantics of each task differ, current research focuses on designing specialized architectures for each task. We present Masked-attention Mask Transformer (Mask2Former), a new architecture capable of addressing any image segmentation task (panoptic, instance or semantic). Its key components include* masked attention*, which extracts localized features by constraining cross-attention within predicted mask regions. In addition to reducing the research effort by at least three times, it outperforms the best specialized architectures by a significant margin on four popular datasets. Most notably, Mask2Former sets a new state-of-the-art for panoptic segmentation (57.8 PQ on COCO), instance segmentation (50.1 AP on COCO) and semantic segmentation (57.7 mIoU on ADE20K).*

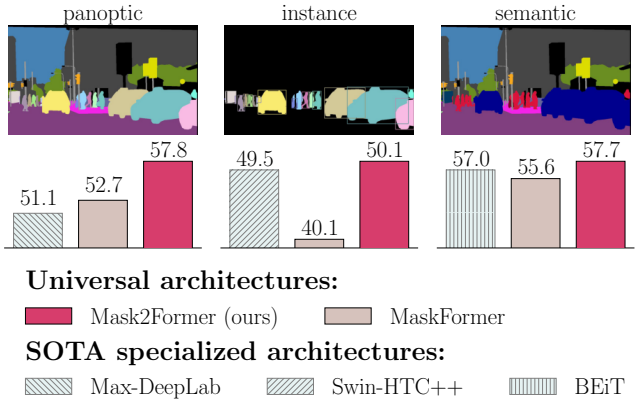


Figure 1. State-of-the-art segmentation architectures are typically specialized for each image segmentation task. Although recent work has proposed universal architectures that attempt all tasks and are competitive on semantic and panoptic segmentation, they struggle with segmenting instances. We propose **Mask2Former**, which, for the first time, outperforms the best specialized architectures on three studied segmentation tasks on multiple datasets.

1. Introduction

Image segmentation studies the problem of grouping pixels. Different semantics for grouping pixels, *e.g.*, category or instance membership, have led to different types of segmentation tasks, such as panoptic, instance or semantic segmentation. While these tasks differ only in semantics, current methods develop specialized architectures for each task. Per-pixel classification architectures based on Fully Convolutional Networks (FCNs) [37] are used for semantic segmentation, while mask classification architectures [5,24] that predict a set of binary masks each associated with a single category, dominate instance-level segmentation. Although such *specialized* architectures [6, 10, 24, 37] have advanced each individual task, they lack the flexibility to generalize to the other tasks. For example, FCN-based architectures struggle at instance segmentation, leading to the evolution of different architectures for instance segmentation compared to semantic segmentation. Thus, duplicate research and (hardware) optimization effort is spent on each specialized architecture for every task.

To address this fragmentation, recent work [14, 62] has attempted to design *universal architectures*, that are capable of addressing all segmentation tasks with the same architecture (*i.e.*, universal image segmentation). These architectures are typically based on an end-to-end set prediction objective (*e.g.*, DETR [5]), and successfully tackle multiple tasks without modifying the architecture, loss, or the training procedure. Note, universal architectures are still trained separately for different tasks and datasets, albeit having the same architecture. In addition to being flexible, universal architectures have recently shown state-of-the-art results on semantic and panoptic segmentation [14]. However, recent work still focuses on advancing specialized architectures [20, 39, 45], which raises the question: why haven't universal architectures replaced specialized ones?

Although existing universal architectures are flexible enough to tackle any segmentation task, as shown in Figure 1, in practice their performance lags behind the best specialized architectures. For instance, the best reported

*Work done during an internship at Facebook AI Research.

performance of universal architectures [14, 62], is currently lower ($>$ 9 AP) than the SOTA specialized architecture for instance segmentation [6]. Beyond the inferior performance, universal architectures are also harder to train. They typically require more advanced hardware and a much longer training schedule. For example, training MaskFormer [14] takes 300 epochs to reach 40.1 AP and it can only fit a single image in a GPU with 32G memory. In contrast, the specialized Swin-HTC++ [6] obtains better performance in only 72 epochs. Both the performance and training efficiency issues hamper the deployment of universal architectures.

In this work, we propose a universal image segmentation architecture named Masked-attention Mask Transformer (**Mask2Former**) that outperforms specialized architectures across different segmentation tasks, while still being easy to train on every task. We build upon a simple meta architecture [14] consisting of a backbone feature extractor [25, 36], a pixel decoder [33] and a Transformer decoder [51]. We propose key improvements that enable better results and efficient training. First, we use *masked attention* in the Transformer decoder which restricts the attention to localized features centered around predicted segments, which can be either objects or regions depending on the specific semantic for grouping. Compared to the cross-attention used in a standard Transformer decoder which attends to all locations in an image, our masked attention leads to faster convergence and improved performance. Second, we use *multi-scale high-resolution features* which help the model to segment small objects/regions. Third, we propose *optimization improvements* such as switching the order of self and cross-attention, making query features learnable, and removing dropout; all of which improve performance without additional compute. Finally, we save 3$\times$ training memory without affecting the performance by *calculating mask loss on few randomly sampled points*. These improvements not only boost the model performance, but also make training significantly easier, making universal architectures more accessible to users with limited compute.

We evaluate Mask2Former on three image segmentation tasks (panoptic, instance and semantic segmentation) using four popular datasets (COCO [35], Cityscapes [16], ADE20K [65] and Mapillary Vistas [42]). For the first time, on all these benchmarks, our single architecture performs on par or better than specialized architectures. Mask2Former sets the new state-of-the-art of **57.8 PQ** on COCO panoptic segmentation [28], **50.1 AP** on COCO instance segmentation [35] and **57.7 mIoU** on ADE20K semantic segmentation [65] using the exact same architecture.

2. Related Work

Specialized semantic segmentation architectures typically treat the task as a per-pixel classification problem. FCN-based architectures [37] independently predict a category label for every pixel. Follow-up methods find context to play an important role for precise per-pixel classification and focus on designing customized context modules [7,8,63] or self-attention variants [21,26,45,55,61,64].

Specialized instance segmentation architectures are typically based upon "mask classification." They predict a set of binary masks each associated with a single class label. The pioneering work, Mask R-CNN [24], generates masks from detected bounding boxes. Follow-up methods either focus on detecting more precise bounding boxes [4, 6], or finding new ways to generate a dynamic number of masks, *e.g.*, using dynamic kernels [3, 49, 56] or clustering algorithms [11, 29]. Although the performance has been advanced in each task, these specialized innovations lack the flexibility to generalize from one to the other, leading to duplicated research effort. For instance, although multiple approaches have been proposed for building feature pyramid representations [33], as we show in our experiments, BiFPN [47] performs better for instance segmentation while FaPN [39] performs better for semantic segmentation.

Panoptic segmentation has been proposed to unify both semantic and instance segmentation tasks [28]. Architectures for panoptic segmentation either combine the best of specialized semantic and instance segmentation architectures into a single framework [11, 27, 31, 60] or design novel objectives that equally treat semantic regions and instance objects [5, 52]. Despite those new architectures, researchers continue to develop specialized architectures for different image segmentation tasks [20, 45]. We find panoptic architectures usually only report performance on a single panoptic segmentation task [52], which does not guarantee good performance on other tasks (Figure 1). For example, panoptic segmentation does not measure architectures' abilities to rank predictions as instance segmentations. Thus, we refrain from referring to architectures that are only evaluated for panoptic segmentation as universal architectures. Instead, here, we evaluate our Mask2Former on all studied tasks to guarantee generalizability.

Universal architectures have emerged with DETR [5] and show that mask classification architectures with an end-to-end set prediction objective are general enough for any image segmentation task. MaskFormer [14] shows that mask classification based on DETR not only performs well on panoptic segmentation but also achieves state-of-the-art on semantic segmentation. K-Net [62] further extends set prediction to instance segmentation. Unfortunately, these architectures fail to replace specialized models as their performance on particular tasks or datasets is still worse than the best specialized architecture (*e.g.*, MaskFormer [14] cannot segment instances well). To our knowledge, Mask2Former is the first architecture that outperforms state-of-the-art specialized architectures on all considered tasks and datasets.

3. Masked-attention Mask Transformer

We now present Mask2Former. We first review a meta architecture for mask classification that Mask2Former is built upon. Then, we introduce our new Transformer decoder with *masked attention* which is the key to better convergence and results. Lastly, we propose training improvements that make Mask2Former efficient and accessible.

3.1. Mask classification preliminaries

Mask classification architectures group pixels into N segments by predicting N binary masks, along with N corresponding category labels. Mask classification is sufficiently general to address any segmentation task by assigning different semantics, *e.g.*, categories or instances, to different segments. However, the challenge is to find good representations for each segment. For example, Mask R-CNN [24] uses bounding boxes as the representation which limits its application to semantic segmentation. Inspired by DETR [5], each segment in an image can be represented as a C-dimensional feature vector ("object query") and can be processed by a Transformer decoder, trained with a set prediction objective. A simple meta architecture would consist of three components. A *backbone* that extracts low-resolution features from an image. A *pixel decoder* that gradually upsamples low-resolution features from the output of the backbone to generate high-resolution per-pixel embeddings. And finally a *Transformer decoder* that operates on image features to process object queries. The final binary mask predictions are decoded from per-pixel embeddings with object queries. One successful instantiation of such a meta architecture is MaskFormer [14], and we refer readers to [14] for more details.

3.2. Transformer decoder with masked attention

Mask2Former adopts the aforementioned meta architecture, with our proposed Transformer decoder (Figure 2 right) replacing the standard one. The key components of our Transformer decoder include a *masked attention* operator, which extracts localized features by constraining cross-attention to within the foreground region of the predicted mask for each query, instead of attending to the full feature map. To handle small objects, we propose an efficient multi-scale strategy to utilize high-resolution features. It feeds successive feature maps from the pixel decoder's feature pyramid into successive Transformer decoder layers in a round robin fashion. Finally, we incorporate optimization improvements that boost model performance without introducing additional computation. We now discuss these improvements in detail.

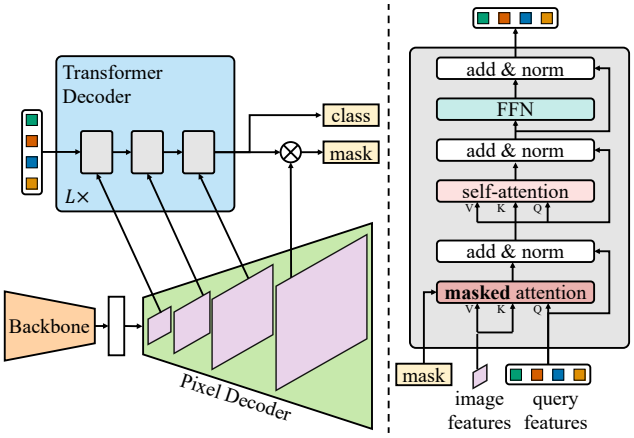


Figure 2. **Mask2Former overview.** Mask2Former adopts the same meta architecture as MaskFormer [14] with a backbone, a pixel decoder and a Transformer decoder. We propose a new Transformer decoder with *masked attention* instead of the standard cross-attention (Section 3.2.1). To deal with small objects, we propose an efficient way of utilizing high-resolution features from a pixel decoder by feeding one scale of the multi-scale feature to one Transformer decoder layer at a time (Section 3.2.2). In addition, we switch the order of self and cross-attention (*i.e.*, our masked attention), make query features learnable, and remove dropout to make computation more effective (Section 3.2.3). Note that positional embeddings and predictions from intermediate Transformer decoder layers are omitted in this figure for readability.

3.2.1 Masked attention

Context features have been shown to be important for image segmentation [7, 8, 63]. However, recent studies [22, 46] suggest that the slow convergence of Transformer-based models is due to global context in the cross-attention layer, as it takes many training epochs for cross-attention to learn to attend to localized object regions [46]. We hypothesize that local features are enough to update query features and context information can be gathered through self-attention. For this we propose *masked attention*, a variant of cross-attention that only attends within the foreground region of the predicted mask for each query.

Standard cross-attention (with residual path) computes

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

Here, l is the layer index, $\mathbf{X}_l \in \mathbb{R}^{N \times C}$ refers to N C-dimensional query features at the l^{th} layer and $\mathbf{Q}_l = f_Q(\mathbf{X}_{l-1}) \in \mathbb{R}^{N \times C}$. $\mathbf{X}_0$ denotes input query features to the Transformer decoder. $\mathbf{K}_l, \mathbf{V}_l \in \mathbb{R}^{H_l W_l \times C}$ are the image features under transformation $f_K(\cdot)$ and $f_V(\cdot)$ respectively, and H_l and W_l are the spatial resolution of image features that we will introduce next in Section 3.2.2. f_Q, f_K and f_V are linear transformations.

Our masked attention modulates the attention matrix via

$$\mathbf{X}_l = \text{softmax}(\mathcal{M}_{l-1} + \mathbf{Q}_l\mathbf{K}_l^{\text{T}})\mathbf{V}_l + \mathbf{X}_{l-1}. \quad (2)$$

Moreover, the attention mask $\mathcal{M}_{l-1}$ at feature location (x, y) is

$$\mathcal{M}_{l-1}(x,y) = \begin{cases} 0 & \text{if } \mathbf{M}_{l-1}(x,y) = 1 \\ -\infty & \text{otherwise} \end{cases}. \quad (3)$$

Here, $\mathbf{M}_{l-1} \in \{0,1\}^{N \times H_l W_l}$ is the binarized output (thresholded at 0.5) of the resized mask prediction of the previous $(l-1)$-th Transformer decoder layer. It is resized to the same resolution of $\mathbf{K}_l$. $\mathbf{M}_0$ is the binary mask prediction obtained from $\mathbf{X}_0$, *i.e.*, before feeding query features into the Transformer decoder.

3.2.2 High-resolution features

High-resolution features improve model performance, especially for small objects [5]. However, this is computationally demanding. Thus, we propose an efficient multi-scale strategy to introduce high-resolution features while controlling the increase in computation. Instead of always using the high-resolution feature map, we utilize a feature pyramid which consists of both low- and high-resolution features and feed one resolution of the multi-scale feature to one Transformer decoder layer at a time.

Specifically, we use the feature pyramid produced by the *pixel decoder* with resolution $1/32$, $1/16$ and $1/8$ of the original image. For each resolution, we add both a sinusoidal positional embedding $e_{\text{pos}} \in \mathbb{R}^{H_l W_l \times C}$, following [5], and a learnable scale-level embedding $e_{\text{lvl}} \in \mathbb{R}^{1 \times C}$, following [66]. We use those, from lowest-resolution to highest-resolution for the corresponding Transformer decoder layer as shown in Figure 2 left. We repeat this 3-layer Transformer decoder L times. Our final Transformer decoder hence has $3L$ layers. More specifically, the first three layers receive a feature map of resolution $H_1 = H/32$, $H_2 = H/16$, $H_3 = H/8$ and $W_1 = W/32$, $W_2 = W/16$, $W_3 = W/8$, where H and W are the original image resolution. This pattern is repeated in a round robin fashion for all following layers.

3.2.3 Optimization improvements

A standard Transformer decoder layer [51] consists of three modules to process query features in the following order: a self-attention module, a cross-attention and a feed-forward network (FFN). Moreover, query features ($\mathbf{X}_0$) are *zero initialized* before being fed into the Transformer decoder and are associated with *learnable* positional embeddings. Furthermore, dropout is applied to both residual connections and attention maps.

To optimize the Transformer decoder design, we make the following three improvements. First, we switch the order of self- and cross-attention (our new "masked attention") to make computation more effective: query features to the first self-attention layer are image-independent and do not have signals from the image, thus applying self-attention is unlikely to enrich information. Second, we make query features ($\mathbf{X}_0$) learnable as well (we still keep the learnable query positional embeddings), and learnable query features are directly supervised before being used in the Transformer decoder to predict masks ($\mathbf{M}_0$). We find these learnable query features function like a region proposal network [43] and have the ability to generate mask proposals. Finally, we find dropout is not necessary and usually decreases performance. We thus completely remove dropout in our decoder.

3.3. Improving training efficiency

One limitation of training universal architectures is the large memory consumption due to high-resolution mask prediction, making them less accessible than the more memory-friendly specialized architectures [6, 24]. For example, MaskFormer [14] can only fit a single image in a GPU with 32G memory. Motivated by PointRend [30] and Implicit PointRend [13], which show a segmentation model can be trained with its mask loss calculated on K randomly sampled points instead of the whole mask, we calculate the mask loss with sampled points in both the matching and the final loss calculation. More specifically, in the *matching loss* that constructs the cost matrix for bipartite matching, we *uniformly* sample the same set of K points for all prediction and ground truth masks. In the *final loss* between predictions and their matched ground truths, we sample different sets of K points for different pairs of prediction and ground truth using *importance sampling* [30]. We set $K = 12544$, *i.e.*, 112×112 points. This new training strategy effectively reduces training memory by $3\times$, from 18GB to 6GB per image, making Mask2Former more accessible to users with limited computational resources.

4. Experiments

We demonstrate Mask2Former is an effective architecture for universal image segmentation through comparisons with specialized state-of-the-art architectures on standard benchmarks. We evaluate our proposed design decisions through ablations on all three tasks. Finally we show Mask2Former generalizes beyond the standard benchmarks, obtaining state-of-the-art results on four datasets.

Datasets. We study Mask2Former using four widely used image segmentation datasets that support semantic, instance and panoptic segmentation: COCO [35] (80 "things" and 53 "stuff" categories), ADE20K [65] (100 "things" and 50 "stuff" categories), Cityscapes [16] (8 "things" and 11 "stuff" categories) and Mapillary Vistas [42] (37 "things" and 28 "stuff" categories). Panoptic and semantic seg-

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

Table 1. **Panoptic segmentation on COCO panoptic `val2017` with 133 categories.** Mask2Former consistently outperforms MaskFormer [14] by a large margin with different backbones on all metrics. Our best model outperforms prior state-of-the-art MaskFormer by 5.1 PQ and K-Net [62] by 3.2 PQ. Backbones pre-trained on ImageNet-22K are marked with †.

mentation tasks are evaluated on the union of "things" and "stuff" categories while instance segmentation is only evaluated on the "things" categories.

Evaluation metrics. For *panoptic segmentation*, we use the standard **PQ** (panoptic quality) metric [28]. We further report $\mathbf{AP^{Th}_{pan}}$, which is the AP evaluated on the "thing" categories using instance segmentation annotations, and $\mathbf{mIoU_{pan}}$, which is the mIoU for semantic segmentation by merging instance masks from the same category, of the same model trained *only* with panoptic segmentation annotations. For *instance segmentation*, we use the standard **AP** (average precision) metric [35]. For *semantic segmentation*, we use **mIoU** (mean Intersection-over-Union) [19].

4.1. Implementation details

We adopt settings from [14] with the following differences:

Pixel decoder. Mask2Former is compatible with any existing pixel decoder module. In MaskFormer [14], FPN [33] is chosen as the default for its simplicity. Since our goal is to demonstrate strong performance across different segmentation tasks, we use the more advanced multi-scale deformable attention Transformer (MSDeformAttn) [66] as our default pixel decoder. Specifically, we use 6 MSDeformAttn layers applied to feature maps with resolution $1/8$, $1/16$ and $1/32$, and use a simple upsampling layer with lateral connection on the final $1/8$ feature map to generate the feature map of resolution $1/4$ as the per-pixel embedding. In our ablation study, we show that this pixel decoder provides best results across different segmentation tasks.

Transformer decoder. We use our Transformer decoder proposed in Section 3.2 with $L = 3$ (*i.e.*, 9 layers total) and 100 queries by default. An auxiliary loss is added to every intermediate Transformer decoder layer and to the learnable query features before the Transformer decoder.

Loss weights. We use the binary cross-entropy loss (instead of focal loss [34] in [14]) and the dice loss [41] for our mask loss: $\mathcal{L}_{\text{mask}} = \lambda_{\text{ce}}\mathcal{L}_{\text{ce}} + \lambda_{\text{dice}}\mathcal{L}_{\text{dice}}$. We set $\lambda_{\text{ce}} = 5.0$ and $\lambda_{\text{dice}} = 5.0$. The final loss is a combination of mask loss and classification loss: $\mathcal{L}_{\text{mask}} + \lambda_{\text{cls}}\mathcal{L}_{\text{cls}}$ and we set $\lambda_{\text{cls}} = 2.0$ for predictions matched with a ground truth and 0.1 for the "no object," *i.e.*, predictions that have not been matched with any ground truth.

Post-processing. We use the exact same post-processing as [14] to acquire the expected output format for panoptic and semantic segmentation from pairs of binary masks and class predictions. Instance segmentation requires additional confidence scores for each prediction. We multiply class confidence and mask confidence (*i.e.*, averaged foreground per-pixel binary mask probability) for a final confidence.

4.2. Training settings

Panoptic and instance segmentation. We use Detectron2 [57] and follow the updated Mask R-CNN [24] baseline settings[1] for the COCO dataset. More specifically, we use AdamW [38] optimizer and the step learning rate schedule. We use an initial learning rate of 0.0001 and a weight decay of 0.05 for all backbones. A learning rate multiplier of 0.1 is applied to the backbone and we decay the learning rate at 0.9 and 0.95 fractions of the total number of training steps by a factor of 10. If not stated otherwise, we train our models for 50 epochs with a batch size of 16. For data augmentation, we use the large-scale jittering (LSJ) augmentation [18, 23] with a random scale sampled from range 0.1 to 2.0 followed by a fixed size crop to 1024×1024. We use the standard Mask R-CNN inference setting where we resize an image with shorter side to 800 and longer side up-to 1333. We also report FLOPs and fps. FLOPs are averaged over 100 validation images (COCO images have varying sizes). Frames-per-second (fps) is measured on a V100 GPU with a batch size of 1 by taking the average runtime on the entire validation set including post-processing time.

Semantic segmentation. We follow the same settings as [14] to train our models, except: 1) a learning rate multiplier of 0.1 is applied to *both* CNN and Transformer backbones instead of only applying it to CNN backbones in [14], 2) both ResNet and Swin backbones use an initial learning rate of 0.0001 and a weight decay of 0.05, instead of using

[1] https://github.com/facebookresearch/detectron2/blob/main/MODEL_ZOO.md#new-baselines-using-large-scale-jitter-and-longer-training-schedule

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

Table 2. **Instance segmentation on COCO `val2017` with 80 categories.** Mask2Former outperforms strong Mask R-CNN [24] baselines for both AP and $AP^{boundary}$ [12] metrics when training with 8× fewer epochs. Our best model is also competitive to the state-of-the-art specialized instance segmentation model on COCO and has higher boundary quality. For a fair comparison, we only consider single-scale inference and models trained using only COCO `train2017` set data. Backbones pre-trained on ImageNet-22K are marked with †.

different learning rates in [14].

4.3. Main results

Panoptic segmentation. We compare Mask2Former with state-of-the-art models for panoptic segmentation on the COCO panoptic [28] dataset in Table 1. Mask2Former consistently outperforms MaskFormer by more than 5 PQ across different backbones while converging 6× faster. With Swin-L backbone, our Mask2Former sets a new state-of-the-art of 57.8 PQ, outperforming existing state-of-the-art [14] by 5.1 PQ and concurrent work, K-Net [62], by 3.2 PQ. Mask2Former even outperforms the best ensemble models with extra training data in the COCO challenge (see Appendix A.1 for test set results).

Beyond the PQ metric, our Mask2Former also achieves higher performance on two other metrics compared to DETR [5] and MaskFormer: AP^{Th}_{pan}, which is the AP evaluated on the 80 "thing" categories using *instance segmentation annotation*, and $mIoU_{pan}$, which is the mIoU evaluated on the 133 categories for semantic segmentation converted from panoptic segmentation annotation. This shows Mask2Former's universality: trained *only* with panoptic segmentation annotations, it can be used for instance and semantic segmentation.

Instance segmentation. We compare Mask2Former with state-of-the-art models on the COCO [35] dataset in Table 2. With ResNet [25] backbone, Mask2Former outperforms a strong Mask R-CNN [24] baseline using large-scale jittering (LSJ) augmentation [18, 23] while requiring 8× fewer training iterations. With Swin-L backbone, Mask2Former outperforms the state-of-the-art HTC++ [6]. Although we only observe +0.6 AP improvement over HTC++, the Boundary AP [12] improves by 2.1, suggesting that our predictions have a better boundary quality thanks to the high-resolution mask predictions. Note that for a fair comparison, we only consider single-scale inference and models trained with only COCO `train2017` set data.

With a ResNet-50 backbone Mask2Former improves over MaskFormer on small objects by 7.0 AP^S, while overall the highest gains come from large objects (+10.6 AP^L). The performance on AP^S still lags behind other state-of-the-art models. Hence there still remains room for improvement on small objects, *e.g.*, by using dilated backbones like in DETR [5], which we leave for future work.

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 †.

Semantic segmentation. We compare Mask2Former with state-of-the-art models for semantic segmentation on the ADE20K [65] dataset in Table 3. Mask2Former outperforms MaskFormer [14] across different backbones, suggesting that the proposed improvements even boost semantic segmentation results where [14] was already state-of-the-art. With Swin-L as backbone and FaPN [39] as pixel decoder, Mask2Former sets a new state-of-the-art of 57.7 mIoU. We also report the test set results in Appendix A.3.

4.4. Ablation studies

We now analyze Mask2Former through a series of ablation studies using a ResNet-50 backbone [25]. To test the generality of the proposed components for universal image segmentation, *all ablations are performed on three tasks*.

	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
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

(b) Optimization improvements increase the performance without introducing extra compute. Following DETR [5], query features are zero-initialized when not learnable. 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	46.6	213G
mask pooling [62]	43.1	51.5	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**	51.8	**47.4**	239G
naïve m.s. (3 scales)	**44.0**	**51.9**	46.3	247G
efficient m.s. (3 scales)	43.7	**51.9**	47.2	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.

	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.

Table 4. **Mask2Former ablations.** We perform ablations on three tasks: instance (AP on COCO `val2017`), panoptic (PQ on COCO panoptic `val2017`) and semantic (mIoU on ADE20K `val`) segmentation. FLOPs are measured on COCO instance segmentation.

Transformer decoder. We validate the importance of each component by removing them one at a time. As shown in Table 4a, masked attention leads to the biggest improvement across all tasks. The improvement is larger for instance and panoptic segmentation than for semantic segmentation. Moreover, using high-resolution features from the efficient multi-scale strategy is also important. Table 4b shows additional optimization improvements further improve the performance without extra computation.

Masked attention. Concurrent work has proposed other variants of cross-attention [22, 40] that aim to improve the convergence and performance of DETR [5] for object detection. Most recently, K-Net [62] replaced cross-attention with a mask pooling operation that averages features within mask regions. We validate the importance of our masked attention in Table 4c. While existing cross-attention variants may improve on a specific task, our masked attention performs the best on all three tasks.

Feature resolution. Table 4d shows that Mask2Former benefits from using high-resolution features (*e.g.*, a single scale of 1/8) in the Transformer decoder. However, this introduces additional computation. Our efficient multi-scale (efficient m.s.) strategy effectively reduces the FLOPs without affecting the performance. Note that, naively concatenating multi-scale features as input to every Transformer decoder layer (naïve m.s.) does not yield additional gains.

Pixel decoder. As shown in Table 4e, Mask2Former is compatible with any existing pixel decoder. However, we observe different pixel decoders specialize in different tasks: while BiFPN [47] performs better on instance-level segmentation, FaPN [39] works better for semantic segmentation. Among all studied pixel decoders, the MSDeformaAttn [66] consistently performs the best across all tasks and thus is selected as our default. This set of ablations also suggests that designing a module like a pixel decoder for a specific task does not guarantee generalization across segmentation tasks. Mask2Former, as a universal model, could serve as a testbed for a generalizable module design.

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.

Calculating loss with points *vs*. masks. In Table 5 we study the performance and memory implications when calculating the loss based on either mask or sampled points. Calculating the final training loss with sampled points reduces training memory by 3× without affecting the performance. Additionally, calculating the matching loss with sampled points improves performance across all three tasks.

Learnable queries as region proposals. Region proposals [1, 50], either in the form of boxes or masks, are regions that are likely to be "objects." With learnable queries being supervised by the mask loss, predictions from learnable queries can serve as mask proposals. In Figure 3 top, we visualize mask predictions of selected learnable queries *before* feeding them into the Transformer decoder (the proposal generation process is shown in Figure 3 bottom right). In Figure 3 bottom left, we further perform a quantitative analysis on the quality of these proposals by calculating the class-agnostic average recall with 100 predictions (AR@100) on COCO `val2017`. We find these learnable queries already achieve good AR@100 compared to the fi-

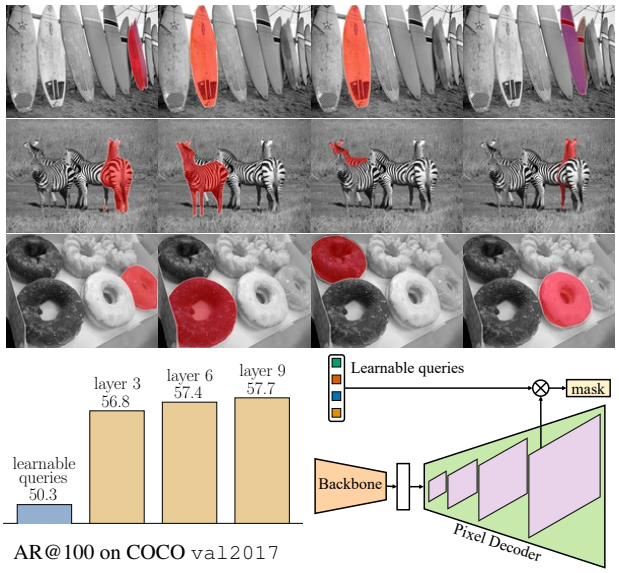


Figure 3. **Learnable queries as "region proposals".** *Top:* We visualize mask predictions of four selected learnable queries *before* feeding them into the Transformer decoder (using R50 backbone). *Bottom left:* We calculate the class-agnostic average recall with 100 proposals (AR@100) and observe that these learnable queries provide good proposals compared to the final predictions of Mask2Former after the Transformer decoder layers (layer 9). *Bottom right:* Illustration of proposal generation process.

method	backbone	panoptic model PQ	AP^{Th}_{pan}	$mIoU_{pan}$	semantic model mIoU (s.s.)	(m.s.)
Panoptic FCN [31]	Swin-L†	65.9	-	-	-	-
Panoptic-DeepLab [11]	SWideRNet [9]	66.4	40.1	82.2	-	-
Panoptic-DeepLab [11]	SWideRNet [9]	67.5*	43.9*	82.9*	-	-
SETR [64]	ViT-L† [17]	-	-	-	-	82.2
SegFormer [59]	MiT-B5 [59]	-	-	-	-	84.0
Mask2Former (ours)	R50	62.1	37.3	77.5	79.4	82.2
	Swin-B†	66.1	42.8	82.7	**83.3**	**84.5**
	Swin-L†	**66.6**	**43.6**	**82.9**	**83.3**	84.3

Table 6. **Cityscapes `val`.** Mask2Former is competitive to specialized models on Cityscapes. Panoptic segmentation models use single-scale inference by default, multi-scale numbers are marked with *. For semantic segmentation, we report both single-scale (s.s.) and multi-scale (m.s.) inference results. Backbones pre-trained on ImageNet-22K are marked with †.

nal predictions of Mask2Former after the Transformer decoder layers, *i.e.*, layer 9, and AR@100 consistently improves with more decoder layers.

4.5. Generalization to other datasets

To show our Mask2Former can generalize beyond the COCO dataset, we further perform experiments on other popular image segmentation datasets. In Table 6, we show results on Cityscapes [16]. Please see Appendix B for detailed training settings on each dataset as well as more results on ADE20K [65] and Mapillary Vistas [42].

	(a) COCO PQ	AP	mIoU	(b) ADE20K PQ	AP	mIoU	(c) Cityscapes 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**

Table 7. **Limitations of Mask2Former.** Although a single Mask2Former can address any segmentation task, we still need to train it on different tasks. Across three datasets we find Mask2Former trained with panoptic annotations performs slightly worse than the exact same model trained specifically for instance and semantic segmentation tasks with the corresponding data.

We observe that our Mask2Former is competitive to state-of-the-art methods on these datasets as well. It suggests Mask2Former can serve as a universal image segmentation model and results generalize across datasets.

4.6. Limitations

Our ultimate goal is to train a *single* model for *all* image segmentation tasks. In Table 7, we find Mask2Former trained on panoptic segmentation only performs slightly worse than the exact same model trained with the corresponding annotations for instance and semantic segmentation tasks across three datasets. This suggests that even though Mask2Former can generalize to different tasks, it still needs to be trained for those specific tasks. In the future, we hope to develop a model that can be trained only once for multiple tasks and even for multiple datasets.

Furthermore, as seen in Tables 2 and 4d, even though it improves over baselines, Mask2Former struggles with segmenting small objects and is unable to fully leverage multi-scale features. We believe better utilization of the feature pyramid and designing losses for small objects are critical.

5. Conclusion

We present Mask2Former for universal image segmentation. Built upon a simple meta framework [14] with a new Transformer decoder using the proposed masked attention, Mask2Former obtains top results in all three major image segmentation tasks (panoptic, instance and semantic) on four popular datasets, outperforming even the best specialized models designed for each benchmark while remaining easy to train. Mask2Former saves 3× research effort compared to designing specialized models for each task, and it is accessible to users with limited computational resources. We hope to attract interest in universal model design.

Ethical considerations: While our technical innovations do not appear to have any inherent biases, the models trained with our approach on real-world datasets should undergo ethical review to ensure the predictions do not propagate problematic stereotypes, and the approach is not used for applications including but not limited to illegal surveillance.

Acknowledgments: Thanks to Nicolas Carion and Xingyi Zhou for helpful feedback. BC and AS are supported in part by NSF #1718221, 2008387, 2045586, 2106825, MRI #1725729, NIFA 2020-67021-32799 and Cisco Systems Inc. (CG 1377144 - thanks for access to Arcetri).

References

[1] Pablo Arbeláez, Jordi Pont-Tuset, Jonathan T Barron, Ferran Marques, and Jitendra Malik. Multiscale combinatorial grouping. In *CVPR*, 2014.

[2] Hangbo Bao, Li Dong, and Furu Wei. BEiT: BERT pre-training of image transformers. *arXiv*, 2021.

[3] Daniel Bolya, Chong Zhou, Fanyi Xiao, and Yong Jae Lee. YOLACT++: Better real-time instance segmentation, 2019.

[4] Zhaowei Cai and Nuno Vasconcelos. Cascade R-CNN: Delving into high quality object detection. In *CVPR*, 2018.

[5] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *ECCV*, 2020.

[6] Kai Chen, Jiangmiao Pang, Jiaqi Wang, Yu Xiong, Xiaoxiao Li, Shuyang Sun, Wansen Feng, Ziwei Liu, Jianping Shi, Wanli Ouyang, et al. Hybrid task cascade for instance segmentation. In *CVPR*, 2019.

[7] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L Yuille. DeepLab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected CRFs. *PAMI*, 2018.

[8] Liang-Chieh Chen, George Papandreou, Florian Schroff, and Hartwig Adam. Rethinking atrous convolution for semantic image segmentation. *arXiv:1706.05587*, 2017.

[9] Liang-Chieh Chen, Huiyu Wang, and Siyuan Qiao. Scaling wide residual networks for panoptic segmentation. *arXiv:2011.11675*, 2020.

[10] Liang-Chieh Chen, Yukun Zhu, George Papandreou, Florian Schroff, and Hartwig Adam. Encoder-decoder with atrous separable convolution for semantic image segmentation. In *ECCV*, 2018.

[11] Bowen Cheng, Maxwell D Collins, Yukun Zhu, Ting Liu, Thomas S Huang, Hartwig Adam, and Liang-Chieh Chen. Panoptic-DeepLab: A simple, strong, and fast baseline for bottom-up panoptic segmentation. In *CVPR*, 2020.

[12] Bowen Cheng, Ross Girshick, Piotr Dollár, Alexander C Berg, and Alexander Kirillov. Boundary iou: Improving object-centric image segmentation evaluation. In *CVPR*, 2021.

[13] Bowen Cheng, Omkar Parkhi, and Alexander Kirillov. Pointly-supervised instance segmentation. *arXiv*, 2021.

[14] Bowen Cheng, Alexander G. Schwing, and Alexander Kirillov. Per-pixel classification is not all you need for semantic segmentation. In *NeurIPS*, 2021.

[15] François Chollet. Xception: Deep learning with depthwise separable convolutions. In *CVPR*, 2017.

[16] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The Cityscapes dataset for semantic urban scene understanding. In *CVPR*, 2016.

[17] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. In *ICLR*, 2021.

[18] Xianzhi Du, Barret Zoph, Wei-Chih Hung, and Tsung-Yi Lin. Simple training strategies and model scaling for object detection. *arXiv preprint arXiv:2107.00057*, 2021.

[19] Mark Everingham, SM Ali Eslami, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. The PASCAL visual object classes challenge: A retrospective. *IJCV*, 2015.

[20] Yuxin Fang, Shusheng Yang, Xinggang Wang, Yu Li, Chen Fang, Ying Shan, Bin Feng, and Wenyu Liu. Instances as queries. In *ICCV*, 2021.

[21] Jun Fu, Jing Liu, Haijie Tian, Yong Li, Yongjun Bao, Zhiwei Fang, and Hanqing Lu. Dual attention network for scene segmentation. In *CVPR*, 2019.

[22] Peng Gao, Minghang Zheng, Xiaogang Wang, Jifeng Dai, and Hongsheng Li. Fast convergence of detr with spatially modulated co-attention. In *ICCV*, 2021.

[23] Golnaz Ghiasi, Yin Cui, Aravind Srinivas, Rui Qian, Tsung-Yi Lin, Ekin D Cubuk, Quoc V Le, and Barret Zoph. Simple copy-paste is a strong data augmentation method for instance segmentation. In *CVPR*, 2021.

[24] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask R-CNN. In *ICCV*, 2017.

[25] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016.

[26] Zilong Huang, Xinggang Wang, Lichao Huang, Chang Huang, Yunchao Wei, and Wenyu Liu. CCNet: Criss-cross attention for semantic segmentation. In *ICCV*, 2019.

[27] Alexander Kirillov, Ross Girshick, Kaiming He, and Piotr Dollár. Panoptic feature pyramid networks. In *CVPR*, 2019.

[28] Alexander Kirillov, Kaiming He, Ross Girshick, Carsten Rother, and Piotr Dollár. Panoptic segmentation. In *CVPR*, 2019.

[29] Alexander Kirillov, Evgeny Levinkov, Bjoern Andres, Bogdan Savchynskyy, and Carsten Rother. InstanceCut: from edges to instances with multicut. In *CVPR*, 2017.

[30] Alexander Kirillov, Yuxin Wu, Kaiming He, and Ross Girshick. PointRend: Image segmentation as rendering. In *CVPR*, 2020.

[31] Yanwei Li, Hengshuang Zhao, Xiaojuan Qi, Yukang Chen, Lu Qi, Liwei Wang, Zeming Li, Jian Sun, and Jiaya Jia. Fully convolutional networks for panoptic segmentation with point-based supervision. *arXiv preprint arXiv:2108.07682*, 2021.

[32] Zhiqi Li, Wenhai Wang, Enze Xie, Zhiding Yu, Anima Anandkumar, Jose M Alvarez, Tong Lu, and Ping Luo. Panoptic segformer. *arXiv preprint arXiv:2109.03814*, 2021.

[33] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In *CVPR*, 2017.

[34] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *ICCV*, 2017.

[35] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft COCO: Common objects in context. In *ECCV*, 2014.

[36] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. *arXiv:2103.14030*, 2021.

[37] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *CVPR*, 2015.

[38] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *ICLR*, 2019.

[39] Shihua Huang Zhichao Lu, Ran Cheng, and Cheng He. Fapn: Feature-aligned pyramid network for dense image prediction. *arXiv*, 2021.

[40] Depu Meng, Xiaokang Chen, Zejia Fan, Gang Zeng, Houqiang Li, Yuhui Yuan, Lei Sun, and Jingdong Wang. Conditional detr for fast training convergence. In *ICCV*, 2021.

[41] Fausto Milletari, Nassir Navab, and Seyed-Ahmad Ahmadi. V-Net: Fully convolutional neural networks for volumetric medical image segmentation. In *3DV*, 2016.

[42] Gerhard Neuhold, Tobias Ollmann, Samuel Rota Bulò, and Peter Kontschieder. The mapillary vistas dataset for semantic understanding of street scenes. In *CVPR*, 2017.

[43] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster R-CNN: Towards real-time object detection with region proposal networks. In *NeurIPS*, 2015.

[44] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *IJCV*, 2015.

[45] Robin Strudel, Ricardo Garcia, Ivan Laptev, and Cordelia Schmid. Segmenter: Transformer for semantic segmentation. In *ICCV*, 2021.

[46] Zhiqing Sun, Shengcao Cao, Yiming Yang, and Kris M Kitani. Rethinking transformer-based set prediction for object detection. In *ICCV*, 2021.

[47] Mingxing Tan, Ruoming Pang, and Quoc V Le. Efficientdet: Scalable and efficient object detection. In *CVPR*, 2020.

[48] Andrew Tao, Karan Sapra, and Bryan Catanzaro. Hierarchical multi-scale attention for semantic segmentation. *arXiv:2005.10821*, 2020.

[49] Zhi Tian, Chunhua Shen, and Hao Chen. Conditional convolutions for instance segmentation. In *ECCV*, 2020.

[50] Jasper RR Uijlings, Koen EA Van De Sande, Theo Gevers, and Arnold WM Smeulders. Selective search for object recognition. *IJCV*, 2013.

[51] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *NeurIPS*, 2017.

[52] Huiyu Wang, Yukun Zhu, Hartwig Adam, Alan Yuille, and Liang-Chieh Chen. MaX-DeepLab: End-to-end panoptic segmentation with mask transformers. In *CVPR*, 2021.

[53] Jingdong Wang, Ke Sun, Tianheng Cheng, Borui Jiang, Chaorui Deng, Yang Zhao, Dong Liu, Yadong Mu, Mingkui Tan, Xinggang Wang, Wenyu Liu, and Bin Xiao. Deep high-resolution representation learning for visual recognition. *PAMI*, 2019.

[54] Wenhai Wang, Enze Xie, Xiang Li, Deng-Ping Fan, Kaitao Song, Ding Liang, Tong Lu, Ping Luo, and Ling Shao. Pvtv2: Improved baselines with pyramid vision transformer. *arXiv preprint arXiv:2106.13797*, 2021.

[55] Xiaolong Wang, Ross Girshick, Abhinav Gupta, and Kaiming He. Non-local neural networks. In *CVPR*, 2018.

[56] Xinlong Wang, Rufeng Zhang, Tao Kong, Lei Li, and Chunhua Shen. SOLOv2: Dynamic and fast instance segmentation. *NeurIPS*, 2020.

[57] Yuxin Wu, Alexander Kirillov, Francisco Massa, Wan-Yen Lo, and Ross Girshick. Detectron2. https://github.com/facebookresearch/detectron2, 2019.

[58] Tete Xiao, Yingcheng Liu, Bolei Zhou, Yuning Jiang, and Jian Sun. Unified perceptual parsing for scene understanding. In *ECCV*, 2018.

[59] Enze Xie, Wenhai Wang, Zhiding Yu, Anima Anandkumar, Jose M Alvarez, and Ping Luo. Segformer: Simple and efficient design for semantic segmentation with transformers. In *NeurIPS*, 2021.

[60] Yuwen Xiong, Renjie Liao, Hengshuang Zhao, Rui Hu, Min Bai, Ersin Yumer, and Raquel Urtasun. Upsnet: A unified panoptic segmentation network. In *CVPR*, 2019.

[61] Yuhui Yuan, Lang Huang, Jianyuan Guo, Chao Zhang, Xilin Chen, and Jingdong Wang. OCNet: Object context for semantic segmentation. *IJCV*, 2021.

[62] Wenwei Zhang, Jiangmiao Pang, Kai Chen, and Chen Change Loy. K-net: Towards unified image segmentation. In *NeurIPS*, 2021.

[63] Hengshuang Zhao, Jianping Shi, Xiaojuan Qi, Xiaogang Wang, and Jiaya Jia. Pyramid scene parsing network. In *CVPR*, 2017.

[64] Sixiao Zheng, Jiachen Lu, Hengshuang Zhao, Xiatian Zhu, Zekun Luo, Yabiao Wang, Yanwei Fu, Jianfeng Feng, Tao Xiang, Philip HS Torr, et al. Rethinking semantic segmentation from a sequence-to-sequence perspective with transformers. In *CVPR*, 2021.

[65] Bolei Zhou, Hang Zhao, Xavier Puig, Sanja Fidler, Adela Barriuso, and Antonio Torralba. Scene parsing through ADE20K dataset. In *CVPR*, 2017.

[66] Xizhou Zhu, Weijie Su, Lewei Lu, Bin Li, Xiaogang Wang, and Jifeng Dai. Deformable detr: Deformable transformers for end-to-end object detection. In *ICLR*, 2021.

Appendix

We first provide more results for Mask2Former with different backbones as well as test-set performance on standard benchmarks (Appendix A): We use COCO panoptic [28] for panoptic, COCO [35] for instance, and ADE20K [65] for semantic segmentation. Then, we provide more detailed results on additional datasets (Appendix B). Finally, we provide additional ablation studies (Appendix C) and visualization of Mask2Former predictions for all three segmentation tasks (Appendix D).

A. Additional results

Here, we provide more results of Mask2Former with different backbones on COCO panoptic [28] for panoptic segmentation, COCO [35] for instance segmentation and ADE20K [65] for semantic segmentation. More specifically, for each benckmark, we evaluate Mask2Former with ResNet [25] with 50 and 101 layers, as well as Swin [36] Tiny, Small, Base and Large variants as backbones. We use ImageNet [44] pre-trained checkpoints to initialize backbones.

A.1. Panoptic segmentation.

In Table I, we report Mask2Former with various backbones on COCO panoptic `val2017`. Mask2Former outperforms *all* existing panoptic segmentation models with various backbones. Our best model sets a new state-of-the-art of 57.8 PQ.

In Table II, we further report the best Mask2Former model on the `test-dev` set. Note that Mask2Former **trained only with the standard `train2017` data**, achieves the *absolute* new state-of-the-art performance on both validation and test set. Mask2Former even outperforms the best COCO competition entry which uses extra training data and test-time augmentation.

A.2. Instance segmentation.

In Table III, we report Mask2Former results obtained with various backbones on COCO `val2017`. Mask2Former outperforms the best single-scale model, HTC++ [6, 36]. Note that it is non-trivial to do multi-scale inference for instance-level segmentation tasks without introducing complex post-processing like non-maximum suppression. Thus, we only compare Mask2Former with other single-scale inference models. We believe multi-scale inference can further improve Mask2Former performance and it remains an interesting future work.

In Table IV, we further report the best Mask2Former model on the `test-dev` set. Mask2Former achieves the *absolute* new state-of-the-art performance on both validation and test set. On the one hand, Mask2Former is extremely good at segmenting large objects: we can even outperform the challenge winner (which uses extra training data, model ensemble, *etc*.) on AP^L by a large margin without any bells-and-whistles. On the other hand, the poor performance on small objects leaves room for further improvement in the future.

A.3. Semantic segmentation.

In Table V, we report Mask2Former results obtained with various backbones on ADE20K `val`. Mask2Former outperforms *all* existing semantic segmentation models with various backbones. Our best model sets a new state-of-the-art of 57.7 mIoU.

In Table VI, we further report the best Mask2Former model on the `test` set. Following [14], we train Mask2Former on the union of ADE20K `train` and `val` set with ImageNet-22K pre-trained checkpoint and use multi-scale inference. Mask2Former is able to outperform previous state-of-the-art methods on all metrics.

B. Additional datasets

We study Mask2Former on three image segmentation tasks (panoptic, instance and semantic segmentation) using four datasets. Here we report additional results on Cityscapes [16], ADE20K [65] and Mapillary Vistas [42] as well as more detailed training settings.

B.1. Cityscapes

Cityscapes is an urban egocentric street-view dataset with high-resolution images (1024×2048 pixels). It contains 2975 images for training, 500 images for validation and 1525 images for testing with a total of 19 classes.

Training settings. For all three segmentation tasks: we use a crop size of 512×1024, a batch size of 16 and train all models for 90k iterations. During inference, we operate on the whole image (1024×2048). Other implementation details largely follow Section 4.1 (panoptic and instance segmentation follow semantic segmentation training settings), except that we use 200 queries for panoptic and instance segmentation models with Swin-L backbone. All other backbones or semantic segmentation models use 100 queries.

Results. In Table VII, we report Mask2Former results obtained with various backbones on Cityscapes for three segmentation tasks and compare it with other state-of-the-art methods *without using extra data*. For panoptic segmentation, Mask2Former with Swin-L backbone outperforms the state-of-the-art Panoptic-DeepLab [11] with SWideRnet [9] using single-scale inference. For semantic segmentation, Mask2Former with Swin-B backbone outperforms the state-of-the-art SegFormer [59].

	method	backbone	search space	epochs	PQ	PQ^{Th}	PQ^{St}	AP^{Th}_{pan}	$mIoU_{pan}$	#params.	FLOPs
CNN backbones	DETR [5]	R50	100 queries	500+25	43.4	48.2	36.3	31.1	-	-	-
		R101	100 queries	500+25	45.1	50.5	37.0	33.0	-	-	-
	K-Net [62]	R50	100 queries	36	47.1	51.7	40.3	-	-	-	-
	Panoptic SegFormer [32]	R50	400 queries	50	50.0	56.1	40.8	-	-	47M	246G
	MaskFormer [14]	R50	100 queries	300	46.5	51.0	39.8	33.0	57.8	45M	181G
		R101	100 queries	300	47.6	52.5	40.3	34.1	59.3	64M	248G
	Mask2Former (ours)	R50	100 queries	50	51.9	57.7	43.0	41.7	61.7	44M	226G
		R101	100 queries	50	**52.6**	**58.5**	**43.7**	**42.6**	**62.4**	63M	293G
Transformer backbones	Max-DeepLab [52]	Max-S	128 queries	216	48.4	53.0	41.5	-	-	62M	324G
		Max-L	128 queries	216	51.1	57.0	42.2	-	-	451M	3692G
	Panoptic SegFormer [32]	PVTv2-B5 [54]	400 queries	50	54.1	60.4	44.6	-	-	101M	391G
	K-Net [62]	Swin-L†	100 queries	36	54.6	60.2	46.0	-	-	-	-
	MaskFormer [14]	Swin-T	100 queries	300	47.7	51.7	41.7	33.6	60.4	42M	179G
		Swin-S	100 queries	300	49.7	54.4	42.6	36.1	61.3	63M	259G
		Swin-B	100 queries	300	51.1	56.3	43.2	37.8	62.6	102M	411G
		Swin-B†	100 queries	300	51.8	56.9	44.1	38.5	63.6	102M	411G
		Swin-L†	100 queries	300	52.7	58.5	44.0	40.1	64.8	212M	792G
	Mask2Former (ours)	Swin-T	100 queries	50	53.2	59.3	44.0	43.3	63.2	47M	232G
		Swin-S	100 queries	50	54.6	60.6	45.7	44.7	64.2	69M	313G
		Swin-B	100 queries	50	55.1	61.0	46.1	45.2	65.1	107M	466G
		Swin-B†	100 queries	50	56.4	62.4	47.3	46.3	67.1	107M	466G
		Swin-L†	*200 queries*	*100*	**57.8**	**64.2**	**48.1**	**48.6**	**67.4**	216M	868G

Table I. **Panoptic segmentation on COCO panoptic `val2017` with 133 categories.** Mask2Former outperforms *all* existing panoptic segmentation models by a large margin with different backbones on all metrics. Our best model sets a new state-of-the-art of 57.8 PQ. Besides PQ for panoptic segmentation, we also report AP^{Th}_{pan} (the AP evaluated on the 80 "thing" categories using *instance segmentation annotation*) and $mIoU_{pan}$ (the mIoU evaluated on the 133 categories for semantic segmentation converted from panoptic segmentation annotation) of the same model trained for panoptic segmentation (**note: we train all our models with panoptic segmentation annotation only**). Backbones pre-trained on ImageNet-22K are marked with †.

method	backbone	PQ	PQ^{Th}	PQ^{St}	SQ	RQ
Max-DeepLab [52]	Max-L	51.3	57.2	42.4	82.5	61.3
Panoptic FCN [31]	Swin-L	52.7	59.4	42.5	-	-
MaskFormer [14]	Swin-L	53.3	59.1	44.5	82.0	64.1
Panoptic SegFormer [32]	PVTv2-B5 [54]	54.4	61.1	44.3	83.3	64.6
K-Net [62]	Swin-L	55.2	61.2	46.2	-	-
Megvii (challenge winner)	-	54.7	64.6	39.8	83.6	64.3
Mask2Former (ours)	Swin-L	**58.3**	**65.1**	**48.1**	**84.1**	**68.6**

Table II. **Panoptic segmentation on COCO panoptic `test-dev` with 133 categories.** Mask2Former, without any bells-and-whistles, outperforms the challenge winner (which uses extra training data, model ensemble, *etc*.) on the `test-dev` set. We only train our model on the COCO `train2017` set with ImageNet-22K pre-trained checkpoint.

B.2. ADE20K

Training settings. For panoptic and instance segmentation, we use the exact same training parameters as we used for semantic segmentation, except that we always use a crop size of 640×640 for all backbones. Other implementation details largely follow Section 4.1 , except that we use 200 queries for panoptic and instance segmentation models with Swin-L backbone. All other backbones or semantic segmentation models use 100 queries.

Results. In Table VIII, we report the results of Mask2Former obtained with various backbones on ADE20K for three segmentation tasks and compare it with other state-of-the-art methods. Mask2Former with Swin-L backbone sets a new state-of-the-art performance on ADE20K for panoptic segmentation. As there are few papers reporting results on ADE20K, we hope this experiment could set up a useful benchmark for future research.

B.3. Mapillary Vistas

Mapillary Vistas is a large-scale urban street-view dataset with 18k, 2k and 5k images for training, validation and testing. It contains images with a variety of resolutions, ranging from 1024×768 to 4000×6000. We only report panoptic and semantic segmentation results for this dataset.

Training settings. For both panoptic and semantic segmentation, we follow the same data augmentation of [14]: standard random scale jittering between 0.5 and 2.0, random horizontal flipping, random cropping with a crop size of 1024×1024 as well as random color jittering. We train

	method	backbone	search space	epochs	AP	AP^S	AP^M	AP^L	$AP^{boundary}$	#params.	FLOPs
CNN backbones	Mask R-CNN [24]	R50	dense anchors	36	37.2	18.6	39.5	53.3	23.1	44M	201G
		R50	dense anchors	400	42.5	23.8	45.0	60.0	28.0	46M	358G
		R101	dense anchors	36	38.6	19.5	41.3	55.3	24.5	63M	266G
		R101	dense anchors	400	43.7	**24.6**	46.4	61.8	29.1	65M	423G
	Mask2Former (ours)	R50	100 queries	50	43.7	23.4	47.2	64.8	30.6	44M	226G
		R101	100 queries	50	**44.2**	23.8	**47.7**	**66.7**	**31.1**	63M	293G
Transformer backbones	QueryInst [20]	Swin-L†	300 queries	50	48.9	30.8	52.6	68.3	33.5	-	-
	Swin-HTC++ [6, 36]	Swin-B†	dense anchors	36	49.1	-	-	-	-	160M	1043G
		Swin-L†	dense anchors	72	49.5	**31.0**	52.4	67.2	34.1	284M	1470G
	Mask2Former (ours)	Swin-T	100 queries	50	45.0	24.5	48.3	67.4	31.8	47M	232G
		Swin-S	100 queries	50	46.3	25.3	50.3	68.4	32.9	69M	313G
		Swin-B	100 queries	50	46.7	26.1	50.5	68.8	33.2	107M	466G
		Swin-B†	100 queries	50	48.1	27.8	52.0	71.1	34.4	107M	466G
		Swin-L†	*200 queries*	*100*	**50.1**	29.9	**53.9**	**72.1**	**36.2**	216M	868G

Table III. **Instance segmentation on COCO `val2017` with 80 categories.** Mask2Former outperforms strong Mask R-CNN [24] baselines with 8× fewer training epochs for both AP and $AP^{boundary}$ [12] metrics. Our best model is also competitive to the state-of-the-art specialized instance segmentation model on COCO and has higher boundary quality. For a fair comparison, we only consider single-scale inference and models trained using only COCO `train2017` set data. Backbones pre-trained on ImageNet-22K are marked with †.

method	backbone	AP	AP50	AP75	AP^S	AP^M	AP^L
QueryInst [20]	Swin-L	49.1	74.2	53.8	**31.5**	51.8	63.2
Swin-HTC++ [6, 36]	Swin-L	50.2	-	-	-	-	-
Swin-HTC++ [6, 36] (multi-scale)	Swin-L	51.1	-	-	-	-	-
Megvii (challenge winner)	-	53.1	76.8	58.6	36.6	56.5	67.7
Mask2Former (ours)	Swin-L	**50.5**	**74.9**	**54.9**	29.1	**53.8**	**71.2**

Table IV. **Instance segmentation on COCO `test-dev` with 80 categories.** Mask2Former is extremely good at segmenting large objects: we can even outperform the challenge winner (which uses extra training data, model ensemble, *etc*.) on AP^L by a large margin without any bells-and-whistles. We only train our model on the COCO `train2017` set with ImageNet-22K pre-trained checkpoint.

our model for 300k iterations with a batch size of 16 using the "poly" learning rate schedule [7]. During inference, we resize the longer side to 2048 pixels. Our panoptic segmentation model with a Swin-L backbone uses 200 queries. All other backbones or semantic segmentation models use 100 queries.

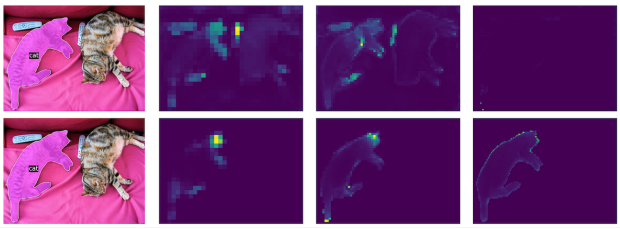
Results. In Table IX, we report Mask2Former results obtained with various backbones on Mapillary Vistas for panoptic and semantic segmentation tasks and compare it with other state-of-the-art methods. Our Mask2Former is very competitive compared to state-of-the art specialized models even if it is not designed for Mapillary Vistas.

C. Additional ablation studies

We perform additional ablation studies of Mask2Former using the same settings that we used in the main paper: a single ResNet-50 backbone [25].

C.1. Convergence analysis

We train Mask2Former with 12, 25, 50 and 100 epochs with either standard scale augmentation (Standard Aug.) [57] or the more recent large-scale jittering augmentation (LSJ Aug.) [18, 23]. As shown in Figure IV, Mask2Former converges in 25 epochs using standard augmentation and almost converges in 50 epochs using large-scale jittering augmentation. This shows that Mask2Former with our proposed Transformer decoder converges faster than models using the standard Transformer decoder: *e.g.*, DETR [5] and MaskFormer [14] require 500 epochs and 300 epochs respectively.



(a) Visualization of cross-attention (top) and masked attention (bottom) for different resolutions.

	1/32		1/16		1/8		average	
	fg	bg	fg	bg	fg	bg	fg	bg
cross-attention	0.23	0.77	0.23	0.77	0.15	0.85	0.20	0.80
masked attention	0.53	0.47	0.61	0.39	0.64	0.36	0.59	0.41

(b) Cumulative attention weights on foreground (fg) and background (bg) regions for different resolutions.

Figure I. Masked attention analysis.

C.2. Masked attention analysis

We quantitatively and qualitatively analyzed the COCO panoptic model with the R50 backbone. First, we visual-

	method	backbone	crop size	mIoU (s.s.)	mIoU (m.s.)	#params.	FLOPs
CNN	MaskFormer [14]	R50	512 × 512	44.5	46.7	41M	53G
		R101	512 × 512	45.5	47.2	60M	73G
	Mask2Former (ours)	R50	512 × 512	47.2	49.2	44M	71G
		R101	512 × 512	**47.8**	**50.1**	63M	90G
Transformer backbones	Swin-UperNet [36, 58]	Swin-L†	640 × 640	-	53.5	234M	647G
	FaPN-MaskFormer [14, 39]	Swin-L†	640 × 640	55.2	56.7	-	-
	BEiT-UperNet [2, 58]	BEiT-L†	640 × 640	-	57.0	502M	-
	MaskFormer [14]	Swin-T	512 × 512	46.7	48.8	42M	55G
		Swin-S	512 × 512	49.8	51.0	63M	79G
		Swin-B	640 × 640	51.1	52.3	102M	195G
		Swin-B†	640 × 640	52.7	53.9	102M	195G
		Swin-L†	640 × 640	54.1	55.6	212M	375G
	Mask2Former (ours)	Swin-T	512 × 512	47.7	49.6	47M	74G
		Swin-S	512 × 512	51.3	52.4	69M	98G
		Swin-B	640 × 640	52.4	53.7	107M	223G
		Swin-B†	640 × 640	53.9	55.1	107M	223G
		Swin-L†	640 × 640	56.1	57.3	215M	403G
		Swin-L-FaPN†	640 × 640	**56.4**	**57.7**	217M	-

Table V. **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], with less than half of the parameters. We report both single-scale (s.s.) and multi-scale (m.s.) inference results. Backbones pre-trained on ImageNet-22K are marked with †.

method	backbone	P.A.	mIoU	score
SETR [64]	ViT-L	78.35	45.03	61.69
Swin-UperNet [36, 58]	Swin-L	78.42	47.07	62.75
MaskFormer [14]	Swin-L	79.36	49.67	64.51
Mask2Former (ours)	Swin-L-FaPN	**79.80**	**49.72**	**64.76**

Table VI. **Semantic segmentation on ADE20K `test` with 150 categories.** Mask2Former outperforms previous state-of-the-art methods on all three metrics: pixel accuracy (P.A.), mIoU, as well as the final test score (average of P.A. and mIoU). We train our model on the union of ADE20K `train` and `val` set with ImageNet-22K pre-trained checkpoint following [14] and use multi-scale inference.

ize the last three attention maps of our model using cross-attention (Figure Ia top) and masked attention (Figure Ia bottom) of a single query that predicts the "cat." With cross-attention, the attention map spreads over the entire image and the region with highest response is outside the object of interest. We believe this is because the softmax used in cross-attention never attains zero, and small attention weights on large background regions start to dominate. Instead, masked attention limits the attention weights to focus on the object. We validate this hypothesis in Table Ib: we compute the cumulative attention weights on foreground (defined by the matching ground truth to each prediction) and background for all queries on the entire COCO `val` set. On average, only 20% of the attention weights in cross-attention focus on the foreground while masked attention increases this ratio to almost 60%. Second, we plot the panoptic segmentation performance using output from each Transformer decoder layer (Figure II). We find masked attention with a single Transformer decoder layer already outperforms cross-attention with 9 layers. We hope the effectiveness of masked attention, together with this analysis, leads to better attention design.

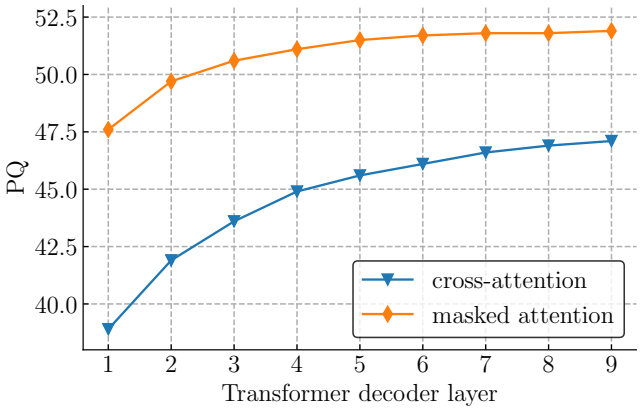


Figure II. Panoptic segmentation performance of each Transformer decoder layer.

C.3. Object query analysis

Object queries play an important role in Mask2Former. We ablate different design choices of object queries including the number of queries and making queries learnable.

Number of queries. We study the effect of different num-

method	backbone	panoptic model				instance model		semantic model	
		PQ (s.s.)	PQ (m.s.)	AP^{Th}_{pan}	$mIoU_{pan}$	AP	AP50	mIoU (s.s.)	mIoU (m.s.)
Panoptic-DeepLab [11]	R50	60.3	-	32.1	78.7	-	-	-	-
	X71 [15]	63.0	64.1	35.3	80.5	-	-	-	-
	SWideRNet [9]	66.4	67.5	40.1	82.2	-	-	-	-
Panoptic FCN [31]	Swin-L†	65.9	-	-	-	-	-	-	-
Segmenter [45]	ViT-L†	-	-	-	-	-	-	-	81.3
SETR [64]	ViT-L†	-	-	-	-	-	-	-	82.2
SegFormer [59]	MiT-B5	-	-	-	-	-	-	-	84.0
Mask2Former (ours)	R50	62.1	-	37.3	77.5	37.4	61.9	79.4	82.2
	R101	62.4	-	37.7	78.6	38.5	63.9	80.1	81.9
	Swin-T	63.9	-	39.1	80.5	39.7	66.9	82.1	83.0
	Swin-S	64.8	-	40.7	81.8	41.8	70.4	82.6	83.6
	Swin-B†	66.1	-	42.8	82.7	42.0	68.8	**83.3**	**84.5**
	Swin-L†	**66.6**	-	**43.6**	**82.9**	**43.7**	**71.4**	**83.3**	84.3

Table VII. **Image segmentation results on Cityscapes `val`.** We report both single-scale (s.s.) and multi-scale (m.s.) inference results for PQ and mIoU. All other metrics are evaluated with *single-scale* inference. Since Mask2Former is an end-to-end model, we only use single-scale inference for instance-level segmentation tasks to avoid the need for further post-processing (*e.g.*, NMS).

method	backbone	panoptic model			instance model				semantic model	
		PQ	AP^{Th}_{pan}	$mIoU_{pan}$	AP	AP^S	AP^M	AP^L	mIoU (s.s.)	mIoU (m.s.)
MaskFormer [14]	R50	34.7	-	-	-	-	-	-	-	-
Panoptic-DeepLab [11]	SWideRNet [9]	37.9*	-	50.0*	-	-	-	-	-	-
Swin-UperNet [36, 58]	Swin-L†	-	-	-	-	-	-	-	-	53.5
MaskFormer [14]	Swin-L†	-	-	-	-	-	-	-	54.1	55.6
FaPN-MaskFormer [14, 39]	Swin-L†	-	-	-	-	-	-	-	55.2	56.7
BEiT-UperNet [2, 58]	BEiT-L†	-	-	-	-	-	-	-	-	57.0
Mask2Former (ours)	R50	39.7	26.5	46.1	26.4	10.4	28.9	43.1	47.2	49.2
	Swin-L†	**48.1**	**34.2**	54.5	**34.9**	**16.3**	**40.0**	**54.7**	56.1	57.3
	Swin-L-FaPN†	46.2	33.2	**55.4**	33.4	14.6	37.6	54.6	**56.4**	**57.7**

Table VIII. **Image segmentation results on ADE20K `val`.** Mask2Former is competitive to specialized models on ADE20K. Panoptic segmentation models use single-scale inference by default, multi-scale numbers are marked with *. For semantic segmentation, we report both single-scale (s.s.) and multi-scale (m.s.) inference results.

ber of queries for three image segmentation tasks in Table Xa. For instance and semantic segmentation, using 100 queries achieves the best performance, while using 200 queries can further improve panoptic segmentation results. As panoptic segmentation is a combination of instance and semantic segmentation, it has more segments per image than the other two tasks. This ablation suggests that picking the number of queries for Mask2Former may depend on the number of segments per image for a particular task or dataset.

Learnable queries. An object query consists of two parts: object query features and object query positional embeddings. Object query features are only used as the initial input to the Transformer decoder and are updated through decoder layers; whereas query positional embeddings are added to query features in every Transformer decoder layer when computing the attention weights. In DETR [5], query features are zero-initialized and query positional embeddings are learnable. Furthermore, there is no direct supervision on these query features before feeding them into the Transformer (since they are zero vectors). In our Mask2Former, we still make query positional embeddings learnable. In addition, we make query features learnable as well and directly apply losses on these learnable query features before feeding them into the Transformer decoder.

In Table Xb, we compare our learnable query features with zero-initialized query features in DETR. We find it is important to directly supervise object queries even before feeding them into the Transformer decoder. Learnable queries *without* supervision perform similarly well as zero-initialized queries in DETR.

C.4. MaskFormer *vs*. Mask2Former

Mask2Former builds upon the same meta architecture as MaskFormer [14] with two major differences: 1) We use more advanced training parameters summarized in Table XIa; and 2) we propose a new Transformer decoder with masked attention, instead of using the standard Transformer decoder, as well as some optimization improvements summarized in Table XIb. To better understand Mask2Former's improvements over MaskFormer, we perform ablation studies on training parameter improvements and Transformer decoder improvements in isolation.

In Table XIc, we study our new training parameters. We

method	backbone	panoptic model PQ	panoptic model mIoU_{pan}	semantic model mIoU (s.s.)	semantic model mIoU (m.s.)
Panoptic-DeepLab [11]	ensemble	42.2*	58.7*	-	-
	SWideRNet [9]	43.7	59.4	-	-
	SWideRNet [9]	44.8*	60.0*	-	-
Panoptic FCN [31]	Swin-L†	**45.7**	-	-	-
MaskFormer [14]	R50	-	-	53.1	55.4
HMSANet [48]	HRNet [53]	-	-	-	61.1
Mask2Former (ours)	R50	36.3	50.7	57.4	59.0
	Swin-L†	**45.5**	**60.8**	**63.2**	**64.7**

Table IX. **Image segmentation results on Mapillary Vistas `val`.** Mask2Former is competitive to specialized models on Mapillary Vistas. Panoptic segmentation models use single-scale inference by default, multi-scale numbers are marked with *. For semantic segmentation, we report both single-scale (s.s.) and multi-scale (m.s.) inference results.

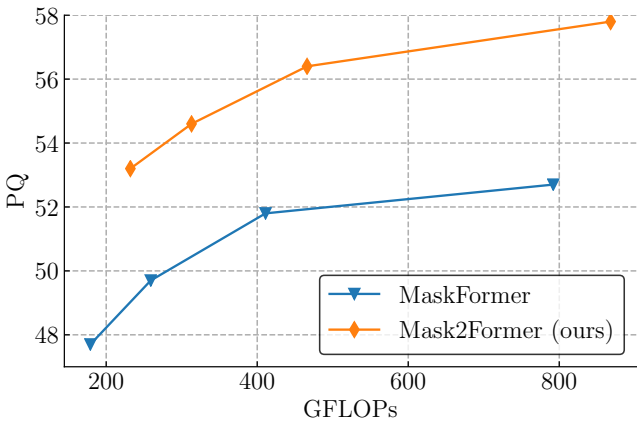


Figure III. MaskFormer [14] *vs*. Mask2Former (ours) with different Swin Transformer backbones.

train the MaskFormer model with either its original training parameters in [14] or our new training parameters. We observe significant improvements of using our new training parameters for MaskFormer as well. This shows the new training parameters are also generally applicable to other models.

In Table XId, we study our new Transformer decoder. We train a MaskFormer model and a Mask2Former model with the exact same backbone, *i.e*., a ResNet-50; pixel decoder, *i.e*., a FPN; and training parameters. That is, the only difference is in the Transformer decoder, summarized in Table XIb. We observe improvements for all three tasks, suggesting that the new Transformer decoder itself is indeed better than the standard Transformer decoder.

While computational efficiency was not our primary goal, we find that Mask2Former actually has a better compute-performance trade-off compared to MaskFormer (Figure III). Even the lightest instantiation of Mask2Former outperforms the heaviest MaskFormer instantiation, using $\frac{1}{4}^{\text{th}}$ the FLOPs.

D. Visualization

We visualize sample predictions of the Mask2Former model with Swin-L [36] backbone on three tasks: COCO panoptic `val2017` set for panoptic segmentation (57.8 PQ) in Figure V, COCO `val2017` set for instance segmentation (50.1 AP) in Figure VI and ADE20K validation set for semantic segmentation (57.7 mIoU, multi-scale inference) in Figure VII.

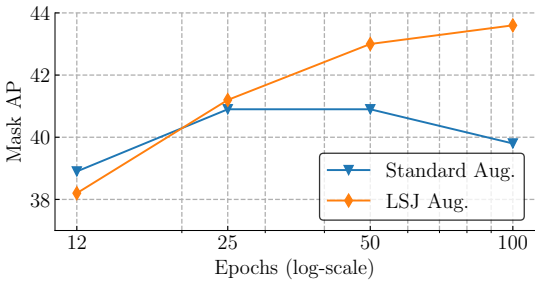


Figure IV. **Convergence analysis.** We train Mask2Former with different epochs using either standard scale augmentation (Standard Aug.) [57] or the more recent large-scale jittering augmentation (LSJ Aug.) [18, 23]. Mask2Former converges in 25 epochs using standard augmentation and almost converges in 50 epochs using large-scale jittering augmentation. Using LSJ also improves performance with longer training epochs (*i.e.*, with more than 25 epochs).

	AP (COCO)	PQ (COCO)	mIoU (ADE20K)	FLOPs (COCO)
50	42.4	50.5	46.2	217G
100	**43.7**	51.9	**47.2**	226G
200	43.5	**52.2**	47.0	246G
300	43.5	52.1	46.5	265G
1000	40.3	50.7	44.8	405G

(a) **Number of queries ablation.** For instance and semantic segmentation, using 100 queries achieves the best performance while using 200 queries can further improve panoptic segmentation results.

	AP (COCO)	PQ (COCO)	mIoU (ADE20K)	FLOPs (COCO)
zero-initialized (DETR [5])	42.9	51.2	45.5	226G
learnable *w/o* supervision	42.9	51.2	47.0	226G
learnable *w/* supervision	**43.7**	**51.9**	**47.2**	226G

(b) **Learnable queries ablation.** It is important to supervise object queries before feeding them into the Transformer decoder. Learnable queries *without* supervision perform similarly well as zero-initialized queries in DETR.

Table X. **Analysis of object queries.** Table Xa: ablation on number of queries. Table Xb: ablation on using learnable queries.

training parameters	MaskFormer	Mask2Former (ours)
learning rate	0.0001	0.0001
weight decay	0.0001	0.05
batch size	16*	16
epochs	75*	50
data augmentation	standard scale aug. w/ crop	LSJ aug.
λ_{cls}	1.0	2.0
λ_{focal} / λ_{ce}	20.0 / -	- / 5.0
λ_{dice}	1.0	5.0
mask loss	mask	12544 sampled points

(a) Comparison of training parameters for MaskFormer [14] and our Mask2Former on the COCO dataset. *: in the original MaskFormer implementation, the model is trained with a batch size of 64 for 300 epochs. We find MaskFormer achieves similar performance when trained with a batch size of 16 for 75 epochs, *i.e.*, the same number of iterations with a smaller batch size.

Transformer decoder	MaskFormer	Mask2Former (ours)
# of layers	6	9
single layer	SA-CA-FFN	MA-SA-FFN
dropout	0.1	0.0
feature resolution	$\{1/32\} \times 6$	$\{1/32, 1/16, 1/8\} \times 3$
input query features	zero init.	learnable
query p.e.	learnable	learnable

(b) Comparison of Transformer decoder in MaskFormer [14] and our Mask2Former. SA: self-attention, CA: cross-attention, FFN: feed-forward network, MA: masked attention, p.e.: positional embedding.

model	training params.	AP (COCO)	PQ (COCO)	mIoU (ADE20K)
MaskFormer	MaskFormer	34.0	46.5	44.5
MaskFormer	Mask2Former	37.8 (+3.8)	48.2 (+1.7)	45.3 (+0.8)

(c) Improvements from better **training parameters**.

Transformer decoder	pixel decoder	AP (COCO)	PQ (COCO)	mIoU (ADE20K)
MaskFormer	FPN	37.8	48.2	45.3
Mask2Former	FPN	41.5 (+3.7)	50.7 (+2.5)	45.6 (+0.3)

(d) Improvements from better **Transformer decoder**.

Table XI. **MaskFormer *vs.* Mask2Former.** Table XIa and Table XIb provide an in-depth comparison between MaskFormer and our Mask2Former settings. Table XIc: MaskFormer benefits from our new training parameters as well. Table XId: Comparison between MaskFormer and our Mask2Former with the exact same backbone, pixel decoder and training parameters. The improvements solely come from a better Transformer decoder.

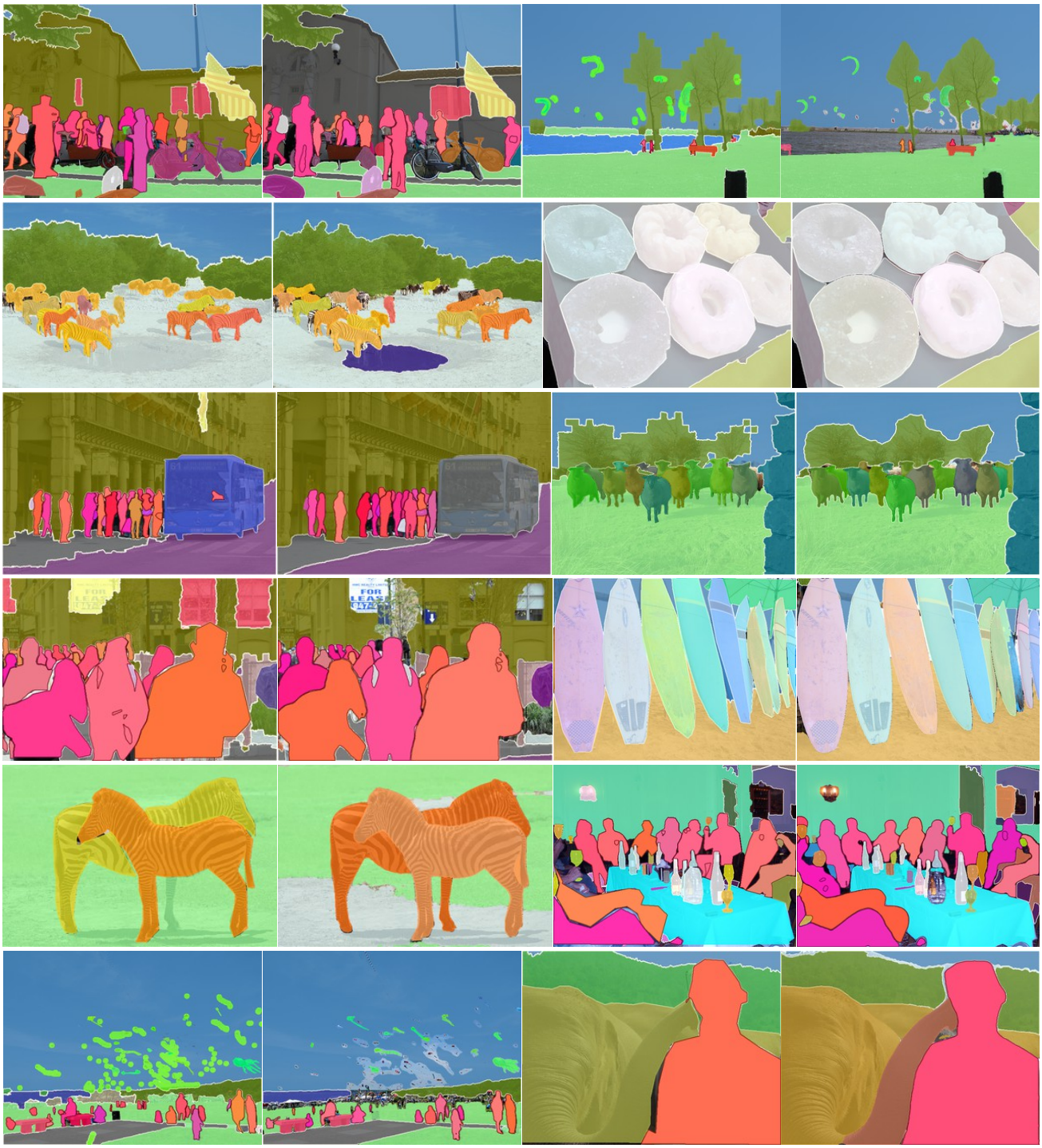


Figure V. Visualization of **panoptic segmentation** predictions on the COCO panoptic dataset: Mask2Former with Swin-L backbone which achieves 57.8 PQ on the validation set. First and third columns: ground truth. Second and fourth columns: prediction. **Last row shows failure cases.**

Figure VI. Visualization of **instance segmentation** predictions on the COCO dataset: Mask2Former with Swin-L backbone which achieves 50.1 AP on the validation set. First and third columns: ground truth. Second and fourth columns: prediction. **Last row shows failure cases.** We show predictions with confidence scores greater than 0.5.

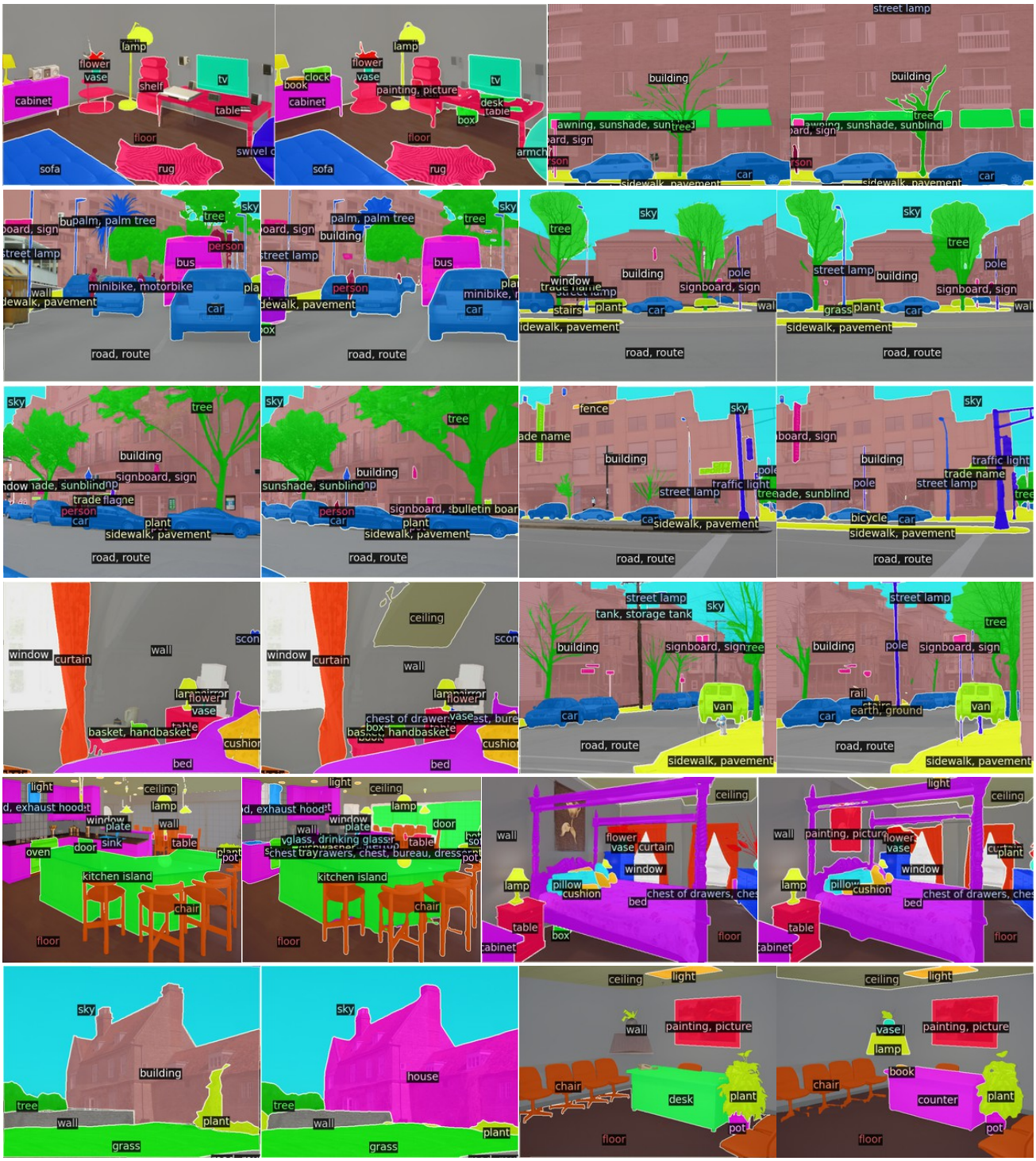


Figure VII. Visualization of **semantic segmentation** predictions on the ADE20K dataset: Mask2Former with Swin-L backbone which achieves 57.7 mIoU (multi-scale) on the validation set. First and third columns: ground truth. Second and fourth columns: prediction. **Last row shows failure cases.**