OneFormer: One Transformer to Rule Universal Image Segmentation

Jitesh Jain[1,2], Jiachen Li[1*], MangTik Chiu[1*], Ali Hassani[1], Nikita Orlov[3], Humphrey Shi[1,3]
[1]SHI Labs @ U of Oregon & UIUC, [2]IIT Roorkee, [3]Picsart AI Research (PAIR)

https://github.com/SHI-Labs/OneFormer

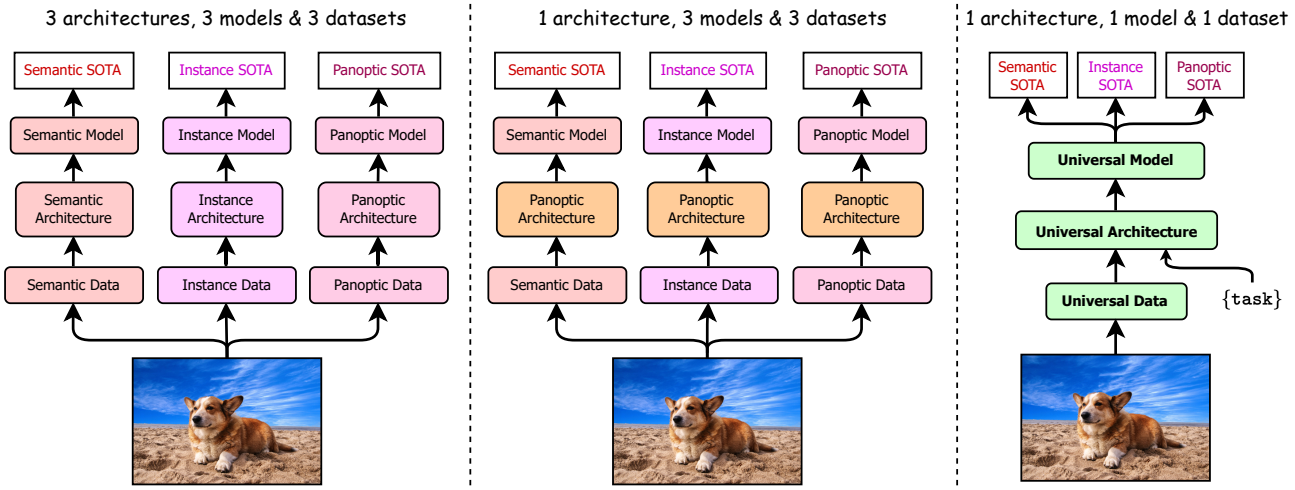

Figure 1. **A Path to Universal Image Segmentation.** (a) Traditional segmentation methods developed specialized architectures and models for each task to achieve top performance. (b) Recently, new panoptic architectures [12, 61] used the same architecture to achieve top performance across different tasks. However, they still need to train different models for different tasks, resulting in a semi-universal approach. (c) We propose a unique multi-task universal architecture with a task-conditioned joint training strategy that sets new state-of-the-arts across semantic, instance and panoptic segmentation tasks with a single model, unifying segmentation across architecture, model and dataset. Our work significantly reduces the underlying resource requirements and makes segmentation more universal and accessible.

Abstract

Universal Image Segmentation is not a new concept. Past attempts to unify image segmentation in the last decades include scene parsing, panoptic segmentation, and, more recently, new panoptic architectures. However, such panoptic architectures do not truly unify image segmentation because they need to be trained individually on the semantic, instance, or panoptic segmentation to achieve the best performance. Ideally, a truly universal framework should be trained only once and achieve SOTA performance across all three image segmentation tasks. To that end, we propose OneFormer, a universal image segmentation framework that unifies segmentation with a multi-task train-once design. We first propose a task-conditioned joint training strategy that enables training on ground truths of each domain (semantic, instance, and panoptic segmentation) within a single multi-task training process. Secondly, we introduce a task token to condition our model on the task at hand, making our model task-dynamic to support multi-task training and inference. Thirdly, we propose using a query-text contrastive loss during training to establish better inter-task and inter-class distinctions. Notably, our single OneFormer model outperforms specialized Mask2Former models across all three segmentation tasks on ADE20k, Cityscapes, and COCO, despite the latter being trained on each of the three tasks individually with three times the resources. With new ConvNeXt and DiNAT backbones, we observe even more performance improvement. We believe OneFormer is a significant step towards making image segmentation more universal and accessible. To support further research, we open-source our code and models at https://github.com/SHI-Labs/OneFormer.

1. Introduction

Image Segmentation is the task of grouping pixels into multiple segments. Such grouping can be semantic-based (*e.g.*, road, sky, building), or instance-based (objects with well-defined boundaries). Earlier segmentation ap-

proaches [7,23,40] tackled these two segmentation tasks individually, with specialized architectures and therefore separate research effort into each. In a recent effort to unify semantic and instance segmentation, Kirillov *et al.* [29] proposed panoptic segmentation, with pixels grouped into an amorphous segment for amorphous background regions (labeled "stuff") and distinct segments for objects with well-defined shape (labeled "thing"). This effort, however, led to new specialized panoptic architectures [11] instead of unifying the previous tasks (see Fig. 1a). More recently, the research trend shifted towards unifying image segmentation with new panoptic architectures, such as K-Net [61], MaskFormer [13], and Mask2Former [12]. Such panoptic architectures can be trained on all three tasks and obtain high performance without changing architecture. They do need to, however, be trained individually on each task to achieve the best performance (see Fig. 1b). The individual training policy requires extra training time and produces different sets of model weights for each task. In that regard, they can only be considered a semi-universal approach. For example, Mask2Former [12] is trained for 160K iterations on ADE20K [15] for each of the semantic, instance, and panoptic segmentation tasks to obtain the best performance for each task, yielding a total of 480k iterations in training, and three models to store and host for inference.

In an effort to truly unify image segmentation, we propose a multi-task universal image segmentation framework (**OneFormer**), which outperforms existing state-of-the-arts on all three image segmentation tasks (see Fig. 1c), by only training once on one panoptic dataset. Through this work, we aim to answer the following questions:

(i) *Why are existing panoptic architectures [12, 13] not successful with a single training process or model to tackle all three tasks?* We hypothesize that existing methods need to train individually on each segmentation task due to the absence of task guidance in their architectures, making it challenging to learn the inter-task domain differences when trained jointly or with a single model. To tackle this challenge, we introduce a task input token in the form of text: "the task is {`task`}", to condition the model on the task in focus, making our architecture task-guided for training, and task-dynamic for inference, all with a single model. We uniformly sample {`task`} from {`panoptic, instance, semantic`} and the corresponding ground truth during our joint training process to ensure our model is unbiased in terms of tasks. Motivated by the ability of panoptic [29] data to capture both semantic and instance information, we derive the semantic and instance labels from the corresponding panoptic annotations during training. Consequently, we only need panoptic data during training. Moreover, our joint training time, model parameters, and FLOPs are comparable to the existing methods, decreasing training time and storage requirements up to 3×, making image segmentation less resource intensive and more accessible.

(ii) *How can the multi-task model better learn inter-task and inter-class differences during the single joint training process?* Following the recent success of transformer frameworks [3,12,21,22,27,38,60] in computer vision, we formulate our framework as a transformer-based approach, which can be guided through the use of query tokens. To add task-specific context to our model, we initialize our queries as repetitions of the task token (obtained from the task input) and compute a query-text contrastive loss [43, 57] with the text derived from the corresponding ground-truth label for the sampled task as shown in Fig. 2. We hypothesize that a contrastive loss on the queries helps guide the model to be more task-sensitive. Furthermore, it also helps reduce the category mispredictions to a certain extent.

We evaluate OneFormer on three major segmentation datasets: ADE20K [15], Cityscapes [14], and COCO [34], each with all three (semantic, instance, and panoptic) segmentation tasks. OneFormer sets the new state of the arts for all three tasks with a single jointly trained model. To summarize, our main contributions are:

- We propose OneFormer, the first multi-task universal image segmentation framework based on transformers that need to be trained only once with a single universal architecture, a single model, and on a single dataset, to outperform existing frameworks across semantic, instance, and panoptic segmentation tasks, despite the latter need to be trained separately on each task using multiple times of the resources.
- OneFormer uses a task-conditioned joint training strategy, uniformly sampling different ground truth domains (semantic, instance, or panoptic) by deriving all labels from panoptic annotations to train its multi-task model. Thus, OneFormer actually achieves the orignial unification goal of panoptic segmentation [29].
- We validate OneFormer through extensive experiments on three major benchmarks: ADE20K [15], Cityscapes [14], and COCO [34]. OneFormer sets a new state-of-the-art performance on all three segmentation tasks compared with methods using the standard Swin-L [38] backbone, and improves even more with new ConvNeXt [39] and DiNAT [21] backbones.

2. Related Work

2.1. Image Segmentation

Image segmentation is one of the most fundamental tasks in image processing and computer vision. Traditional works usually tackle one of the three image segmentation tasks with specialized network architectures (Fig. 1a).

Semantic Segmentation. Semantic segmentation was long tackled as a pixel classification problem with CNNs [6,

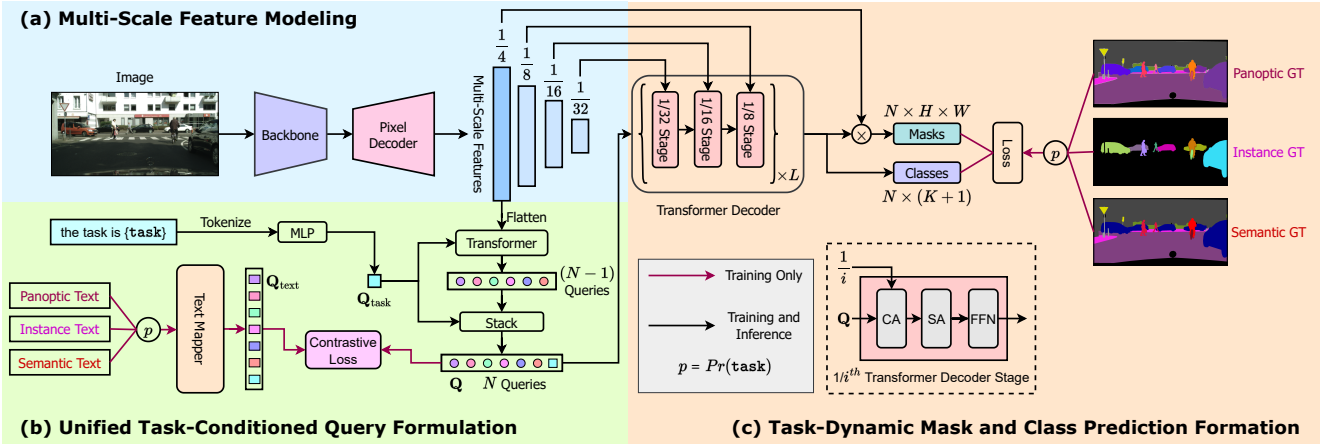

Figure 2. **OneFormer Framework Architecture.** (a) We extract multi-scale features for an input image using a backbone, followed by a pixel decoder. (b) We formulate a unified set of $N-1$ task-conditioned object queries with guidance from the task token ($\mathbf{Q}_{\text{task}}$) and flattened 1/4-scale features inside a transformer [49]. Next, we concatenate $\mathbf{Q}_{\text{task}}$ with the $N-1$ queries from the transformer. We uniformly ($p = 1/3$) sample the task during training and generate the corresponding text queries ($\mathbf{Q}_{\text{text}}$) using a text mapper (Fig. 4). We calculate a query-text contrastive loss to learn the inter-task distinctions. We can drop the text mapper during inference, thus, making our model parameter efficient. (c) We use a multi-stage L-layer transformer decoder to obtain the task-dynamic class and mask predictions.

7, 10, 40]. More recent works [26, 27, 44, 56] have shown the success of transformer-based methods in semantic segmentation following its success in language and vision [3, 49]. Among them, MaskFormer [13] treated semantic segmentation as a mask classification problem following early works [4, 16, 20], through using a transformer decoder with object queries [3]. We also formulate semantic segmentation as a mask classification problem.

Instance Segmentation. Traditional instance segmentation methods [2, 5, 23] are also formulated as mask classifiers, which predict binary masks and a class label for each mask. We also formulate instance segmentation as a mask classification problem.

Panoptic Segmentation. Panoptic Segmentation [29] was proposed to unify instance and semantic segmentation. One of the earliest architectures in this scope was Panoptic-FPN [28], which introduced separate instance and semantic task branches. Works that followed significantly improved performance with transformer-based architectures [12, 13, 50, 51, 59, 60]. Despite the progress made so far, panoptic segmentation models are still behind in performance compared to individual instance and semantic segmentation models, therefore not living up to their full unification potential. Motivated by this, we design our OneFormer to be trained with panoptic annotations only.

2.2. Universal Image Segmentation

The concept of universal image segmentation has existed for some time, starting with image and scene parsing [47, 48, 58], followed by panoptic segmentation as an effort to unify semantic and instance segmentation [29]. More recently, promising architectures [12, 13, 61] designed specifically for panoptic segmentation have emerged which also perform well on semantic and instance segmentation tasks. K-Net [61], a CNN, uses dynamic learnable instance and semantic kernels with bipartite matching. MaskFormer [13] is a transformer-based architecture, serving as a mask classifier. It was inspired by DETR's [3] reformulation of object detection in the scope of transformers, where the image is fed to the encoder, and the decoder produces proposals based on queries. Mask2Former [12] improved upon MaskFormer with learnable queries, deformable multi-scale attention [64] in the decoder, a masked cross-attention and set the new state of the art on all three tasks. Unfortunately, it requires training the model individually on each task to achieve the best performance. Therefore, there remains a gap in truly unifying the three segmentation tasks. To the best of our knowledge, OneFormer is the first framework to beat state of the art on all three image segmentation tasks with a single universal model.

2.3. Transformer-based Architectures

Architectures based on the transformer encoder-decoder structure [3, 31, 36, 64] have proved effective in object detection since the introduction of DETR [3]. Mask2Former [12, 13] demonstrated the effectiveness of such architectures for image segmentation with a mask classification formulation. Inspired by this success, we also formulate our framework as a query-based mask classification task. Additionally, we claim that calculating a query-text contrastive loss [43, 57] on the task-guided queries can help the model learn inter-task differences and reduce the category mispredictions in the model outputs. Concurrent to our work, LMSeg [1] uses text derived from multiple datasets' taxonomy to calculate a

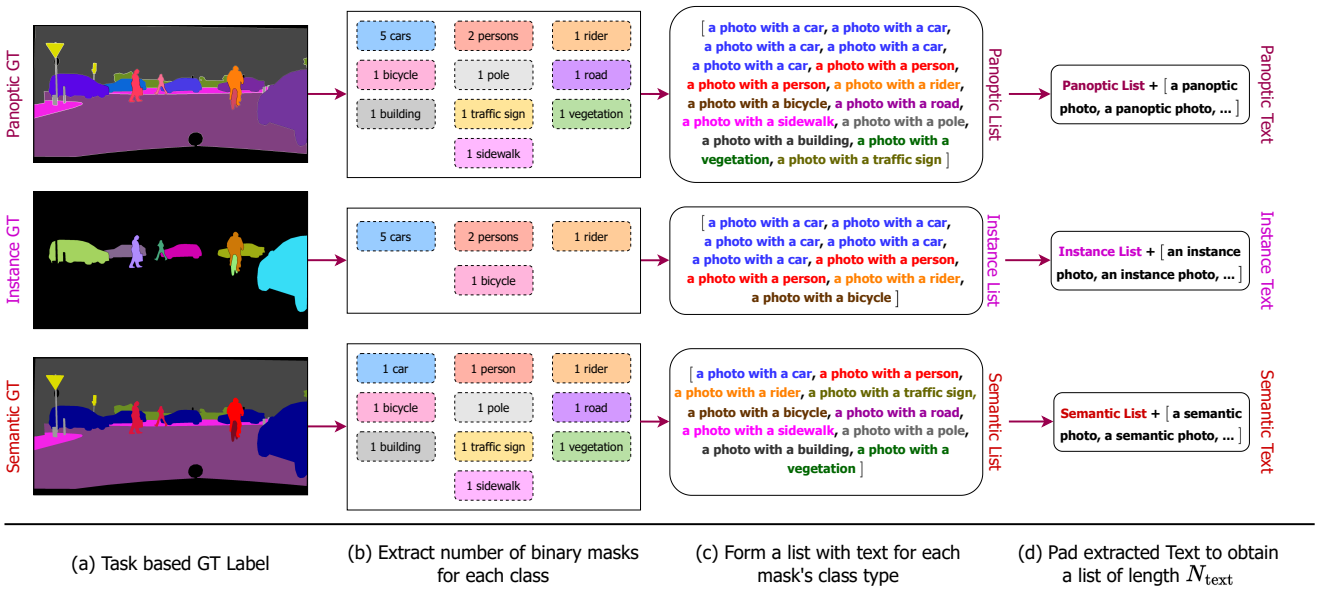

Figure 3. **Input Text Formation.** (a) We uniformly sample the `task` during training. (b) Following the task selection, we extract the number of distinct binary masks for each class to be detected from the corresponding GT label. (c) We form a list with text descriptions for each mask using the template "a photo with a {CLS}", where CLS represents the corresponding class name for the object mask. (d) Finally, we pad the text list to a constant length of N_{text} using "a/an {`task`} photo" entries which represent the `no-object` detections; where `task` $\in$ {panoptic, instance, semantic}.

query-text contrastive loss and tackle the multi-dataset segmentation training challenge. Unlike LMSeg [1], our work focuses on multiple tasks and uses the classes present in the training sample's ground-truth label to calculate the query-text contrastive loss.

3. Method

In this section, we introduce OneFormer, a universal image segmentation framework jointly trained on the panoptic, semantic, and instance segmentation and outperforms individually trained models. We provide an overview of OneFormer in Fig. 2. OneFormer uses two inputs: sample image and task input of the form "the task is {`task`}". During our single joint training process, the `task` is uniformly sampled from {panoptic, instance, semantic} for each image. Firstly, we extract multi-scale features from the input image using a backbone and a pixel decoder. We tokenize the task input to obtain a 1-D task token used to condition the object queries and, consequently, our model on the task for each input. Additionally, we create a text list representing the number of binary masks for each class present in the GT label and map it to text query representations. Note that the text list depends on the input image and the {`task`}. For supervision of the model's task-dynamic predictions, we derive the corresponding ground-truths from panoptic annotations. As the ground truth is task-dependent, we calculate a query-text contrastive loss between the object and text queries to ensure there is task distinction in the object queries. The object queries and multi-scale features are fed into a transformer decoder to produce final predictions. We provide more details in the following sections.

3.1. Task Conditioned Joint Training

Existing semi-universal architectures for image segmentation [12, 13, 61] face a significant drop in performance when jointly trained on all three segmentation tasks (Tab. 7). We attribute their failure to tackle the multi-task challenge to the absence of task-conditioning in their architecture.

We tackle the multi-task train-once challenge for image segmentation using a task-conditioned joint training strategy. Particularly, we first uniformly sample the `task` from {panoptic, semantic, instance} for the GT label. We realize the unification potential of panoptic annotations [29] by deriving the `task`-specific labels from the panoptic annotations, thus, using only one set of annotations.

Next, we extract a set of binary masks for each category present in the image from the task-specific GT label, *i.e.*, semantic task guarantees only one amorphous binary mask for each class present in the image, whereas, instance task signifies non-overlapping binary masks for only thing classes, ignoring the stuff regions. Panoptic task denotes a single amorphous mask for stuff classes and non-overlapping masks for thing classes as shown in Fig. 3. Subsequently, we iterate over the set of masks to create a list of text ($\mathbf{T}_{\text{list}}$) with a template "a photo with a {CLS}", where CLS is the class name for the corresponding binary mask. The number of binary masks per sample varies over the dataset. There-

fore, we pad $\mathbf{T}_{\text{list}}$ with "a/an {task} photo" entries to obtain a padded list ($\mathbf{T}_{\text{pad}}$) of constant length N_{text}, with padded entries representing no-object masks. We later use $\mathbf{T}_{\text{pad}}$ for computing a query-text contrastive loss (Sec. 3.3).

We condition our architecture on the task using a task input ($\mathbf{I}_{\text{task}}$) with the template "the task is {task}", which is tokenized and mapped to a task-token ($\mathbf{Q}_{\text{task}}$). We use $\mathbf{Q}_{\text{task}}$ to condition OneFormer on the task (Sec. 3.2).

3.2. Query Representations

During training, we use two sets of queries in our architecture: text queries ($\mathbf{Q}_{\text{text}}$) and object queries ($\mathbf{Q}$). $\mathbf{Q}_{\text{text}}$ is the text-based representation for the segments in the image, while $\mathbf{Q}$ is the image-based representation.

To obtain $\mathbf{Q}_{\text{text}}$, we first tokenize the text entries $\mathbf{T}_{\text{pad}}$ and pass the tokenized representations through a text-encoder [57], which is a 6-layer transformer [49]. The encoded N_{text} text embeddings represent the number of binary masks and their corresponding classes in the input image. We further concatenate a set of N_{ctx} learnable text context embeddings ($\mathbf{Q}_{\text{ctx}}$) to the encoded text embeddings to obtain the final N text queries ($\mathbf{Q}_{\text{text}}$), as shown in Fig. 4. Our motivation behind using $\mathbf{Q}_{\text{ctx}}$ is to learn a unified textual context [62, 63] for a sample image. We only use the text queries during training; therefore, we can drop the text mapper module during inference to reduce the model size.

To obtain $\mathbf{Q}$, we first initialize the object queries ($\mathbf{Q}'$) as a $N-1$ times repetitions of the task-token ($\mathbf{Q}_{\text{task}}$). Then, we update $\mathbf{Q}'$ with guidance from the flattened 1/4-scale features inside a 2-layer transformer [3, 49]. The updated $\mathbf{Q}'$ from the transformer (rich with image-contextual information) is concatenated with $\mathbf{Q}_{\text{task}}$ to obtain a task-conditioned representation of N queries, $\mathbf{Q}$. Unlike the vanilla all-zeros or random initialization [3], the task-guided initialization of the queries and the concatenation with $\mathbf{Q}_{\text{task}}$ is critical for the model to learn multiple segmentation tasks (Sec. 4.3).

3.3. Task Guided Contrastive Queries

Developing a single model for all three segmentation tasks is challenging due to the inherent differences among the three tasks. The meaning of the object queries, $\mathbf{Q}$, is task-dependent. Should the queries focus only on the thing classes (instance segmentation), or should the queries predict only one amorphous object for each class present in the image (semantic segmentation) or a mix of both (panoptic segmentation)? Existing query-based architectures [12, 13] do not take such differences into account and hence, fail at effectively training a single model on all three tasks.

To this end, we propose to calculate a query-text contrastive loss using $\mathbf{Q}$ and $\mathbf{Q}_{\text{text}}$. We use $\mathbf{T}_{\text{pad}}$ to obtain the text queries representation, $\mathbf{Q}_{text}$, where $\mathbf{T}_{pad}$ is a list of textual representations for each mask-to-be-detected in a given image with "a/an {task} photo" representing the no-object

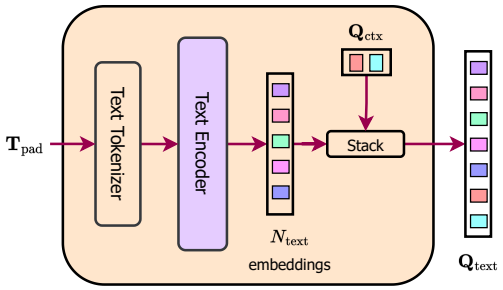

Figure 4. **Text Mapper.** We tokenize and then encode the input text list ($\mathbf{T}_{\text{pad}}$) using a 6-layer transformer text encoder [49, 57] to obtain a set of N_{text} embeddings. We concatenate a set of N_{ctx} learnable embeddings to the encoded representations to obtain the final N text queries ($\mathbf{Q}_{\text{text}}$). The N text queries stand for a text-based representation of the objects present in an image.

detections in $\mathbf{Q}$ [3]. Thus, the text queries align with the purpose of object queries, representing the objects/segments present [3] in an image. Therefore, we can successfully learn the inter-task distinctions in the query representations using a contrastive loss between the ground truth-derived text and object queries. Moreover, contrastive learning on the queries enables us to attend to inter-class differences and reduce category misclassifications.

$$
\begin{aligned}
\mathcal{L}_{\mathbf{Q}\to\mathbf{Q}_{\text{text}}} &= -\frac{1}{B}\sum_{i=1}^{B}\log\frac{\exp(q_i^{obj}\odot q_i^{txt}/\tau)}{\sum_{j=1}^{B}\exp(q_i^{obj}\odot q_j^{txt}/\tau)}, \\
\mathcal{L}_{\mathbf{Q}_{\text{text}}\to\mathbf{Q}} &= -\frac{1}{B}\sum_{i=1}^{B}\log\frac{\exp(q_i^{txt}\odot q_i^{obj}/\tau)}{\sum_{j=1}^{B}\exp(q_i^{txt}\odot q_j^{obj}/\tau)} \\
\mathcal{L}_{\mathbf{Q}\leftrightarrow\mathbf{Q}_{\text{text}}} &= \mathcal{L}_{\mathbf{Q}\to\mathbf{Q}_{\text{text}}} + \mathcal{L}_{\mathbf{Q}_{\text{text}}\to\mathbf{Q}}
\end{aligned}
\tag{1}
$$

Considering that we have a batch of B object-text query pairs $\{(q_i^{obj}, x_i^{txt})\}_{i=1}^{B}$, where q_i^{obj} and q_i^{txt} are the corresponding object and text queries, respectively, of the i-th pair, we measure the similarity between the queries by calculating a dot product. The total contrastive loss is composed of two losses [57]: (i) an object-to-text contrastive loss ($\mathcal{L}_{\mathbf{Q}\to\mathbf{Q}_{\text{text}}}$) and; (ii) a text-to-object contrastive loss ($\mathcal{L}_{\mathbf{Q}_{\text{text}}\to\mathbf{Q}}$) as shown in Eq. (1). τ is a learnable temperature parameter to scale the contrastive logits.

3.4. Other Architecture Components

Backbone and Pixel Decoder: We use the widely used ImageNet [30] pre-trained backbones [21, 38, 39] to extract multi-scale feature representations from the input image. Our pixel decoder aids the feature modeling by gradually upsampling the backbone features. Motivated by the recent success of multi-scale deformable attention [12, 64], we use the same Multi-Scale Deformable Transformer (MSDeformAttn) based architecture for our pixel decoder.

Transformer Decoder: We use a multi-scale strategy [12] to utilize the higher resolution maps inside our transformer

Method	Backbone	#Params	#FLOPs	#Queries	Crop Size	Iters	PQ	AP	mIoU (s.s.)	mIoU (m.s.)
Individual Training										
UPerNet‡ [55]	SwinV2-L† [37]	—	—	—	640×640	40k	—	—	—-	55.9
SeMask Mask2Former [27]	SeMask Swin-L† [27]	223M	426G	200	640×640	160k	—	—	56.4	57.5
UPerNet + K-Net [61]	Swin-L† [38]	—	—	—	640×640	160k	—	—	—	54.3
MaskFormer [13]	Swin-L† [38]	212M	375G	100	640×640	160k	—	—	54.1	55.6
Mask2Former-Panoptic* [12]	Swin-L† [38]	216M	413G	200	640×640	160k	48.7	34.2	54.5	—
Mask2Former-Instance [12]	Swin-L† [38]	216M	411G	200	640×640	160k	—	34.9	—	—
Mask2Former-Semantic [12]	Swin-L† [38]	215M	403G	100	640×640	160k	—	—	56.1	57.3
kMaX-DeepLab‡‡ [60]	ConvNeXt-L† [39]	232M	333G	256	641×641	100k	48.7	—	54.8	—
kMaX-DeepLab‡‡ [60]	ConvNeXt-L† [39]	232M	1302G	256	1281×1281	100k	50.9	—	55.2	—
UPerNet‡‡ [55]	SwinV2-G†† [37]	>3B	—	—	640×640	80k	—	—	59.1	—
Mask2Former‡‡ [12]	BEiT-3†† [52]	1.9B	—	—	896×896	—	—	—	62.0	62.8
Joint Training										
OneFormer	Swin-L† [38]	219M	436G	250	640×640	160k	**49.8**	**35.9**	**57.0**	**57.7**
OneFormer	Swin-L† [38]	219M	801G	250	896×896	160k	**51.1**	**37.6**	**57.4**	**58.3**
OneFormer	Swin-L† [38]	219M	1597G	250	1280×1280	160k	**51.4**	**37.8**	**57.0**	**57.7**
OneFormer	ConvNeXt-L† [39]	220M	389G	250	640×640	160k	**50.0**	**36.2**	**56.6**	**57.4**
OneFormer	ConvNeXt-XL† [39]	372M	607G	250	640×640	160k	**50.1**	**36.3**	**57.4**	**58.8**
OneFormer	DiNAT-L† [21]	223M	359G	250	640×640	160k	**50.5**	**36.0**	**58.3**	**58.4**
OneFormer	DiNAT-L† [21]	223M	678G	250	896×896	160k	**51.2**	**36.8**	**58.1**	**58.6**
OneFormer	DiNAT-L† [21]	223M	1369G	250	1280×1280	160k	**51.5**	**37.1**	**58.2**	**58.7**

Table 1. **SOTA Comparison on the ADE20K val set.** †: backbones pretrained on ImageNet-22K, *: 0.5 confidence threshold; ‡: trained with batch size 32, ‡‡: trained with batch size 64. OneFormer outperforms the individually trained Mask2Former [12] on all metrics. Mask2Former's performance with 250 queries is not listed, as its performance degrades with 250 queries. We compute FLOPs using the corresponding crop size.

decoder. Specifically, we feed the object queries ($\mathbf{Q}$) and the multi-scale outputs from the pixel decoder (F_i), $i \in \{1/4, 1/8, 1/16, 1/32\}$ as inputs. We use the features with resolution 1/8, 1/16 and 1/32 of the original image alternatively to update $\mathbf{Q}$ using a masked cross-attention (CA) operation [12], followed by a self-attention (SA) and finally a feed-forward network (FFN). We perform these sets of alternate operations L times inside the transformer decoder.

The final query outputs from the transformer decoder are mapped to a $K+1$ dimensional space for class predictions, where K denotes the number of classes and an extra +1 for the `no-object` predictions. To obtain the final masks, we decode the pixel features ($F_{1/4}$) at 1/4 resolution of the original image with the help of an `einsum` operation between Q and $F_{1/4}$. During inference, we follow the same post-processing technique as [12] to obtain the final panoptic, semantic, and instance segmentation predictions. We only keep predictions with scores above a threshold of 0.5, 0.8, and 0.8 during post-processing for panoptic segmentation on the ADE20K [15], Cityscapes [14] and COCO [34] datasets, respectively.

3.5. Losses

In addition to the contrastive loss on the queries, we calculate the standard classification CE-loss ($\mathcal{L}_{\text{cls}}$) over the class predictions. Following [12], we use a combination of binary cross-entropy ($\mathcal{L}_{\text{bce}}$) and dice loss ($\mathcal{L}_{\text{dice}}$) over the mask predictions. Therefore, our final loss function is a weighted sum of the four losses (Eq. (2)). We empirically set $\lambda_{\mathbf{Q}\leftrightarrow\mathbf{Q}_{\text{text}}} = 0.5$, $\lambda_{\text{cls}} = 2$, $\lambda_{\text{bce}} = 5$ and $\lambda_{\text{dice}} = 5$. To find the least cost assignment, we use bipartite matching [3, 13] between the set predictions and the ground truths. We set λ_{cls} as 0.1 for the `no-object` predictions [12].

$$\begin{aligned}\mathcal{L}_{\text{final}} = \lambda_{\mathbf{Q}\leftrightarrow\mathbf{Q}_{\text{text}}}\mathcal{L}_{\mathbf{Q}\leftrightarrow\mathbf{Q}_{\text{text}}} + \lambda_{\text{cls}}\mathcal{L}_{cls} \\ + \lambda_{\text{bce}}\mathcal{L}_{bce} + \lambda_{\text{dice}}\mathcal{L}_{dice}\end{aligned} \tag{2}$$

4. Experiments

We illustrate that OneFormer, when trained only once with our task-conditioned joint-training strategy, generalizes well to all three image segmentation tasks on three widely used datasets. Furthermore, we provide extensive ablations to demonstrate the significance of OneFormer's components. Due to space constraints, we provide implementation details in the appendix.

4.1. Datasets and Evaluation Metrics

Datasets. We experiment on three widely used datasets that support all three: semantic, instance, and panoptic segmentation tasks. **Cityscapes** [14] consists of a total 19 (11 "stuff" and 8 "thing") classes with 2,975 training, 500 validation and 1,525 test images. **ADE20K** [15] is another benchmark dataset with 150 (50 "stuff" and 100 "thing") classes among the 20,210 training and 2,000 validation images. **COCO** [34] has 133 (53 "stuff" and 80 "thing") classes with 118k training and 5,000 validation images.

Method	Backbone	#Params	#FLOPs	#Queries	Crop Size	Iters	PQ	AP	mIoU (s.s.)	mIoU (m.s.)
Individual Training										
CMT-DeepLab‡ [59]	MaX-S† [50]	—	—	—	1025×2049	60k	64.6	—	81.4	—
Axial-DeepLab-L‡ [51]	Axial ResNet-L† [51]	45M	687G	—	1025×2049	60k	63.9	35.8	81.0	81.5
Axial-DeepLab-XL‡ [51]	Axial ResNet-XL† [51]	173M	2447G	—	1025×2049	60k	64.4	36.7	80.6	81.1
Panoptic-DeepLab‡ [11]	SWideRNet† [8]	536M	10365G	—	1025×2049	60k	66.4	40.1	82.2	82.9
Mask2Former-Panoptic [12]	Swin-L† [38]	216M	514G	200	512×1024	90k	66.6	43.6	82.9	—
Mask2Former-Instance [12]	Swin-L† [38]	216M	507G	200	512×1024	90k	—	43.7	—	—
Mask2Former-Semantic [12]	Swin-L† [38]	215M	494G	100	512×1024	90k	—	—	83.3	84.3
kMaX-DeepLab‡ [60]	ConvNeXt-L† [39]	232M	1673G	256	1025×2049	60k	68.4	44.0	83.5	—
Joint Training										
OneFormer	Swin-L† [38]	219M	543G	250	512×1024	90k	**67.2**	**45.6**	83.0	**84.4**
OneFormer	ConvNeXt-L† [39]	220M	497G	250	512×1024	90k	**68.5**	**46.5**	83.0	84.0
OneFormer	ConvNeXt-XL† [39]	372M	775G	250	512×1024	90k	**68.4**	**46.7**	**83.6**	**84.6**
OneFormer	DiNAT-L† [21]	223M	450G	250	512×1024	90k	**67.6**	**45.6**	83.1	84.0

Table 2. **SOTA Comparison on Cityscapes val set.** †: backbones pretrained on ImageNet-22K; ‡: trained with batch size 32, *: hidden dimension 1024. OneFormer outperforms the individually trained Mask2Former [12] models. Mask2Former's performance with 250 queries is not listed, as its performance degrades with 250 queries. We compute FLOPs using the corresponding crop size.

Method	Backbone	#Params	#FLOPs	#Queries	Epochs	PQ	PQ^{Th}	PQ^{St}	AP	$AP^{instance}$	mIoU
Individual Training											
MaskFormer [13]	Swin-L† [38]	212M	792G	100	300	52.7	58.5	44.0	—	—	64.8
K-Net [61]	Swin-L† [38]	—	—	100	36	54.6	60.2	46.0	—	—	—
Panoptic SegFormer [33]	Swin-L† [38]	221M	816G	353	24	55.8	61.7	46.9	—	—	—
Mask2Former-Panoptic [12]	Swin-L† [38]	216M	875G	200	100	57.8	64.2	**48.1**	48.7	48.6	**67.4**
Mask2Former-Instance [12]	Swin-L† [38]	216M	868G	200	100	—	—	—	**49.1**	50.1	—
Mask2Former-Semantic‡ [12]	Swin-L† [38]	216M	891G	200	100	—	—	—	—	—	67.2
kMaX-DeepLab* [60]	ConvNeXt-L† [39]	232M	749G	128	81	**57.9**	64.0	**48.6**	—	—	—
kMaX-DeepLab* [60]	ConvNeXt-L† [39]	232M	749G	256	81	**58.0**	64.2	**48.6**	—	—	—
Joint Training											
OneFormer	Swin-L† [38]	219M	891G	150	100	**57.9**	**64.4**	48.0	**49.0**	48.9	**67.4**
OneFormer	DiNAT-L† [21]	223M	736G	150	100	**58.0**	**64.3**	**48.4**	**49.2**	49.2	**68.1**

Table 3. **SOTA Comparison on COCO val2017 set.** †: Imagenet-22k pretrained; ‡: retrained model result; *: trained with batch size 64. OneFormer outperforms the individually trained Mask2Former [12] on all metrics. We evaluate the AP score on instance ground truths derived from the panoptic annotations. Mask2Former's performance with 150 queries is not listed, as its performance degrades with 150 queries. We compute FLOPs using 100 validation COCO images (varying sizes). $AP^{instance}$ represents evaluation on the original instance annotations.

Evaluation Metrics. For all three image segmentation tasks, we report the **PQ** [29], **AP** [34], and **mIoU** [18] scores. Since we only have a single model for all three tasks, we use the value of the `task` token to decide the scores to consider. For *e.g.*, when `task` is `panoptic`, we report the **PQ** score and similarly we report **AP** and **mIoU** scores when `task` is `instance` and `semantic`, respectively.

4.2. Main Results

ADE20K. We compare OneFormer with the existing state-of-the-art pseudo-universal and specialized architectures on the ADE20K [15] val dataset in Tab. 1. With the standard Swin-L† backbone, OneFormer, while being trained only once, outperforms Mask2Former's [12] individually trained models on all three image segmentation tasks and sets a new state-of-the-art performance when compared with other methods using the same backbone.

Cityscapes. We compare OneFormer with the existing state-of-the-art pseudo-universal and specialized architectures on the Cityscapes [15] val dataset in Tab. 2. With Swin-L† backbone, OneFormer outperforms Mask2Former with a +**0.6**% and +**1.9**% improvement on the **PQ** and **AP** metrics, respectively. Additionally, with ConvNeXt-L† and ConvNeXt-XL† backbone, OneFormer sets a new state-of-the-art of 68.5% PQ and 46.7% AP, respectively.

COCO. We compare OneFormer with the existing state-of-the-art pseudo-universal and specialized architectures on the COCO [34] val2017 dataset in Tab. 3. With Swin-L† backbone, OneFormer performs on-par with the individually trained Mask2Former [12] with a +0.1% improvement in the PQ score. Due to the discrepancies between the panoptic and instance annotations in COCO [34], we evaluate the AP score using the instance ground truths derived from the panoptic annotations. We provide more information in the appendix. Following [12], we evaluate mIoU on semantic ground truths derived from panoptic annotations.

	PQ	AP	mIoU
OneFormer (ours)	**67.2**	**45.6**	**83.0**
– task-token ($\mathbf{Q}_{\text{task}}$)	66.5 (-0.7)	43.3 (-2.3)	82.9 (-0.1)
– learnable text context ($\mathbf{Q}_{\text{ctx}}$)	62.7 (-4.5)	45.0 (-0.6)	82.8 (-0.2)
– task-guided query init.	65.8 (-1.4)	44.5 (-1.1)	83.1 (+0.1)

Table 4. **Ablation on Components.** A task-conditioned architecture significantly improves the AP scores and using learnable text context improves the PQ score.

	PQ	AP	mIoU.	#param.
contrastive-loss (ours)	**67.2**	**45.6**	**83.0**	219M
query classification-loss	66.4 (-0.8)	44.7 (-0.9)	82.6 (-0.4)	219M
no contrastive-loss	58.8 (-8.4)	42.4 (-3.2)	82.5 (-0.5)	219M

Table 5. **Ablation on Loss.** The contrastive loss is essential for learning the inter-task distinctions during training.

	PQ	AP	mIoU
"a photo with a {CLS}" (ours)	**67.2**	**45.6**	**83.0**
"a photo with a {CLS} {TYPE}"	65.4 (-1.8)	44.5 (-1.1)	82.8 (-0.2)
"{CLS}"	66.6 (-0.6)	44.7 (-0.9)	82.5 (-0.5)

Table 6. **Ablation on Input Text Templates.** The template for the input text list entries is a critical factor for good performance. CLS represents the class name for the object and TYPE stands for the task-dependent object type.

4.3. Ablation Studies

We analyze OneFormer's components through a series of ablation studies. Unless stated otherwise, we ablate with Swin-L† OneFormer on the Cityscapes [14] dataset.

Task-Conditioned Architecture. We validate the importance of the task token ($\mathbf{Q}_{\text{task}}$), initializing the queries with repetitions of the task token (task-guided query init.) and the learnable text context ($\mathbf{Q}_{\text{ctx}}$) by removing each component one at a time in Tab. 4. Without the task token, we observe a significant drop in the AP score (−2.7%). Furthermore, using a learnable text context ($\mathbf{Q}_{\text{ctx}}$) leads to an improvement of +4.5% in the PQ score, proving its significance. Lastly, initializing the queries as repetitions of the task token (task-guided query init.) instead of using an all-zeros initialization [3] leads to an improvement of +1.4% in the PQ and +1.1% in the AP score, indicating the importance of task-conditioning the initialization of the queries.

Contrastive Query Loss. We report results without the query-text contrastive loss ($\mathcal{L}_{\mathbf{Q}\leftrightarrow\mathbf{Q}_{\text{text}}}$) in Tab. 5. We observe that the contrastive loss significantly benefits the PQ (+8.4%) and AP (+3.2%) scores. We also conduct experiments substituting our query-text contrastive loss with a classification loss ($\mathcal{L}_{\text{cls}}$) on the queries. $\mathcal{L}_{\text{cls}}$ can be regarded as a straightforward alternative for $\mathcal{L}_{\mathbf{Q}\leftrightarrow\mathbf{Q}_{\text{text}}}$ as the both provide supervision for the number of masks for each class present in the image. However, we observe significant drops on all the metrics (−0.8% PQ, −0.9% AP and −0.4% mIoU) using the classification loss instead of the contrastive loss. We attribute the drops to the inability of the classification loss to capture the inter-task differences effectively.

	PQ	AP	mIoU	#param.
OneFormer (ours)	**49.8**	**35.9**	**57.0**	219M
Mask2Former-Joint	48.7 (-1.1)	33.7 (-2.2)	56.2 (-0.8)	216M

Table 7. **Ablation on Joint Training.** Our OneFormer significantly beats the baseline's AP, PQ and mIoU scores. We report results with Swin-L† [38] backbone trained for 160k iterations on the ADE20K [15] dataset.

Task Token Input	PQ	PQ^{Th}	PQ^{St}	AP	mIoU
the task is panoptic	49.3	49.6	50.2	35.8	57.0
the task is instance	33.1	48.8	1.5	35.9	26.4
the task is semantic	40.4	35.5	50.2	25.3	57.0

Table 8. **Ablation on Task Token Input**. Our OneFormer is sensitive to the input task token value. We report results with Swin-L† OneFormer on the ADE20K [15] val set. The numbers in pink denote results on secondary task metrics.

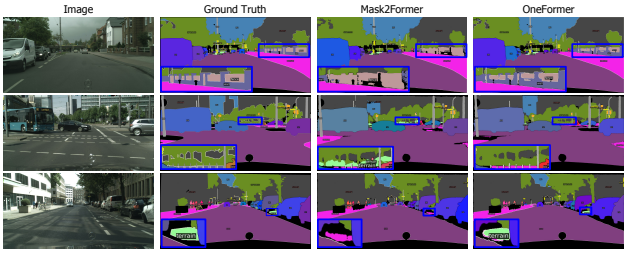

Figure 5. **Reduced Category Misclassifications.** Our OneFormer segments the regions (inside blue boxes) with similar classes more accurately than Mask2Former [12]. **Zoom in for best view.**

Input Text Template. We study the importance of the template choice for the entries in the text list ($\mathbf{T}_{\text{list}}$) in Tab. 6. We experiment with "a photo with a {CLS} {TYPE}" template for our text entries where CLS is the class name for the object mask and TYPE is the task-dependent class-type: "stuff" for amorphous masks (panoptic and semantic task) and "thing" for all distinct object masks. We also experiment with the identity template "{CLS}". Our choice of the template: "a photo with a {CLS}" gives a strong performance as a baseline. We believe more exploration in the text template space could help in improving the performance further.

Task Conditioned Joint Training. As a baseline for comparison, we train a Swin-L† Mask2Former-Joint with our joint training strategy, *i.e.*, uniformly sampling each task's GT on the ADE20K [15] dataset. We compare the Mask2Former-Joint baseline with our Swin-L† OneFormer in Tab. 7. We train both models for 160k iterations with a batch size of 16. Our OneFormer achieves a +1.1%, +2.2%, and +0.8% improvement on the PQ, AP and mIoU metrics, respectively, proving the importance of our architecture design for practical multi-task joint training.

Task Token Input. We demonstrate that our framework is sensitive to the task token input by setting the value of {task} during inference as panoptic, instance, or semantic in Tab. 8. We report results with our Swin-L† OneFormer trained on ADE20K [15] dataset. We observe a significant

drop in the PQ and mIoU metrics when `task` is instance compared to panoptic. Moreover, the PQ^{St} drops to 1.5%, and there is only a −0.8% drop on PQ^{Th} metric, proving that the network learns to focus majorly on the distinct "thing" instances when the `task` is instance. Similarly, there is a sizable drop in the PQ, PQ^{Th} and AP metrics for the semantic task with PQ^{St} staying the same, showing that our framework can segment out amorphous masks for "stuff" regions but does not predict different masks for "thing" objects. Therefore, OneFormer dynamically learns the inter-task distinctions which is critical for a train-once multi-task architecture. We include qualitative analysis on the task dynamic nature of OneFormer in the appendix.

Reduced Category Misclassifications. Our query-text contrastive loss helps OneFormer learn the inter-task distinctions and reduce the number of category misclassifications in the predictions. Mask2Former incorrectly predicts "wall" as "fence" in the first row, "vegetation" as "terrain", and "terrain" as "sidewalk". At the same time, our OneFormer produces more accurate predictions in regions (inside blue boxes) with similar classes, as shown in Fig. 5.

5. Conclusion

In this work, we present OneFormer, a new multi-task universal image segmentation framework with transformers and task-guided queries to unify semantic, instance, and panoptic segmentation with a single universal architecture, a single model, and training on a single dataset. Our jointly trained single OneFormer model outperforms the individually trained specialized Mask2Former models, the previous single-architecture state of the art, on all three segmentation tasks across major datasets. Consequently, OneFormer can cut training time, weight storage, and inference hosting requirements down to a third, making image segmentation more accessible. We believe OneFormer is a significant step towards making image segmentation more universal and accessible and will support further research in this direction by open-sourcing our codes and models.

Acknowledgments. We thank Intelligence Advanced Research Projects Activity (IARPA), University of Oregon, University of Illinois at Urbana-Champaign, and Picsart AI Research (PAIR) for their generous support that made this work possible.

References

[1] Anonymous. LMSeg: Language-guided multi-dataset segmentation. In *Submitted to The Eleventh International Conference on Learning Representations*, 2023. 3, 4

[2] Zhaowei Cai and Nuno Vasconcelos. Cascade R-CNN: Delving into high quality object detection. In *CVPR*, 2018. 3

[3] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *ECCV*, 2020. 2, 3, 5, 6, 8

[4] Joao Carreira, Rui Caseiro, Jorge Batista, and Cristian Sminchisescu. Semantic segmentation with second-order pooling. In *ECCV*, 2012. 3

[5] Kai Chen, Jiangmiao Pang, Jiaqi Wang, Yu Xiong, Xiaoxiao Li, Shuyang Sun, Wansen Feng, Ziwei Liu, Jianping Shi, Wanli Ouyang, Chen Change Loy, and Dahua Lin. Hybrid task cascade for instance segmentation. In *CVPR*, 2019. 3

[6] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L. Yuille. Semantic image segmentation with deep convolutional nets and fully connected crfs. In *ICLR*, 2015. 2

[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. In *TPAMI*, 2017. 2

[8] Liang-Chieh Chen, Huiyu Wang, and Siyuan Qiao. Scaling wide residual networks for panoptic segmentation. *arXiv*, 2020. 7, 15

[9] Zhe Chen, Yuchen Duan, Wenhai Wang, Junjun He, Tong Lu, Jifeng Dai, and Yu Qiao. Vision transformer adapter for dense predictions. *arXiv preprint arXiv:2205.08534*, 2022. 15

[10] Bowen Cheng, Liang-Chieh Chen, Yunchao Wei, Yukun Zhu, Zilong Huang, Jinjun Xiong, Thomas S Huang, Wen-Mei Hwu, and Honghui Shi. Spgnet: Semantic prediction guidance for scene parsing. In *CVPR*, 2019. 2

[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. 2, 7, 15

[12] Bowen Cheng, Ishan Misra, Alexander G. Schwing, Alexander Kirillov, and Rohit Girdhar. Masked-attention mask transformer for universal image segmentation. In *CVPR*, 2022. 1, 2, 3, 4, 5, 6, 7, 8, 12, 13, 14, 15, 16

[13] Bowen Cheng, Alexander G. Schwing, and Alexander Kirillov. Per-pixel classification is not all you need for semantic segmentation. In *NeurIPS*, 2021. 2, 3, 4, 5, 6, 7

[14] 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. 2, 6, 8, 12, 13, 14, 15

[15] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. Semantic understanding of scenes through the ade20k dataset. In *CVPR*, 2017. 2, 6, 7, 8, 12, 13, 14, 15

[16] Jifeng Dai, Kaiming He, and Jian Sun. Convolutional feature masking for joint object and stuff segmentation. In *CVPR*, 2015. 3

[17] 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. 12

[18] 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. 7

[19] 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. 12

[20] Bharath Hariharan, Pablo Arbeláez, Ross Girshick, and Jitendra Malik. Simultaneous detection and segmentation. In *ECCV*, 2014. 3

[21] Ali Hassani and Humphrey Shi. Dilated neighborhood attention transformer. *arXiv:2209.15001*, 2022. 2, 5, 6, 7, 15, 16

[22] Ali Hassani, Steven Walton, Jiachen Li, Shen Li, and Humphrey Shi. Neighborhood attention transformer. *arXiv:2204.07143*, 2022. 2

[23] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *ICCV*, 2017. 2, 3

[24] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016. 12, 13, 14

[25] Shihua Huang, Zhichao Lu, Ran Cheng, and Cheng He. FaPN: Feature-aligned pyramid network for dense image prediction. In *ICCV*, 2021. 15

[26] Zilong Huang, Xinggang Wang, Yunchao Wei, Lichao Huang, Humphrey Shi, Wenyu Liu, and Thomas S. Huang. Ccnet: Criss-cross attention for semantic segmentation. In *TPAMI*, 2020. 3

[27] Jitesh Jain, Anukriti Singh, Nikita Orlov, Zilong Huang, Jiachen Li, Steven Walton, and Humphrey Shi. Semask: Semantically masking transformer backbones for effective semantic segmentation. *arXiv*, 2021. 2, 3, 6, 15

[28] Alexander Kirillov, Ross Girshick, Kaiming He, and Piotr Dollár. Panoptic feature pyramid networks. In *CVPR*, 2019. 3

[29] Alexander Kirillov, Kaiming He, Ross Girshick, Carsten Rother, and Piotr Dollár. Panoptic segmentation. In *CVPR*, 2019. 2, 3, 4, 7

[30] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *NeurIPS*, 2012. 5

[31] Feng Li, Hao Zhang, Shilong Liu, Jian Guo, Lionel M Ni, and Lei Zhang. Dn-detr: Accelerate detr training by introducing query denoising. In *CVPR*, pages 13619–13627, 2022. 3

[32] Feng Li, Hao Zhang, Huaizhe xu, Shilong Liu, Lei Zhang, Lionel M. Ni, and Heung-Yeung Shum. Mask dino: Towards a unified transformer-based framework for object detection and segmentation. *arXiv*, 2022. 15, 16

[33] Zhiqi Li, Wenhai Wang, Enze Xie, Zhiding Yu, Anima Anandkumar, Jose M. Alvarez, Tong Lu, and Ping Luo. Panoptic segformer: Delving deeper into panoptic segmentation with transformers. In *CVPR*, 2022. 7, 16

[34] Tsung-Yi Lin, Michael Maire, Serge Belongie, Lubomir Bourdev, Ross Girshick, James Hays, Pietro Perona, Deva Ramanan, C. Lawrence Zitnick, and Piotr Dollár. Microsoft coco: Common objects in context. In *ECCV*, 2014. 2, 6, 7, 12, 13, 14, 16, 18, 19

[35] Huajun Liu, Fuqiang Liu, Xinyi Fan, and Dong Huang. Polarized self-attention: Towards high-quality pixel-wise regression. *arXiv*, 2021. 15

[36] Shilong Liu, Feng Li, Hao Zhang, Xiao Yang, Xianbiao Qi, Hang Su, Jun Zhu, and Lei Zhang. DAB-DETR: Dynamic anchor boxes are better queries for DETR. In *ICLR*, 2022. 3

[37] Ze Liu, Han Hu, Yutong Lin, Zhuliang Yao, Zhenda Xie, Yixuan Wei, Jia Ning, Yue Cao, Zheng Zhang, Li Dong, et al. Swin transformer v2: Scaling up capacity and resolution. *arXiv*, 2021. 6, 15

[38] 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. In *ICCV*, 2021. 2, 5, 6, 7, 8, 15, 16

[39] Zhuang Liu, Hanzi Mao, Chao-Yuan Wu, Christoph Feichtenhofer, Trevor Darrell, and Saining Xie. A convnet for the 2020s. In *CVPR*, 2022. 2, 5, 6, 7, 15, 16

[40] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *CVPR*, 2015. 2

[41] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *ICLR*, 2019. 12

[42] Rohit Mohan and Abhinav Valada. Efficientps: Efficient panoptic segmentation. *IJCV*, 2021. 15

[43] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. *arXiv*, 2021. 2, 3

[44] Robin Strudel, Ricardo Garcia, Ivan Laptev, and Cordelia Schmid. Segmenter: Transformer for semantic segmentation. In *ICCV*, 2021. 3

[45] Ke Sun, Yang Zhao, Borui Jiang, Tianheng Cheng, Bin Xiao, Dong Liu, Yadong Mu, Xinggang Wang, Wenyu Liu, and Jingdong Wang. High-resolution representations for labeling pixels and regions. *arXiv*, 2019. 15

[46] Mingxing Tan and Quoc V. Le. Efficientnet: Rethinking model scaling for convolutional neural networks. In *ICML*, 2019. 15

[47] Joseph Tighe, Marc Niethammer, and Svetlana Lazebnik. Scene parsing with object instances and occlusion ordering. In *CVPR*, 2014. 3

[48] Z. Tu, Xiangrong Chen, Alan Yuille, and Song Zhu. Image parsing: Unifying segmentation, detection, and recognition. In *IJCV*, 2005. 3

[49] 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. 3, 5

[50] 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. 3, 7

[51] Huiyu Wang, Yukun Zhu, Bradley Green, Hartwig Adam, Alan Yuille, and Liang-Chieh Chen. Axial-DeepLab: Stand-alone axial-attention for panoptic segmentation. In *ECCV*, 2020. 3, 7, 15

[52] Wenhui Wang, Hangbo Bao, Li Dong, Johan Bjorck, Zhiliang Peng, Qiang Liu, Kriti Aggarwal, Owais Khan Mohammed, Saksham Singhal, Subhojit Som, and Furu Wei. Image as a foreign language: Beit pretraining for all vision and vision-language tasks. *arXiv*, 2022. 6, 15

[53] Yixuan Wei, Han Hu, Zhenda Xie, Zheng Zhang, Yue Cao, Jianmin Bao, Dong Chen, and Baining Guo. Contrastive learning rivals masked image modeling in fine-tuning via feature distillation. *Tech Report*, 2022. 15

[54] Yuxin Wu, Alexander Kirillov, Francisco Massa, Wan-Yen Lo, and Ross Girshick. Detectron2. https://github.com/facebookresearch/detectron2, 2019. 12

[55] Tete Xiao, Yingcheng Liu, Bolei Zhou, Yuning Jiang, and Jian Sun. Unified perceptual parsing for scene understanding. In *ECCV*, 2018. 6, 15

[56] 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. 3

[57] Jiarui Xu, Shalini De Mello, Sifei Liu, Wonmin Byeon, Thomas Breuel, Jan Kautz, and Xiaolong Wang. Groupvit: Semantic segmentation emerges from text supervision. In *CVPR*, 2022. 2, 3, 5, 12

[58] Jian Yao, Sanja Fidler, and Raquel Urtasun. Describing the scene as a whole: Joint object detection, scene classification and semantic segmentation. In *CVPR*, 2012. 3

[59] Qihang Yu, Huiyu Wang, Dahun Kim, Siyuan Qiao, Maxwell Collins, Yukun Zhu, Hartwig Adam, Alan Yuille, and Liang-Chieh Chen. Cmt-deeplab: Clustering mask transformers for panoptic segmentation. In *CVPR*, 2022. 3, 7

[60] Qihang Yu, Huiyu Wang, Siyuan Qiao, Maxwell Collins, Yukun Zhu, Hartwig Adam, Alan Yuille, and Liang-Chieh Chen. k-means mask transformer. In *ECCV*, 2022. 2, 3, 6, 7, 14, 15, 16

[61] Wenwei Zhang, Jiangmiao Pang, Kai Chen, and Chen Change Loy. K-Net: Towards unified image segmentation. In *NeurIPS*, 2021. 1, 2, 3, 4, 6, 7

[62] Kaiyang Zhou, Jingkang Yang, Chen Change Loy, and Ziwei Liu. Conditional prompt learning for vision-language models. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022. 5

[63] Kaiyang Zhou, Jingkang Yang, Chen Change Loy, and Ziwei Liu. Learning to prompt for vision-language models. *International Journal of Computer Vision (IJCV)*, 2022. 5

[64] Xizhou Zhu, Weijie Su, Lewei Lu, Bin Li, Xiaogang Wang, and Jifeng Dai. Deformable detr: Deformable transformers for end-to-end object detection. *arXiv*, 2020. 3, 5, 12

Appendix

A. Implementation Details

We implement our framework using the Detectron2 [54] library.

Multi-Scale Feature Modeling. We adopt the settings from [12] for modeling the image pixel-level features. More specifically, we use 6 MSDeformAttn [64] inside our pixel decoder, applied to feature maps with resolutions 1/8, 1/16, and 1/32 of the original image. We use lateral connections and upsampling to aggregate the multi-scale features to a final 1/4 resolution scale. We map all the features to a hidden dimension of 256.

Unified Task-Conditioned Query Formulation. We initialize the $N-1$ queries as repetitions of task-token, $\mathbf{Q}_{\text{task}}$. Unless stated otherwise, we set $N = 250$ and $N_{\text{ctx}} = 16$. Our text tokenizer and text encoder are the same as [57]. We use a single linear layer to project the tokenized task input, followed by a layer-norm to obtain $\mathbf{Q}_{\text{task}}$.

Task-Dynamic Mask and Class Prediction Formation. Following [12], we set $L = 3$ inside the transformer decoder. Therefore, we have a total of $3L$ (9) stages inside our transformer decoder. We also calculate an auxiliary loss on each intermediate class and mask predictions after every transformer decoder stage [12].

Training Settings. We train our model with a batch size of 16. When training on ADE20K [15] and Cityscapes [14], we use the AdamW [41] optimizer with a base learning rate of 0.0001, poly learning rate decay and weight decay 0.1. We use a crop size of 512×512 and 512×1024 on ADE20K and Cityscapes, respectively. We train for 90k and 160k iterations on Cityscapes and ADE20K, respectively. For data augmentation, we use shortest edge resizing, fixed size cropping, and color jittering followed by a random horizontal flip.

When training on COCO [34], we use a step learning rate schedule along with the AdamW [41] optimizer, a base learning rate of 0.0001, 10 warmup iterations, and a weight decay of 0.05. We decay the learning rate at 0.9 and 0.95 fractions of the total number of training steps by a factor of 10. We train for a total of 100 epochs with LSJ augmentation [17, 19] with a random scale sampled from the range 0.1 to 2.0 followed by a fixed size crop to 1024×1024 resolution.

Evaluation Settings. We follow the same evaluation settings as Mask2Former [12]. Unless stated otherwise, we report results for the single-scale inference setting. Unlike the training stage, during evaluation, we use the ground-truth annotations from the respective `task` GT labels to calculate the metric scores instead of deriving the labels from the panoptic annotations. Additionally, we set the value of `task` in "the task is {`task`}" as panoptic, instance and semantic to obtain the corresponding task predictions.

#queries	PQ	AP	mIoU	#param.
100	51.3	41.9	60.8	47M
120	51.0	42.0	60.8	47M
150	**51.5**	**42.5**	**61.2**	47M
200	51.3	**42.5**	60.0	47M

Table I. **Ablation on Number of Queries.** We find $N = 150$ performs best on the COCO dataset.

N_{ctx}	PQ	AP	mIoU	#param.
0	41.7	**27.5**	46.5	47M
8	41.0	27.2	46.5	47M
16	**41.9**	27.3	**47.3**	47M
32	41.7	**27.5**	46.8	47M

Table II. **Ablation on number of learnable text context embeddings.** We find $N_{\text{ctx}} = 16$ performs best.

contrastive-loss weight	PQ	AP	mIoU
$\lambda_{\mathbf{Q}\leftrightarrow\mathbf{Q}_{\text{text}}} = 0.0$	51.1	42.1	60.2
$\lambda_{\mathbf{Q}\leftrightarrow\mathbf{Q}_{\text{text}}} = 0.5$	**51.5**	**42.5**	**61.2**
$\lambda_{\mathbf{Q}\leftrightarrow\mathbf{Q}_{\text{text}}} = 1.0$	50.7	42.0	60.5

Table III. **Ablation on Contrastive Loss' Weight.** We find $\lambda_{\mathbf{Q}\leftrightarrow\mathbf{Q}_{\text{text}}} = 0.5$ gives the best performance.

B. Additional Ablations

Ablation on Number of Queries. We study the effect of the different number of queries on the COCO dataset in Tab. I. We conduct experiments using the ResNet-50 (R50) [24] backbone and train for 50 epochs. We find that $N = 150$ performs the best.

Additionally, we tune the number of queries on the Swin-L† backbone separately. During our experiments, we found that $N = 250$ is the best setting with Swin-l† on ADE20K [15] and Cityscapes [14] datasets. On COCO [34], $N = 150$ gives the best performance with Swin-L†. We also noticed that with smaller backbones like R50 [24], $N = 150$ is the optimal setting on the ADE20K [15] dataset.

Ablation on Contrastive Loss' Weight. We run ablations on the weight for the contrastive loss' weight on the COCO dataset in Tab. III. We conduct our experiments using the ResNet-50 (R50) [24] backbone and train for 50 epochs. We find that $\lambda_{\mathbf{Q}\leftrightarrow\mathbf{Q}_{\text{text}}} = 0.5$ is the optimal weight setting.

Ablation on Number of Learnable Text Context Embeddings. We study the effect of different number of learnable text context embeddings on the ADE20K [15] dataset in Tab. II. We conduct our experiments using the ResNet-50 (R50) [24] backbone and train for 160k iterations. We find that $N_{\text{ctx}} = 16$ performs best.

C. Individual Training

In this section, we analyze our OneFormer's performance with individual training on the panoptic, instance, and semantic segmentation task. For this study, we con-

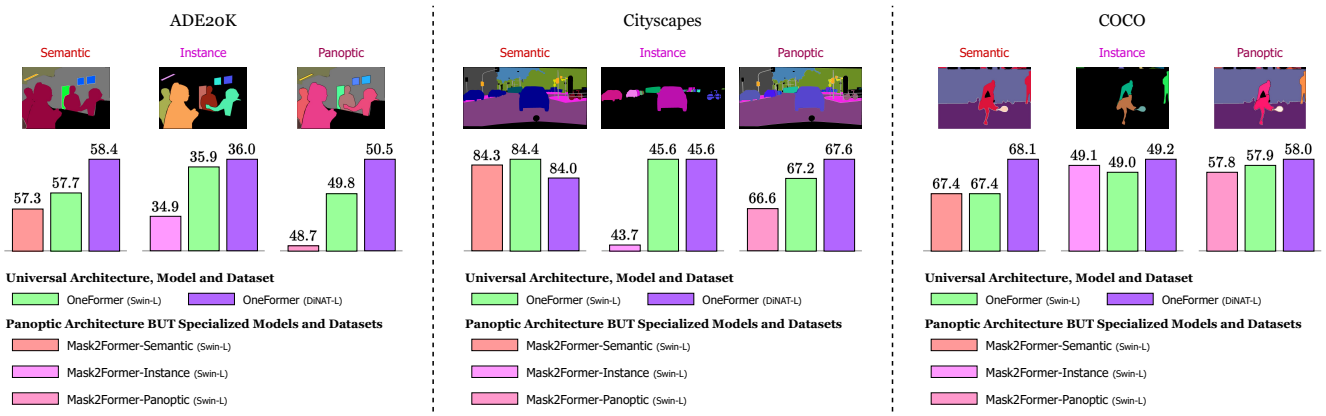

Figure I. **Comparison to Swin-L Mask2Former [12] across leaderboards.** Our single OneFormer model outperforms Mask2Former [12], the previous single architecture SOTA system on ADE20K val [15], Cityscapes val [14], and COCO val2017 [34] for all three segmentation tasks. With DiNAT-L OneFormer, we achieve even more improvements.

duct experiments with the ResNet-50 (R50) [24] backbone on the ADE20K [15] dataset. We train all models for 160k iterations with a batch size of 16.

As shown in Tab. IV, OneFormer outperforms Mask2Former [12] (the previous SOTA pseudo-universal image segmentation method) with every training strategy. Furthermore, with joint training, Mask2Former [12] suffers a significant drop in performance, and OneFormer achieves the highest PQ, AP and mIoU scores.

In order to train OneFormer on a single task, we set the value of `task` as that of the corresponding task in our task token input: "the task is {`task`}" for the samples during training. Therefore, under Panoptic Training, only panoptic ground truth labels will be used, and similarly, for Semantic and Instance Training, only semantic and instance ground truth labels shall be used, respectively. The joint training strategy remains the same as described in Sec 3.1 (main text) with uniform sampling for each task-specific ground truth label. Note that for training OneFormer, we derive all ground truth labels from the panoptic annotations.

D. Analysis on the Task-Dynamic Nature of OneFormer

We analyze OneFormer's ability to capture the inter-task differences by changing the value of {`task`} in the task token input: "the task is {`task`}" as panoptic, instance, or semantic, during inference. We report quantitative report results with our Swin-L† OneFormer trained on Cityscapes [14] dataset in Tab. V. When we set `task` as "instance", we observe that PQ$^{\text{St}}$ drops to 0.0%, and there is only a -0.2% drop on PQ$^{\text{Th}}$ metric as compared to the setting when `task` is panoptic. This observation proves that OneFormer learns to change its feed-forward output depending on the task dynamically. Similarly, there is a sizable drop in the PQ, PQ$^{\text{Th}}$ and AP metrics for the semantic task with PQ$^{\text{St}}$ improving by $+0.2\%$ showing that our framework can segment out amorphous masks for "stuff" regions but does not predict different masks for "thing" objects.

We further provide qualitative evidence in Fig. II. As demonstrated by the first example in Fig. II, the rider and bicycle regions are detected. However, the other "stuff" regions are misclassified in the semantic inference output when `task`="instance". Similarly, the people are detected in the second example, and the other "stuff" regions are misclassified. In further evidence, in both examples, the distinct "thing" objects are segmented into a single amorphous mask in the panoptic and instance inference outputs when `task`="semantic". Therefore, the differences in the qualitative results demonstrate OneFormer's ability to output task-dependent class and mask predictions, which our task token input can guide.

E. Comparison to SOTA Methods at System-Level for Image Segmentation

In this section, we compare OneFormer to other SOTA systems for panoptic, instance, and semantic segmentation tasks on the ADE20K val [15], Cityscapes val [14], and COCO val2017 [34] datasets. As shown in Fig. I, our single OneFormer model outperforms Mask2Former for the three image segmentation tasks on all three datasets. Note that we are comparing the same OneFormer models referenced in our main text to other systems without applying additional system-level training techniques or using additional data and huge backbones.

E.1. SOTA Systems on ADE20K val

As shown in Tab. VI, without using any extra training data, Swin-L OneFormer sets new state-of-the-art performance on instance segmentation with **37.8% AP**, and DiNat-L OneFormer sets new state-of-the-art performance on panoptic segmentation with **51.5% PQ** beating

training strategy	method	PQ	AP	mIoU
Panoptic Training	Mask2Former [12]	40.7	25.2	45.6
	OneFormer (ours)	**41.4** (+0.7)	**27.0** (+1.8)	**46.1** (+0.5)
Instance Training	Mask2Former [12]	—	26.4	—
	OneFormer (ours)	—	**26.7** (+0.3)	—
Semantic Training	Mask2Former [12]	—	—	47.2
	OneFormer (ours)	—	—	**47.3** (+0.1)
Joint Training	Mask2Former† [12]	40.8	25.7	46.6
	OneFormer (ours)	**41.9** (+1.1)	**27.3** (+1.6)	**47.3** (+0.7)

Table IV. **Comparison between Individual and Joint Training.** Unlike Mask2Former [12] which shows large variance in performance among the different training strategies, OneFormer performs fairly well under all training strategies and outperforms Mask2Former [12]. We train all models with R50 [24] backbone on the ADE20K [15] dataset for 160k iterations. † We retrain our own Mask2Former [12] using the joint training strategy.

Task Token Input	PQ	PQ^{Th}	PQ^{St}	AP	mIoU
the task is `panoptic`	67.2	61.0	71.7	45.3	83.0
the task is `instance`	25.6	60.8	0.0	45.6	6.3
the task is `semantic`	56.9	36.2	71.9	27.2	83.0

Table V. **Quantitative Analysis on Task Dynamic Nature of OneFormer**. Our OneFormer is sensitive to the input task token value. We report results with Swin-L† OneFormer on the Cityscapes [14] val set. The numbers in pink denote results on secondary `task` metrics.

the previous state-of-the-art Swin-L Mask2Former's [12] 34.9% AP and ConvNeXt-L KMaX-DeepLab's [60] 50.9% PQ, respectively. Furthermore, DiNAT-L OneFormer and ConvNeXt-L OneFormer achieve the new-state-of-the-art single-scale and multi-scale mIoU scores of **58.3%** and **58.8%**, respectively, compared to other systems that do not use extra data during training.

E.2. SOTA Systems on Cityscapes val

Without any extra data during training, our ConvNeXt-L OneFormer sets the new state-of-the-art performance on panoptic segmentation with **68.5% PQ** with single-scale inference. Similarly, ConvNeXt-XL OneFormer achieves a new state-of-the-art **46.7% AP** score with single-scale inference as shown in Tab. VII.

E.3. SOTA Systems on COCO val

Without using any extra training data, DiNAT-L OneFormer matches the previous state-of-the-art KMaX-DeepLab [60] with **58.0% PQ** score. Swin-L OneFormer achieves the best $\mathbf{PQ^{Th}}$ score of **64.4%**. For evaluating on the semantic segmentation task, we generate semantic GT annotations from the corresponding panoptic annotations. As shown in Tab. VIII, DiNAT-L OneFormer achieves an impressive **68.1% mIoU**.

While analyzing the COCO dataset, we found serious discrepancies between the GT panoptic and instance annotations. Therefore, for fair comparison, during evaluation, we generate the instance annotations from the panoptic annotations for calculating the AP scores as only use panoptic annotations during training. We provide more information about the discrepancies in Appendix F. DiNAT-L OneFormer achieves **49.2% AP** outperforming Mask2Former-Instance [12].

F. Analysis on Discrepancy between Instance and Panoptic Annotations in COCO

During our joint training, we derive the semantic and instance ground-truth labels from the corresponding panoptic annotations. Unlike, Cityscapes [14] and ADE20K [15] datasets, which combine the semantic and instance annotations to generate the corresponding panoptic annotations while preparing the data, COCO [34] has separate sets of panoptic and instance annotations. As expected, there are no discrepancies between the panoptic and instance annotations in the Cityscapes [14] and ADE20K [15] datasets. However, because COCO [34] has separately developed panoptic and instance annotations, we discover significant discrepancies in the COCO train2017 and val2017 [34] datasets as shown in Fig. III and Fig. IV, respectively.

In Fig. III, the instance annotations merge the "tie" object into the "person" object. In another example, instance annotations merge the "dog" and "boat" into a single instance, while the panoptic annotations segment the two instances correctly.

In Fig. IV, the instance annotations skip multiple "person" and "motorcycle" objects in different images, while the panoptic annotations include them all. In another example, instance annotations leave out a group of "person" object instances in the background, and panoptic annotations merge those instances into a single object mask.

These discrepancies are a significant barrier to developing and evaluating a unified image segmentation model. As demonstrated in Fig. III and Fig. IV, our predictions match the panoptic annotations much more than the instance annotations which is expected from our training strategy involv-

Method	Backbone	#Params	Crop Size	Extra Data	PQ	AP	mIoU (s.s.)	mIoU (m.s.)
Individual Training								
Mask2Former [12]	BEiT-3 [52]	1.9B	896×896	✓	—	—	62.0	62.8
UPerNet [55]	FD-SwinV2-G [53]	>3B	896×896	✓	—	—	—	61.4
Mask DINO [32]	Swin-L [38]	223M	896×896	✓	—	—	59.5	60.8
Mask2Former [12]	ViT-Adapter-L [9]	568M	896×896	✓	—	—	59.4	60.5
UPerNet [55]	SwinV2-G [37]	>3B	896×896	✓	—	—	59.3	59.9
UPerNet [55]	ViT-Adapter-L [9]	571M	640×640	✗	—	—	58.0	58.4
MSFaPN-Mask2Former [27]	SeMask Swin-L† [27]	—	640×640	✗	—	—	57.0	58.2
FaPN-Mask2Former [25]	Swin-L [38]	—	640×640	✗	—	—	56.4	57.7
SeMask Mask2Former [27]	SeMask Swin-L† [27]	—	640×640	✗	—	—	56.4	57.5
Mask2Former-Semantic [25]	Swin-L [38]	216M	640×640	✗	—	—	56.1	57.3
Mask2Former-Panoptic [12]	Swin-L [38]	216M	640×640	✗	48.1	34.2	54.5	—
kMaX-DeepLab [60]	ConvNeXt-L† [39]	232M	641×641	✗	48.7	—	54.8	—
Mask2Former-Instance [12]	Swin-L [38]	216M	640×640	✗	—	34.9	—	—
kMaX-DeepLab [60]	ConvNeXt-L† [39]	232M	1281×1281	✗	50.9	—	55.2	—
Joint Training								
OneFormer	Swin-L [38]	219M	640×640	✗	**49.8**	**35.9**	**57.0**	**57.7**
OneFormer	Swin-L [38]	219M	896×896	✗	**51.1**	**37.6**	**57.4**	**58.3**
OneFormer	Swin-L [38]	219M	1280×1280	✗	**51.4**	**37.8**	**57.0**	**57.7**
OneFormer	ConvNeXt-L [39]	220M	640×640	✗	**50.0**	**36.2**	**56.6**	**57.4**
OneFormer	ConvNeXt-XL [39]	372M	640×640	✗	**50.1**	**36.3**	**57.4**	**58.8**
OneFormer	DiNAT-L [21]	223M	640×640	✗	**50.5**	**36.0**	**58.3**	**58.4**
OneFormer	DiNAT-L [21]	223M	896×896	✗	**51.2**	**36.8**	**58.1**	**58.6**
OneFormer	DiNAT-L [21]	223M	1280×1280	✗	**51.5**	**37.1**	**58.2**	**58.7**

Table VI. **Comparison to methods on PwC Leaderboard on ADE20K val [15].** OneFormer achieves new-state-of-the-art performances on all three segmentation tasks when compared with methods **not using extra training data**.

Method	Backbone	#Params	Crop Size	Extra Data	MS (PQ & AP)	PQ	AP	mIoU (s.s.)	mIoU (m.s.)
Individual Training									
HRNetV2-OCR+PSA [35]	HRNetV2-W48 [45]	—	1024×2048	✓	✗	—	—	—	86.9
HRNetV2-OCR [35]	HRNetV2-W48 [45]	—	1024×2048	✓	✗	—	—	—	86.3
Mask2Former [12]	ViT-Adapter-L [9]	571M	896×896	✓	✗	—	—	84.9	85.8
Mask2Former [12]	SeMask Swin-L [27]	223M	512×1024	✗	✗	—	—	84.0	85.0
Mask2Former-Semantic [12]	Swin-L [38]	215M	512×1024	✗	✗	—	—	83.3	84.3
Panoptic-DeepLab [11]	SWideRNet [8]	—	1025×2049	✓	✓	69.6	46.8	—	85.3
Axial-DeepLab-XL [51]	Axial ResNet-XL [51]	173M	1025×2049	✓	✓	68.5	44.2	—	84.6
EfficientPS [42]	EfficientNet [46]	—	1025×2049	✓	✓	67.5	43.5	—	82.1
Panoptic-DeepLab [11]	SWideRNet [8]	—	1025×2049	✓	✗	68.5	42.8	84.6	85.3
Axial-DeepLab-XL [51]	Axial ResNet-XL [51]	173M	1025×2049	✓	✗	67.8	41.9	84.2	—
kMaX-DeepLab [60]	ConvNeXt-L [39]	232M	1025×2049	✗	✗	68.4	44.0	83.5	—
Panoptic-DeepLab [11]	SWideRNet [8]	—	1025×2049	✗	✗	66.4	40.1	82.2	82.9
Axial-DeepLab-XL [51]	Axial ResNet-XL [51]	173M	1025×2049	✗	✗	64.4	36.7	80.6	81.1
Mask2Former-Panoptic [12]	Swin-L [38]	216M	512×1024	✗	✗	66.6	43.6	82.9	—
Mask2Former-Instance [12]	Swin-L [38]	216M	512×1024	✗	✗	—	43.7	—	—
Joint Training									
OneFormer	Swin-L [38]	219M	512×1024	✗	✗	**67.2**	**45.6**	83.0	**84.4**
OneFormer	ConvNeXt-L [39]	220M	512×1024	✗	✗	**68.5**	**46.5**	83.0	84.0
OneFormer	ConvNeXt-XL [39]	372M	512×1024	✗	✗	**68.4**	**46.7**	**83.6**	**84.6**
OneFormer	DiNAT-L [21]	223M	512×1024	✗	✗	**67.6**	**45.6**	83.1	84.0

Table VII. **Comparison to SOTA systems on Cityscapes val [14].** OneFormer achieves new-state-of-the-art performances on the instance and panoptic segmentation tasks when compared with SOTA systems **using single-scale inference**.

Method	Backbone	#Params	Extra Data	PQ	PQ^{Th}	PQ^{St}	AP	$AP^{instance}$	mIoU
Individual Training									
Mask DINO [32]	Swin-L [38]	223M	✓	59.4	—	—	—	54.5	—
kMaX-DeepLab [60]	ConvNeXt-L [39]	232M	✓	58.1	64.3	48.8	—	—	—
kMaX-DeepLab [60]	ConvNeXt-L [39]	232M	✗	58.0	64.2	48.6	—	—	—
Mask2Former-Panoptic [12]	Swin-L [38]	216M	✗	57.8	64.2	48.1	48.7	48.6	67.4
Panoptic SegFormer [33]	Swin-L [38]	221M	✗	55.8	61.7	46.9	—	—	—
Mask2Former-Instance [12]	Swin-L [38]	216M	✗	—	—	—	**49.1**	50.1	—
Joint Training									
OneFormer	Swin-L [38]	219M	✗	**57.9**	**64.4**	48.0	**49.0**	48.9	**67.4**
OneFormer	DiNAT-L [21]	223M	✗	**58.0**	**64.3**	**48.4**	**49.2**	49.2	**68.1**

Table VIII. **Comparison to SOTA systems on COCO val2017 [34].** OneFormer achieves the best PQ^{Th} score among the SOTA systems trained without using any extra data. $AP^{instance}$ represents evaluation on the original instance annotations.

ing only panoptic annotations. Therefore, while comparing our Swin-L† OneFormer to other SOTA methods in Tab. 3 (main text), we evaluate the AP score on instance GTs derived from the panoptic annotations.

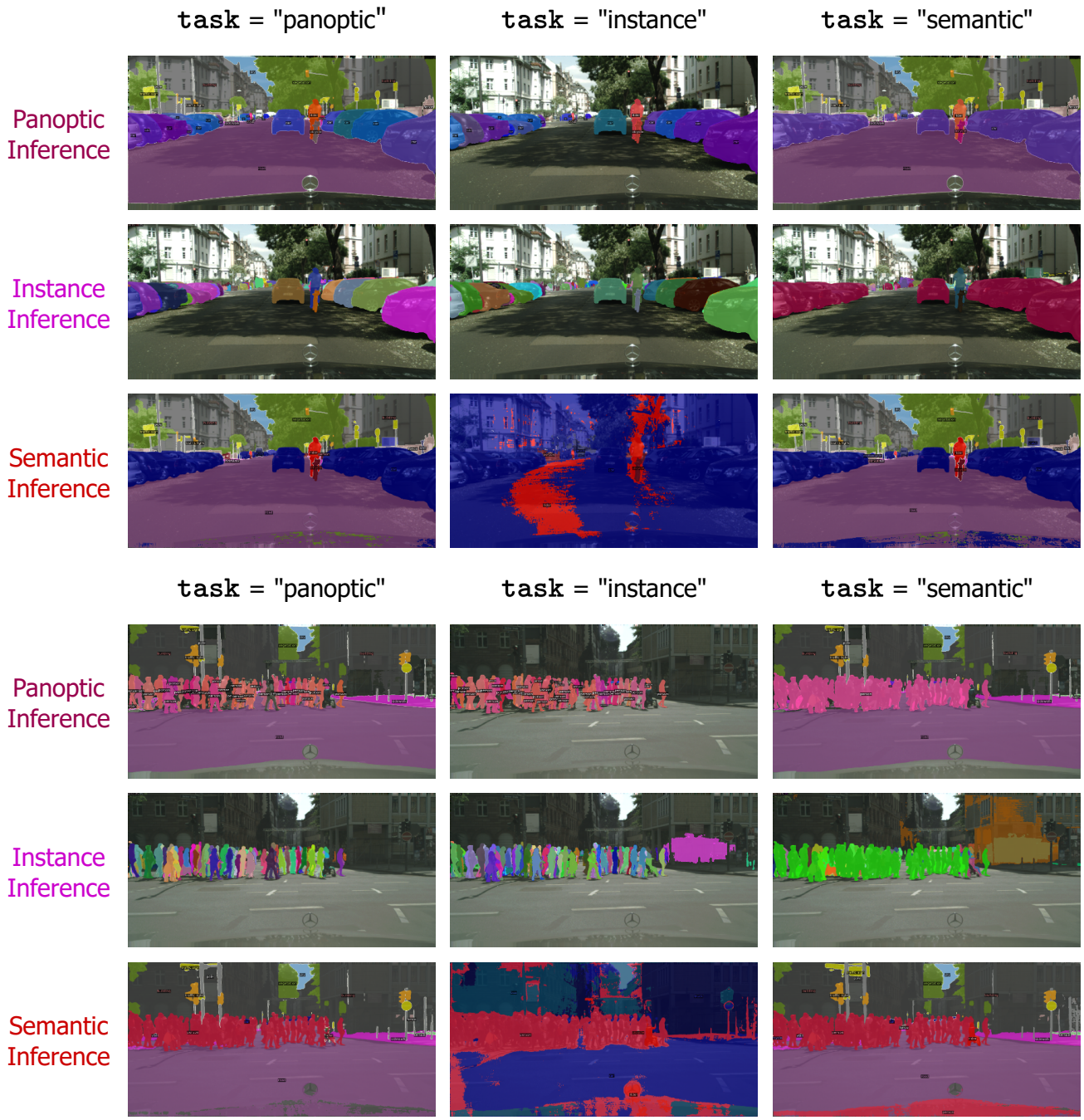

Figure II. **Qualitative Analysis on Task Dynamic Nature of OneFormer.** When `task` = "instance", the semantic inference outputs display fair detection of "thing" regions and misclassifications for the "stuff" regions. Similarly, when `task` = "semantic", the distinct object masks are grouped into a single amorphous mask, as expected by the formulation of the semantic segmentation task. **Zoom in for best view.**

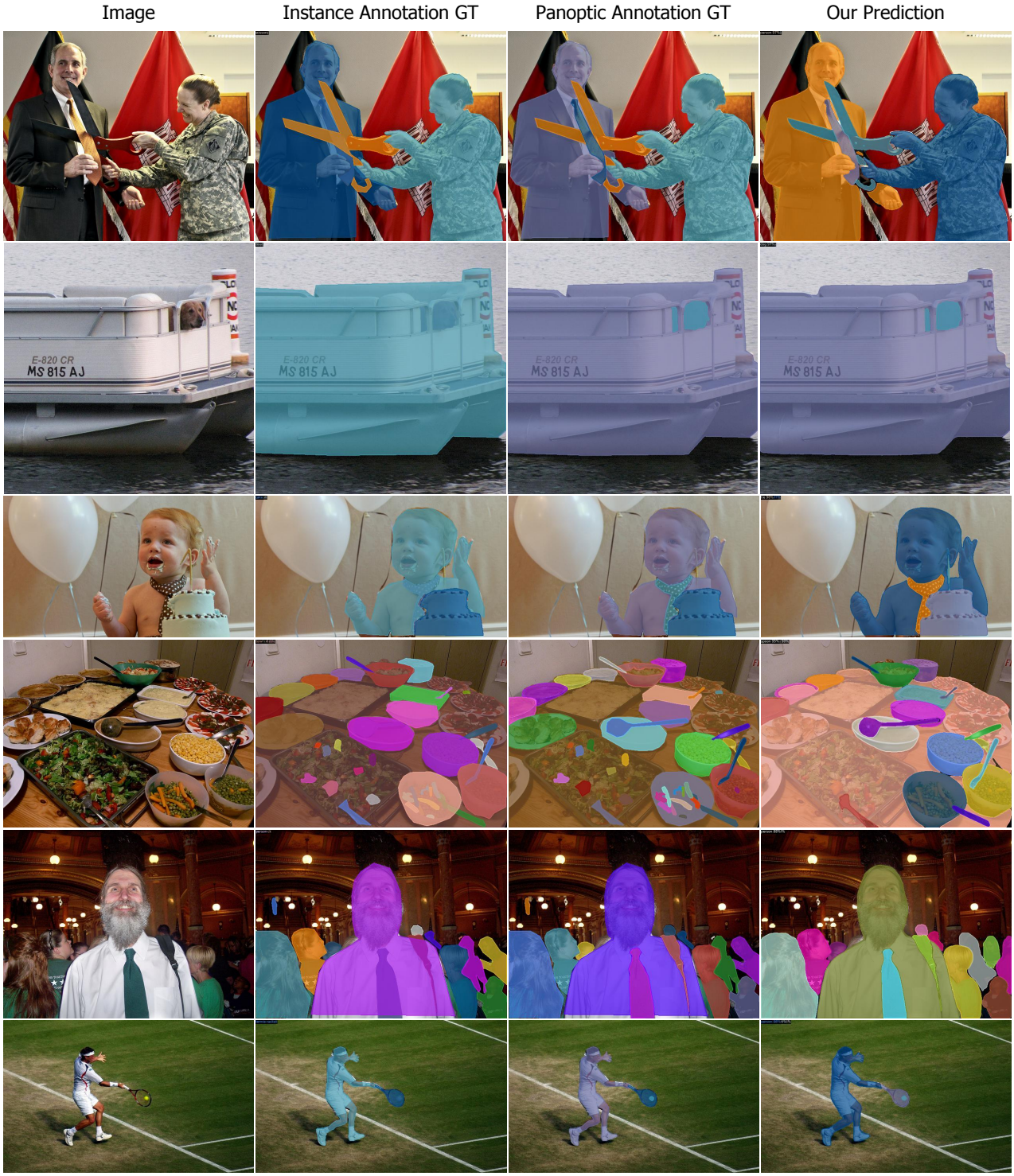

Figure III. **Discrepancy between instance and panoptic annotations in the COCO train2017 [34] dataset.** The "tie" instance is merged into the "person" instance in the instance annotations, whereas the panoptic annotations segment the two objects separately in the first, third, and fifth rows. Similarly, "dog" and "boat" are merged into a single instance in the instance annotations in the second row. The "bowl" and "spoon" are segmented as a single instance in instance annotations in the fourth row. Lastly, the 'tennis racket" and the small "sports ball" are segmented distinctly in panoptic annotations, unlike instance annotations in the last row. **Zoom in for best view.**

Figure IV. **Discrepancy between instance and panoptic annotations in the COCO val2017 [34] dataset.** The instance annotations skip multiple "person" and "motorcycle" objects in the first and fourth rows. The instance annotations leave out a group of "person" objects in the background, and panoptic annotations merge those objects into a single object mask in the second, third, fifth, and sixth rows. A similar case is observed with "bus" in the background in the last row. **Zoom in for best view.**