Appendix for 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

A. Implementation Details

We implement our framework using the Detectron2 [27] library.

Multi-Scale Feature Modeling. We adopt the settings from [4] for modeling the image pixel-level features. More specifically, we use 6 MSDeformAttn [31] 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 [29]. 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 [4], 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 [4].

Training Settings. We train our model with a batch size of 16. When training on ADE20K [6] and Cityscapes [5], we use the AdamW [20] 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 [15], we use a step learning rate schedule along with the AdamW [20] 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 [7, 8] 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 [4]. 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.

	PQ	AP	mIoU	#param.
conv. pos + sinusoidal feats	**49.8**	**35.9**	**57.0**	219M
sinusoidal pos + conv. feats	**49.8**	35.3	56.1	219M

Table IV. **Ablation on Positional Encodings.** We find that using conv. pos + sinusoidal feats give a better performance.

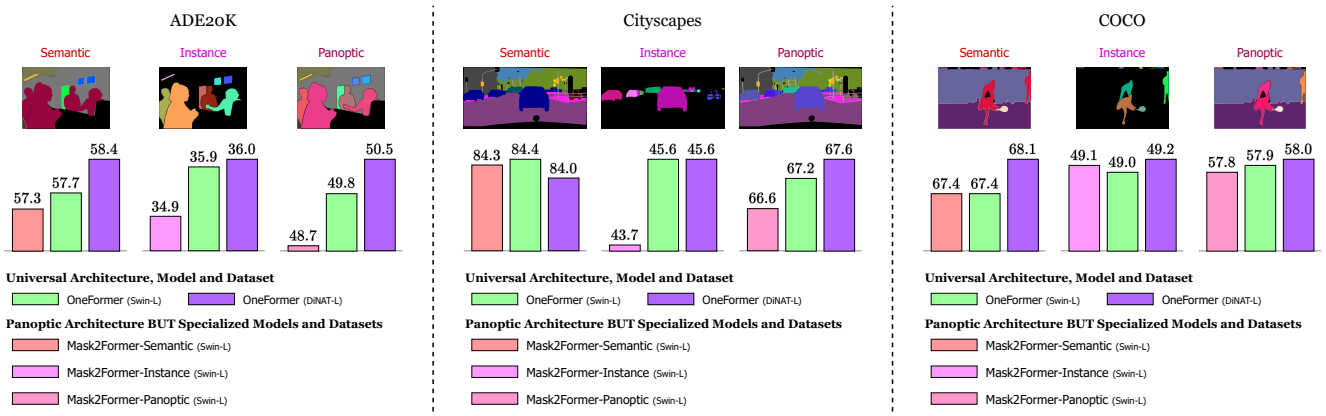

Figure I. **Comparison to Swin-L Mask2Former [4].** Our single OneFormer model outperforms Mask2Former [4], the previous single architecture SOTA system on ADE20K val [6], Cityscapes val [5], and COCO val2017 [15] for all three segmentation tasks. With DiNAT-L OneFormer, we achieve even more improvements.

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 [5] val set. The numbers in pink denote results on secondary `task` metrics.

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) [10] 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 [6] and Cityscapes [5] datasets. On COCO [15], $N = 150$ gives the best performance with Swin-L†. We also noticed that with smaller backbones like R50 [10], $N = 150$ is the optimal setting on the ADE20K [6] 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) [10] 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 [6] dataset in Tab. II. We conduct our experiments using the ResNet-50 (R50) [10] backbone and train for 160k iterations. We find that $N_{\text{ctx}} = 16$ performs best.

Ablation on Positional Encodings. We empirically find that convolutionally mapping the flattened 1/4-scale features (from the pixel decoder) to obtain the positional encodings and using sinusoidal image features as inputs to the class transformer perform better than using the standard sinusoidal positional encoding and convolutional features setting as shown in Tab. IV. We conduct experiments using the Swin-L† [18] backbone and train for 160k iterations.

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 conduct experiments with the ResNet-50 (R50) [10] backbone on the ADE20K [6] dataset. We train all models for 160k iterations with a batch size of 16.

As shown in Tab. VI, OneFormer outperforms Mask2Former [4] (the previous SOTA pseudo-universal image segmentation method) with every training strategy. Furthermore, with joint training, Mask2Former [4] 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.

training strategy	method	PQ	AP	mIoU
Panoptic Training	Mask2Former [4]	40.7	25.2	45.6
	OneFormer (ours)	**41.4** (+0.7)	**27.0** (+1.8)	**46.1** (+0.5)
Instance Training	Mask2Former [4]	—	26.4	—
	OneFormer (ours)	—	**26.7** (+0.3)	—
Semantic Training	Mask2Former [4]	—	—	47.2
	OneFormer (ours)	—	—	**47.3** (+0.1)
Joint Training	Mask2Former† [4]	40.8	25.7	46.6
	OneFormer (ours)	**41.9** (+1.1)	**27.3** (+1.6)	**47.3** (+0.7)

Table VI. **Comparison between Individual and Joint Training.** Unlike Mask2Former [4] which shows large variance in performance among the different training strategies, OneFormer performs fairly well under all training strategies and outperforms Mask2Former [4]. We train all models with R50 [10] backbone on the ADE20K [6] dataset for 160k iterations. † We retrain our own Mask2Former [4] using the joint training strategy.

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 [5] dataset in Tab. V. When we set `task` as "instance", we observe that PQ^{St} drops to 0.0%, and there is only a −0.2% drop on PQ^{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^{Th} and AP metrics for the semantic task with PQ^{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 observe that the mIoU/AP scores when {`task`} is panoptic are close to the best scores. We reason that panoptic segmentation is a superset of instance and semantic segmentation, due to which setting {`task`} as panoptic demonstrates good performance on all three metrics.

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 be guided by the task token and output task-dependent predictions.

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 [6], Cityscapes val [5], and COCO val2017 [15] 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. VII, 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 the previous state-of-the-art Swin-L Mask2Former's [4] 34.9% AP and ConvNeXt-L KMaX-DeepLab's [30] 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

Our ConvNeXt-L OneFormer sets the new state-of-the-art performance on panoptic segmentation with **70.1% PQ** with single-scale inference. Similarly, ConvNeXt-XL OneFormer achieves a new state-of-the-art **48.9% AP** score with single-scale inference as shown in Tab. IX.

E.3. SOTA Systems on COCO val

Without using any extra training data, DiNAT-L OneFormer matches the previous state-of-the-art KMaX-

Method	Backbone	#Params	Crop Size	Extra Data	PQ	AP	mIoU (s.s.)	mIoU (m.s.)
Individual Training								
Mask2Former [4]	BEiT-3 [25]	1.9B	896×896	✓	—	—	62.0	62.8
UPerNet [28]	FD-SwinV2-G [26]	>3B	896×896	✓	—	—	—	61.4
Mask DINO [13]	Swin-L [18]	223M	896×896	✓	—	—	59.5	60.8
Mask2Former [4]	ViT-Adapter-L [2]	568M	896×896	✓	—	—	59.4	60.5
UPerNet [28]	SwinV2-G [17]	>3B	896×896	✓	—	—	59.3	59.9
UPerNet [28]	ViT-Adapter-L [2]	571M	640×640	✗	—	—	58.0	58.4
MSFaPN-Mask2Former [12]	SeMask Swin-L† [12]	—	640×640	✗	—	—	57.0	58.2
FaPN-Mask2Former [11]	Swin-L [18]	—	640×640	✗	—	—	56.4	57.7
SeMask Mask2Former [12]	SeMask Swin-L† [12]	—	640×640	✗	—	—	56.4	57.5
Mask2Former-Semantic [11]	Swin-L [18]	216M	640×640	✗	—	—	56.1	57.3
Mask2Former-Panoptic [4]	Swin-L [18]	216M	640×640	✗	48.1	34.2	54.5	—
kMaX-DeepLab [30]	ConvNeXt-L† [19]	232M	641×641	✗	48.7	—	54.8	—
Mask2Former-Instance [4]	Swin-L [18]	216M	640×640	✗	—	34.9	—	—
kMaX-DeepLab [30]	ConvNeXt-L† [19]	232M	1281×1281	✗	50.9	—	55.2	—
Joint Training								
OneFormer	Swin-L [18]	219M	640×640	✗	**49.8**	**35.9**	**57.0**	**57.7**
OneFormer	Swin-L [18]	219M	896×896	✗	**51.1**	**37.6**	**57.4**	**58.3**
OneFormer	Swin-L [18]	219M	1280×1280	✗	**51.4**	**37.8**	**57.0**	**57.7**
OneFormer	ConvNeXt-L [19]	220M	640×640	✗	**50.0**	**36.2**	**56.6**	**57.4**
OneFormer	ConvNeXt-XL [19]	372M	640×640	✗	**50.1**	**36.3**	**57.4**	**58.8**
OneFormer	DiNAT-L [9]	223M	640×640	✗	**50.5**	**36.0**	**58.3**	**58.4**
OneFormer	DiNAT-L [9]	223M	896×896	✗	**51.2**	**36.8**	**58.1**	**58.6**
OneFormer	DiNAT-L [9]	223M	1280×1280	✗	**51.5**	**37.1**	**58.2**	**58.7**

Table VII. **Comparison to SOTA systems on ADE20K val [6].** OneFormer achieves new-state-of-the-art performances on all three segmentation tasks when compared with SOTA systems **not using extra training data**.

DeepLab [30] with **58.0% PQ** score. Swin-L OneFormer achieves the best **PQ$^{\text{Th}}$** 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 [4].

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 [5] and ADE20K [6] datasets, which combine the semantic and instance annotations to generate the corresponding panoptic annotations while preparing the data, COCO [15] has separate sets of panoptic and instance annotations. As expected, there are no discrepancies between the panoptic and instance annotations in the Cityscapes [5] and ADE20K [6] datasets. However, because COCO [15] has separately developed panoptic and instance annotations, we discover significant discrepancies in the COCO train2017 and val2017 [15] 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 involving only panoptic annotations. Therefore, while comparing

Method	Backbone	#Params	Extra Data	PQ	PQ^{Th}	PQ^{St}	AP	$AP^{instance}$	mIoU
Individual Training									
Mask DINO [13]	Swin-L [18]	223M	✓	59.4	—	—	—	54.5	—
kMaX-DeepLab [30]	ConvNeXt-L [19]	232M	✓	58.1	64.3	48.8	—	—	—
kMaX-DeepLab [30]	ConvNeXt-L [19]	232M	✗	58.0	64.2	48.6	—	—	—
Mask2Former-Panoptic [4]	Swin-L [18]	216M	✗	57.8	64.2	48.1	48.7	48.6	67.4
Panoptic SegFormer [14]	Swin-L [18]	221M	✗	55.8	61.7	46.9	—	—	—
Mask2Former-Instance [4]	Swin-L [18]	216M	✗	—	—	—	**49.1**	50.1	—
Joint Training									
OneFormer	Swin-L [18]	219M	✗	**57.9**	**64.4**	48.0	**49.0**	48.9	**67.4**
OneFormer	DiNAT-L [9]	223M	✗	**58.0**	**64.3**	**48.4**	**49.2**	49.2	**68.1**

Table VIII. **Comparison to SOTA systems on COCO val2017 [15].** 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.

Method	Backbone	#Params	Crop Size	Extra Data	MS (PQ & AP)	PQ	AP	mIoU (s.s.)	mIoU (m.s.)
Individual Training									
HRNetV2-OCR+PSA [16]	HRNetV2-W48 [22]	—	1024×2048	✓	✗	—	—	—	86.9
HRNetV2-OCR [16]	HRNetV2-W48 [22]	—	1024×2048	✓	✗	—	—	—	86.3
Mask2Former [4]	ViT-Adapter-L [2]	571M	896×896	✓	✗	—	—	84.9	85.8
Mask2Former [4]	SeMask Swin-L [12]	223M	512×1024	✗	✗	—	—	84.0	85.0
Mask2Former-Semantic [4]	Swin-L [18]	215M	512×1024	✗	✗	—	—	83.3	84.3
Panoptic-DeepLab [3]	SWideRNet [1]	—	1025×2049	✓	✓	69.6	46.8	—	85.3
Axial-DeepLab-XL [24]	Axial ResNet-XL [24]	173M	1025×2049	✓	✓	68.5	44.2	—	84.6
EfficientPS [21]	EfficientNet [23]	—	1025×2049	✓	✓	67.5	43.5	—	82.1
Panoptic-DeepLab [3]	SWideRNet [1]	—	1025×2049	✓	✗	68.5	42.8	84.6	85.3
Axial-DeepLab-XL [24]	Axial ResNet-XL [24]	173M	1025×2049	✓	✗	67.8	41.9	84.2	—
kMaX-DeepLab [30]	ConvNeXt-L [19]	232M	1025×2049	✗	✗	68.4	44.0	83.5	—
Panoptic-DeepLab [3]	SWideRNet [1]	—	1025×2049	✗	✗	66.4	40.1	82.2	82.9
Axial-DeepLab-XL [24]	Axial ResNet-XL [24]	173M	1025×2049	✗	✗	64.4	36.7	80.6	81.1
Mask2Former-Panoptic [4]	Swin-L [18]	216M	512×1024	✗	✗	66.6	43.6	82.9	—
Mask2Former-Instance [4]	Swin-L [18]	216M	512×1024	✗	✗	—	43.7	—	—
Joint Training									
OneFormer	Swin-L [18]	219M	512×1024	✗	✗	**67.2**	**45.6**	83.0	**84.4**
OneFormer	ConvNeXt-L [19]	220M	512×1024	✗	✗	**68.5**	**46.5**	83.0	84.0
OneFormer	ConvNeXt-XL [19]	372M	512×1024	✗	✗	**68.4**	**46.7**	**83.6**	**84.6**
OneFormer	ConvNeXt-L [19]	220M	512×1024	✓	✗	**70.1**	**48.7**	**84.6**	**85.2**
OneFormer	ConvNeXt-XL [19]	372M	512×1024	✓	✗	**69.7**	**48.9**	**84.5**	**85.8**
OneFormer	DiNAT-L [9]	223M	512×1024	✗	✗	**67.6**	**45.6**	83.1	84.0

Table IX. **Comparison to SOTA systems on Cityscapes val [5].** OneFormer achieves new-state-of-the-art performances on the instance and panoptic segmentation tasks when compared with SOTA systems **using single-scale inference**.

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.

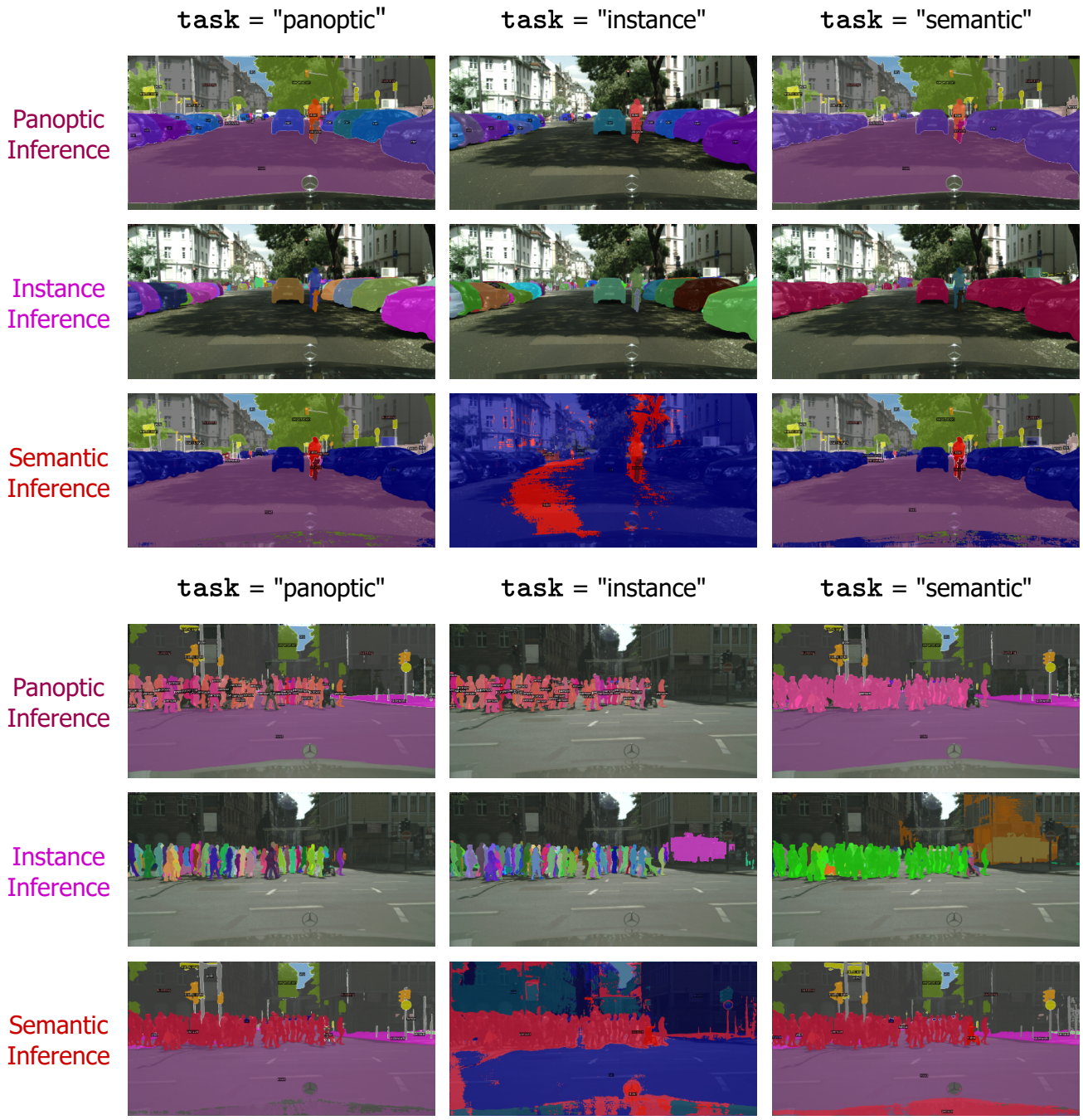

Figure II. **Qualitative Analysis on Task Dynamic Nature of OneFormer.** When `task` = "instance", the semantic inference outputs display fair detection of "thing" regions and misclassifications for the "stuff" regions. Similarly, when `task` = "semantic", the distinct object masks are grouped into a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

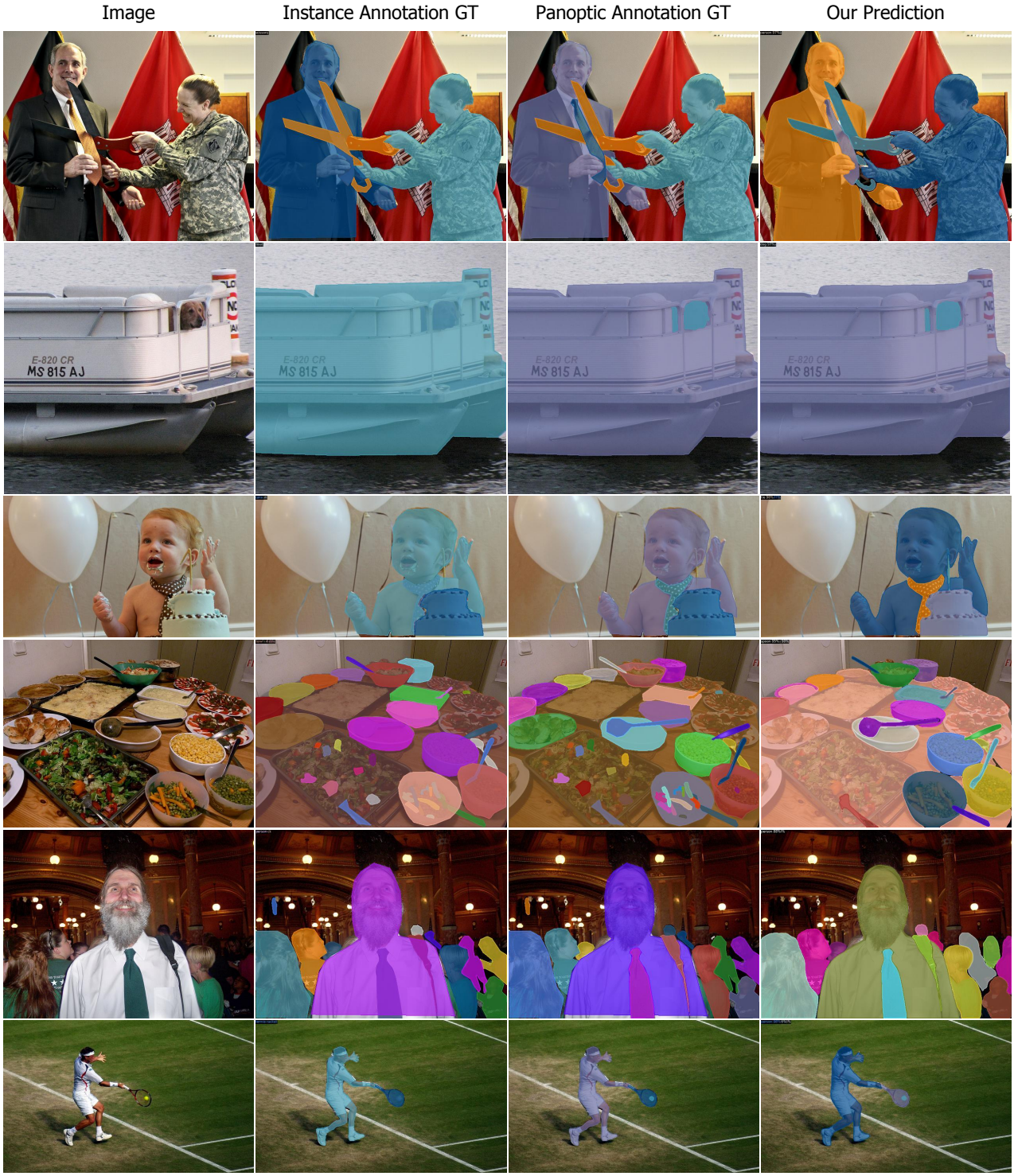

Figure III. **Discrepancy between instance and panoptic annotations in the COCO train2017 [15] dataset.** The "tie" instance is merged into the "person" instance in the instance annotations, whereas the panoptic annotations segment the two objects separately in the first, third, and fifth rows. Similarly, "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 [15] 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.**

References

[1] Liang-Chieh Chen, Huiyu Wang, and Siyuan Qiao. Scaling wide residual networks for panoptic segmentation. *arXiv*, 2020. 5

[2] 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. 4, 5

[3] 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. 5

[4] 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

[5] 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. 1, 2, 3, 4, 5

[6] 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. 1, 2, 3, 4

[7] 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. 1

[8] 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. 1

[9] Ali Hassani and Humphrey Shi. Dilated neighborhood attention transformer. *arXiv:2209.15001*, 2022. 4, 5

[10] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016. 2, 3

[11] Shihua Huang, Zhichao Lu, Ran Cheng, and Cheng He. FaPN: Feature-aligned pyramid network for dense image prediction. In *ICCV*, 2021. 4

[12] 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. 4, 5

[13] 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. 4, 5

[14] 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. 5

[15] 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. 1, 2, 3, 4, 5, 7, 8

[16] Huajun Liu, Fuqiang Liu, Xinyi Fan, and Dong Huang. Polarized self-attention: Towards high-quality pixel-wise regression. *arXiv*, 2021. 5

[17] 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. 4

[18] 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, 4, 5

[19] Zhuang Liu, Hanzi Mao, Chao-Yuan Wu, Christoph Feichtenhofer, Trevor Darrell, and Saining Xie. A convnet for the 2020s. In *CVPR*, 2022. 4, 5

[20] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *ICLR*, 2019. 1

[21] Rohit Mohan and Abhinav Valada. Efficientps: Efficient panoptic segmentation. *IJCV*, 2021. 5

[22] 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. 5

[23] Mingxing Tan and Quoc V. Le. Efficientnet: Rethinking model scaling for convolutional neural networks. In *ICML*, 2019. 5

[24] 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. 5

[25] 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. 4

[26] 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. 4

[27] Yuxin Wu, Alexander Kirillov, Francisco Massa, Wan-Yen Lo, and Ross Girshick. Detectron2. https://github.com/facebookresearch/detectron2, 2019. 1

[28] Tete Xiao, Yingcheng Liu, Bolei Zhou, Yuning Jiang, and Jian Sun. Unified perceptual parsing for scene understanding. In *ECCV*, 2018. 4

[29] 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. 1

[30] 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. 3, 4, 5

[31] 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. 1