FastInst: A Simple Query-Based Model for Real-Time Instance Segmentation Supplementary Material

Junjie He, Pengyu Li, Yifeng Geng, Xuansong Xie
DAMO Academy, Alibaba Group
hejunjie.hjj@alibaba-inc.com, lipengyu007@gmail.com, cangyu.gyf@alibaba-inc.com
xingtong.xxs@taobao.com

Appendix

We first provide several additional ablation studies for FastInst in Appendix A. Then, we demonstrate the performance of FastInst on a different dataset, *i.e.*, Cityscapes [3], in Appendix B, and show its unified segmentation ability in Appendix C. Finally, we visualize many predictions of FastInst on the COCO [6] validation set in Appendix D.

A. Additional ablation studies

A.1. Improvements on original Mask2Former

Considering that the proposed FastInst is developed based on Mask2Former, we investigate the improvements of three proposed key components, *i.e.*, instance activation-guided queries, dual-path update strategy, and ground truth mask-guided learning, on the original Mask2Former. The results are shown in Table I. Dual-path update strategy improves Mask2Former's efficiency the most, not only accelerating its inference speed but also reducing the model parameters dramatically. GT-mask guided learning also performs well on the original Mask2Former. The instance activation-guided queries improve the original Mask2Former little since the original Mask2Former already has enough Transformer decoder layers (*i.e.*, nine layers), and the learnable queries can also be decoded well. Note that IA-guided queries contribute to the 3-layer dual-path Transformer decoder.

A					✓	✓	✓	✓	✓
B		✓		✓		✓		✓	✓
C			✓	✓			✓	✓	✓
E									✓
Param.(M)	40.9	41.4	40.9	41.4	**33.5**	34.1	**33.5**	34.1	34.2
FPS	25.3	24.5	25.3	24.5	34.3	32.9	34.3	32.9	**35.5**
AP^{val}_{coco}	37.2	37.3	37.6	37.6	36.9	37.3	37.5	37.8	**37.9**

Table I. **Improvements on original Mask2Former.** A: 3-layer dual-path Transformer decoder (including little head change). B: IA-guided queries. C: GT mask-guided learning. E: learnable positional embeddings and auxiliary queries. The baseline is Mask2Former with PPM-FPN and 9 Transformer decoder layers. With A, B, C, and E, we achieve our FastInst-D3 model (R50 backbone).

A.2. Effect of location cost on deformable convolutional networks

We employ Hungarian loss [1] with a location cost during training to supervise the auxiliary classification head. The location cost restricts the matched pixels inside the object region, which reduces the matching space and, thus, accelerates training convergence. Table II shows that this location cost is also effective for FastInst with the backbone that employs deformable convolutional networks (DCNs) [8]. DCNs add 2D offsets to the regular grid sampling locations in the standard convolution and enable the pixels not located in the object region to have a chance of being activated for the segmentation. Despite this, the pixels outside the object region are not good IA-guided query candidates since they rely on precise offset predictions. Figure I visualizes the distributions of IA-guided queries with/without the location cost in a FastInst-D1 model with DCNs. The location cost helps produce more concentrated and higher-quality IA-guided queries.

A.3. Effect of local-maximum-first selection strategy

During prediction, we first select the pixel embeddings in E_4 with p_{i,k_i} that is the local maximum in the corresponding class plane and then pick the ones with the top foreground probabilities. Such a local-maximum-first selection strategy prevents the selected IA-guided queries from focusing on some salient objects. Table III demonstrates the effectiveness of the local-maximum-first selection strategy. It improves the performance, especially when the IA-guided query number is small. Figure II shows the influence of the local-maximum-first selection strategy on the selected IA-guided queries. Without the local-maximum-first selection strategy, two selected queries fall on the same salient object (*i.e.*, handbag), which hurts the recall of other instances.

	backbone	AP^{val}	AP_S	AP_M	AP_L	FPS
w/o location cost	R50-d-DCN	36.5	14.8	39.6	59.2	43.3
w/ location cost	R50-d-DCN	**38.1**(+1.6)	**16.2**	**41.5**	**60.6**	43.3

Table II. **Effect of location cost on DCNs.** The location cost is also important for FastInst with the backbone that employs DCNs. We conduct ablation studies on the FastInst-D1-640 model.

	N_a	AP^{val}	AP_S	AP_M	AP_L	FPS
w/o local-maximum-first	10	28.6	9.0	30.0	49.3	53.5
w/ local-maximum-first	10	**30.0**(+1.4)	**9.8**	**32.1**	**51.3**	52.9
w/o local-maximum-first	50	34.4	13.9	36.9	55.3	51.3
w/ local-maximum-first	50	**34.8**(+0.4)	**14.2**	**37.6**	**55.7**	51.0
w/o local-maximum-first	100	35.3	**14.6**	38.2	56.3	49.0
w/ local-maximum-first	100	**35.6**(+0.3)	14.3	**38.8**	**56.6**	48.8

Table III. **Effect of local-maximum-first selection strategy.** The local-maximum-first selection strategy is effective, especially when the IA-guided query number (*i.e.*, N_a) is small. We conduct ablation studies on FastInst-D1-640 with ResNet-50 backbone.

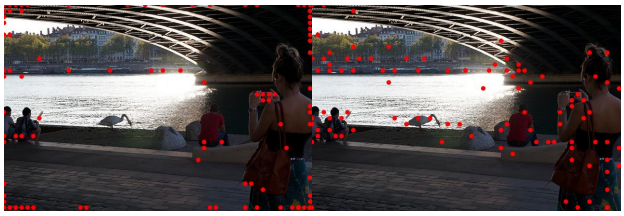

Figure I. **Effect of location cost on IA-guided queries with DCNs.** Left: not use the location cost. Right: use the location cost. We visualize the distributions of IA-guided queries in FastInst-D1-640 with a ResNet-50-d-DCN backbone. A few IA-guided queries are located in the padding area (for the size divisibility of 32) and not shown in the figure.

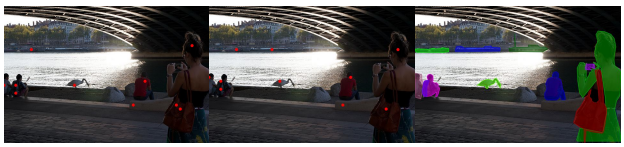

Figure II. **Effect of local-maximum-first selection strategy on IA-guided queries.** We visualize the IA-guided query distributions in FastInst-D1 without (left) and with (middle) the local-maximum-first selection strategy. Here $N_a = 10$. The right figure shows the ground truth.

A.4. Auxiliary query analysis

We visualize the cross-attention maps of three auxiliary learnable queries in the last Transformer decoder layer of FastInst-D3 (R50 backbone) in Figure III. As seen, auxiliary queries attend to general information such as edges (including background edge) and class-agnostic foreground objects.

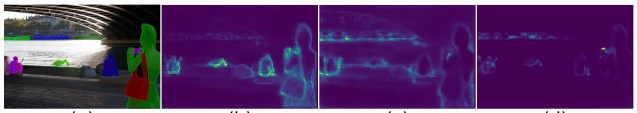

Figure III. **Visualization of cross-attention maps of auxiliary learnable queries.** (a) Ground truth. (b-d) Cross-attention maps of three (of eight) auxiliary learnable queries in the last Transformer decoder layer of FastInst-D3. The auxiliary queries in (b) and (c) attend to general edges, including background edges. The auxiliary query in (d) attends to class-agnostic foreground objects.

	backbone	AP^{val}	AP_{50}	FPS
Mask2Former†	R50	31.4	55.9	9.2
FastInst-D3 (ours)	R50	**35.5**(+4.1)	**59.0**	9.2

Table IV. **Instance segmentation results on Cityscapes `val`.** Mask2Former† denotes a light version of Mask2Former [2] that uses the same pixel decoder and training settings as FastInst.

	backbone	Cityscapes `val` AP	PQ	mIoU	#Param. (M)
Mask2Former†	R50	31.4	53.9	74.4	40.9
FastInst-D3	R50	**35.5**	**56.4**	**74.7**	**34.2**

Table V. **Panoptic (PQ) and semantic (mIoU) segmentation results on Cityscapes `val`.** Model and training settings are the same as instance segmentation (see Appendix B). FastInst performs better in instance-level segmentation than Mask2Former.

B. Additional datasets

B.1. Cityscapes

Cityscapes [3] is a high-resolution (1024×2048 pixels) street-view dataset that contains 2975 training, 500 validation, and 1525 testing images. We evaluate the performance of FastInst in terms of instance segmentation AP over eight semantic classes of the dataset.

Training settings. We use a batch size of 16 and train the model for 90K iterations. We set the initial learning rate as 0.0001 and drop it by multiplying 0.1 at 0.9 and 0.95 fractions of the total number of training steps. During training, we randomly resize the image to a shorter edge from 800 to 1024 pixels with a step of 32 pixels, followed by a crop size of 512×1024. During inference, we operate on the full image with a resolution of 1024×2048.

Results. Table IV shows the result of FastInst on Cityscapes `val` set. We also report the result of Mask2Former [2] that uses the same pixel decoder, *i.e.*, the pyramid pooling module [7]-based FPN (PPM-FPN) and the same training settings. FastInst outperforms the Mask2Former by a large margin (*i.e.*, 4.1 AP) with a similar speed, showing good efficiency in instance segmentation tasks.

C. Unified Segmentation

According to the practice of Mask2Former, FastInst can be easily transferred to other segmentation tasks. We show the panoptic and semantic segmentation results on Cityscapes in Table V.

D. Visualization

We visualize some predictions of the FastInst-D3 model with ResNet-50-d-DCN [4,5,8] backbone on the COCO [6] `val2017` set (40.1 AP) in Figure IV and Figure V. Figure VI shows two typical failure cases.

References

[1] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *ECCV*, 2020. 1

[2] Bowen Cheng, Ishan Misra, Alexander G. Schwing, Alexander Kirillov, and Rohit Girdhar. Masked-attention mask transformer for universal image segmentation. In *CVPR*, 2022. 2

[3] 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

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

[5] Tong He, Zhi Zhang, Hang Zhang, Zhongyue Zhang, Junyuan Xie, and Mu Li. Bag of tricks for image classification with convolutional neural networks. In *CVPR*, 2019. 3

[6] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft COCO: Common objects in context. In *ECCV*, 2014. 1, 3

[7] Hengshuang Zhao, Jianping Shi, Xiaojuan Qi, Xiaogang Wang, and Jiaya Jia. Pyramid scene parsing network. In *CVPR*, 2017. 2

[8] Xizhou Zhu, Han Hu, Stephen Lin, and Jifeng Dai. Deformable convnets v2: More deformable, better results. In *CVPR*, 2019. 1, 3

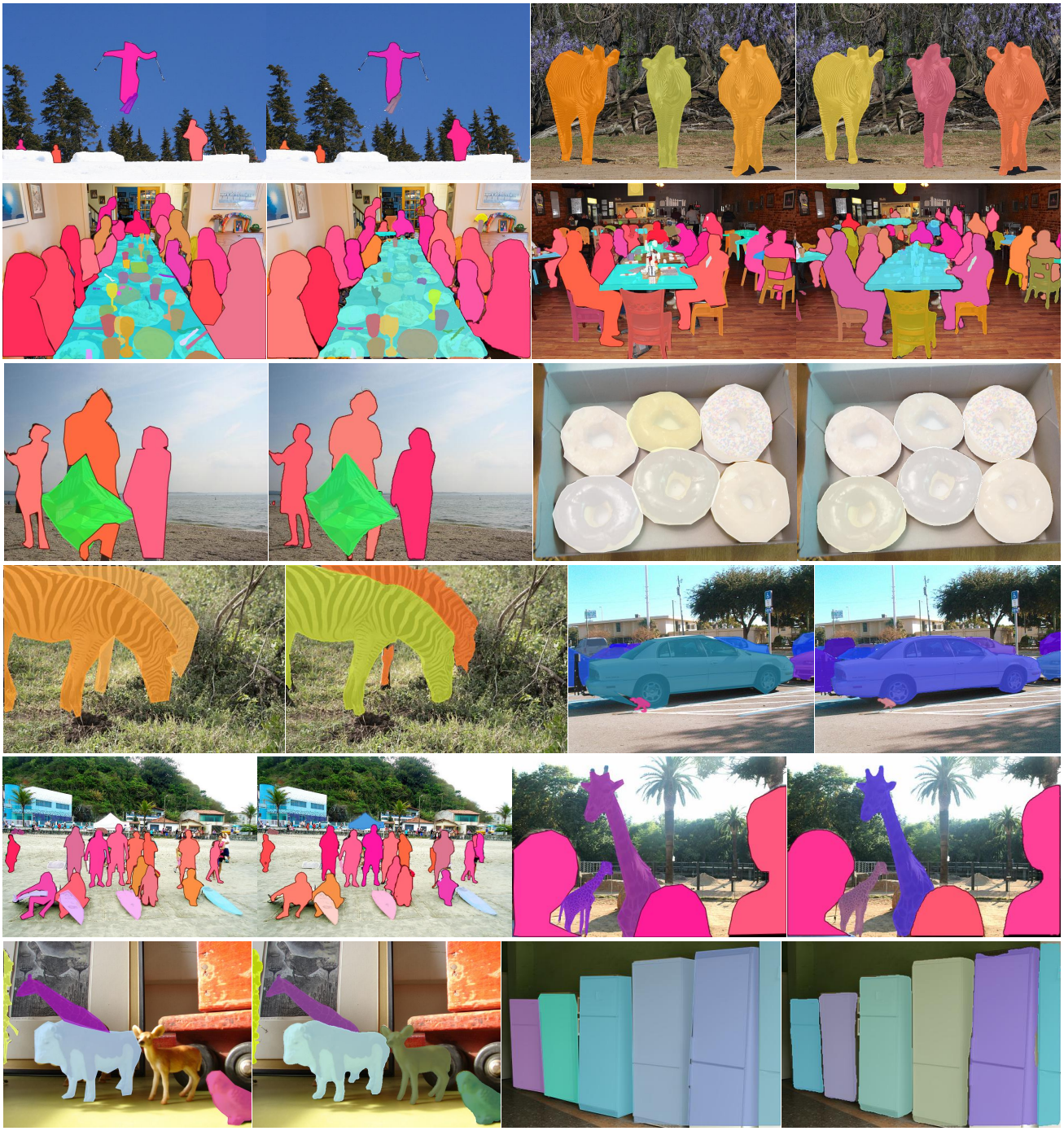

Figure IV. **Visualization of some predictions on the COCO dataset.** We use FastInst-D3 with a ResNet-50-d-DCN backbone that achieves 40.1 AP on the validation set with a speed of 32.5 FPS on a single V100 GPU. The first and third columns show the ground truth, and the second and fourth columns show the predictions. We set the confidence threshold to 0.5.

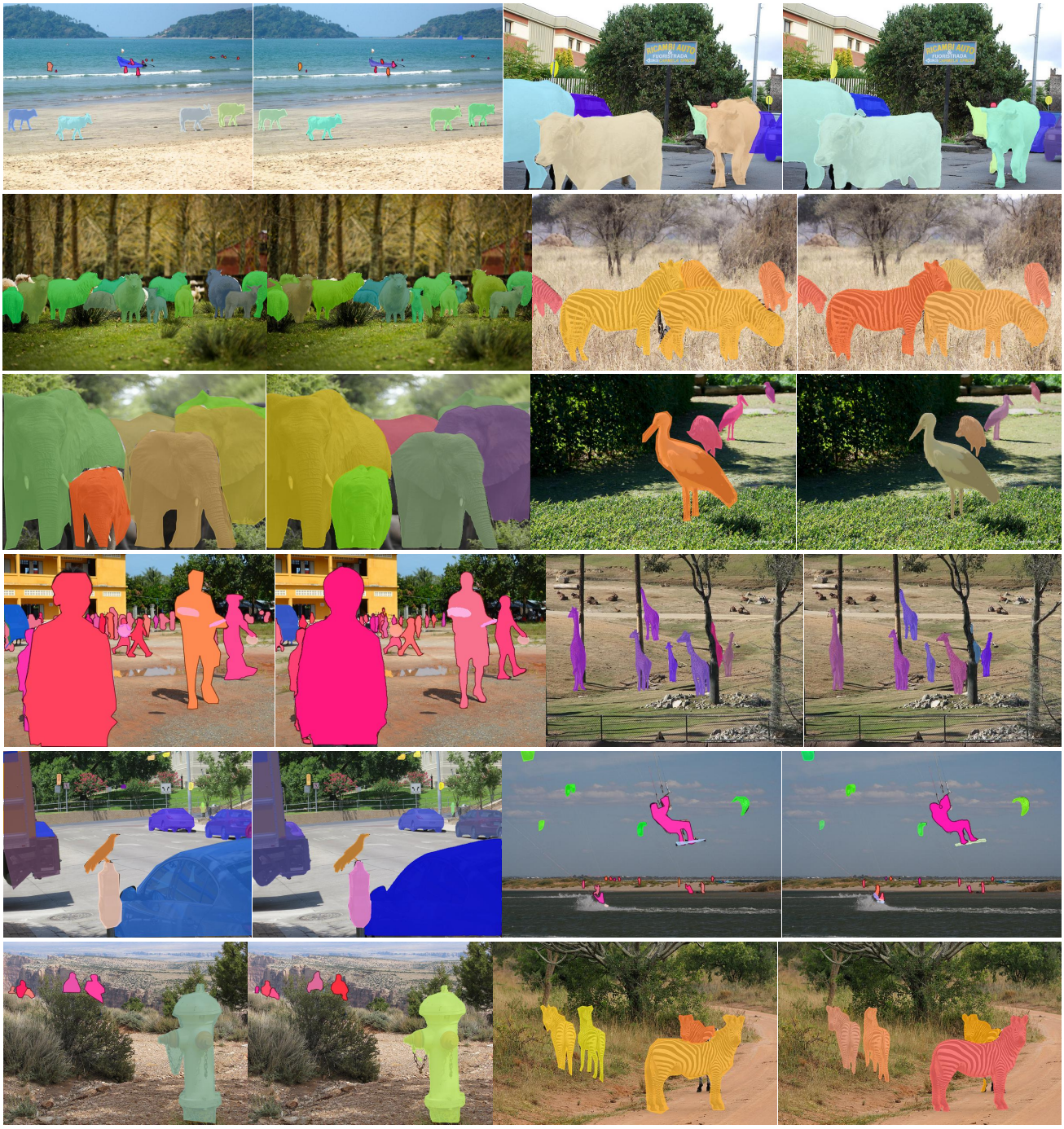

Figure V. **Visualization of another group of predictions on the COCO dataset.** We use FastInst-D3 with a ResNet-50-d-DCN backbone that achieves 40.1 AP on the validation set with a speed of 32.5 FPS on a single V100 GPU. The first and third columns show the ground truth, and the second and fourth columns show the predictions. We set the confidence threshold to 0.5.

Figure VI. **Visualization of two typical failure cases on the COCO dataset.** Left: duplicate predictions (*e.g.*, the person in the center). Right: over segmentation (*e.g.*, the cow in the upper right corner). Also, there are a few false positive and false negative predictions (see the left sample result). Here the first and third columns are the ground truth, and the second and fourth columns are the failure predictions. The confidence threshhold is set to 0.5, as in Figure IV and Figure V.