NTClick: Achieving Precise Interactive Segmentation With Noise-tolerant Clicks

Supplementary Material

1. Explicit Coarse Perception Network

1.1. Encoding of Click

We use Disk Map to encode *foreground click*, *background click* and *noise-tolerant click*, similar to RITM [7], FocalClick [1] and SimpleClick [4]. The encoded 3-channel tensor is then performed a patch embedding and added to the result of the patch embedding of the RGB image. This strategy of cooperating user interaction is proposed in SimpleClick [4].

1.2. Model Structure

The overall structure of **E**xplicit **C**oarse **P**erception (ECP) Network is shown in Fig. A, with the decoder structure illustrated in Fig. B. During decoder, the feature map output by the last block of the plain ViT is passed through four groups of convolutions, obtaining four feature maps with different sizes and dimensions. These feature maps are then upsampled to the same size and concatenated together, after which a simple MLP is used to obtain the final result. This approach of using a simple FPN-like decoder [3] for the ViT backbone was proposed in ViT-Det [2] and later optimized in SimpleClick [4].

As depicted, a significant feature of ECP is that it does not predict masks but instead predicts the FBU map. Consequently, we need to manually prepare the ground truth for the FBU map, which will be discussed in the next part.

1.3. Training Data Processing

To convert a ground truth mask to a ground truth FBU map in Fig. A, a simple morphology operation is required. Since the FBU map is essentially a coarse estimation, its ground truth is not uniquely defined. If the *foreground* region covers the simple parts of the object, the *background* region covers large continuous areas, and the *uncertain* region covers areas with fine structures, the FBU map can be considered correct. The three classes of clicks required for training and evaluation are also sampled from the corresponding regions of the FBU map. The core code segment for generating the FBU map during the training stage is as follows:

```
import cv2
import numpy as np
fg_mask = mask
bg_mask = 1 - mask
fg_mask = cv2.erode(fg_mask, kernel,
    iterations=1)
bg_mask = cv2.erode(bg_mask, kernel,
    iterations=1)
fbumap = np.ones_like(mask) * 128
fbumap[fg_mask == 1] = 255
fbumap[bg_mask == 1] = 0
```

kernel above is an elliptical structure created using OpenCV, with the size randomly sampled between 15 and 30.

However, to ensure the reproducibility of evaluation results, we fix the kernel size and the number of iterations for the FBU map during evaluation. The core code segment for generating the FBU map during the evaluation is as follows:

```
import cv2
import numpy as np

dilated = cv2.dilate(mask, kernel,
    iterations=dilation_iter)
eroded = cv2.erode(mask, kernel,
    iterations=erosion_iter)
fbumap = np.full(mask.shape, 128, dtype=
    np.uint8)
fbumap[eroded == 255] = 255
fbumap[dilated == 0] = 0
```

kernel and dilation_iter are 5 and 3, respectively. We deliberately use different codes and operators to generate FBU maps, demonstrating that our High Resolution Refine Network is capable of predicting stable results based on various FBU maps.

During training, we also use downsampled images and ground truth FBU map. This is because: 1) Large image sizes in the training set lead to slow I/O speed, while downsampling significantly improves training speed. 2) ECP is designed to perform a coarse perception at a relatively low resolution, so the loss of detail due to downsampling at this stage does not impact the performance.

2. High Resolution Refine Network

2.1. Model Structure

In the main paper, we discuss the backbone of **H**igh **R**esolution **R**efine Network (HRR). Here, we introduce the structure of HRR's decoder, as illustrated in Fig. C. 4 groups of convolutions extract low-level features with 4 different sizes from the RGB image. Then the feature map output by the final block of the backbone will be upsampled by a factor of 2 and then concatenated with the smallest low-level feature map. A simple convolutional layer is utilized to fuse backbone feature map and low-level feature map. This fusion process is repeated 4 times, during which the size of the feature map continuously increases. Finally, a group of convolutions is used to predict the final mask.

The grid attention used in HRR's backbone requires setting a interval of K, which we set to 8 in NTClick. The

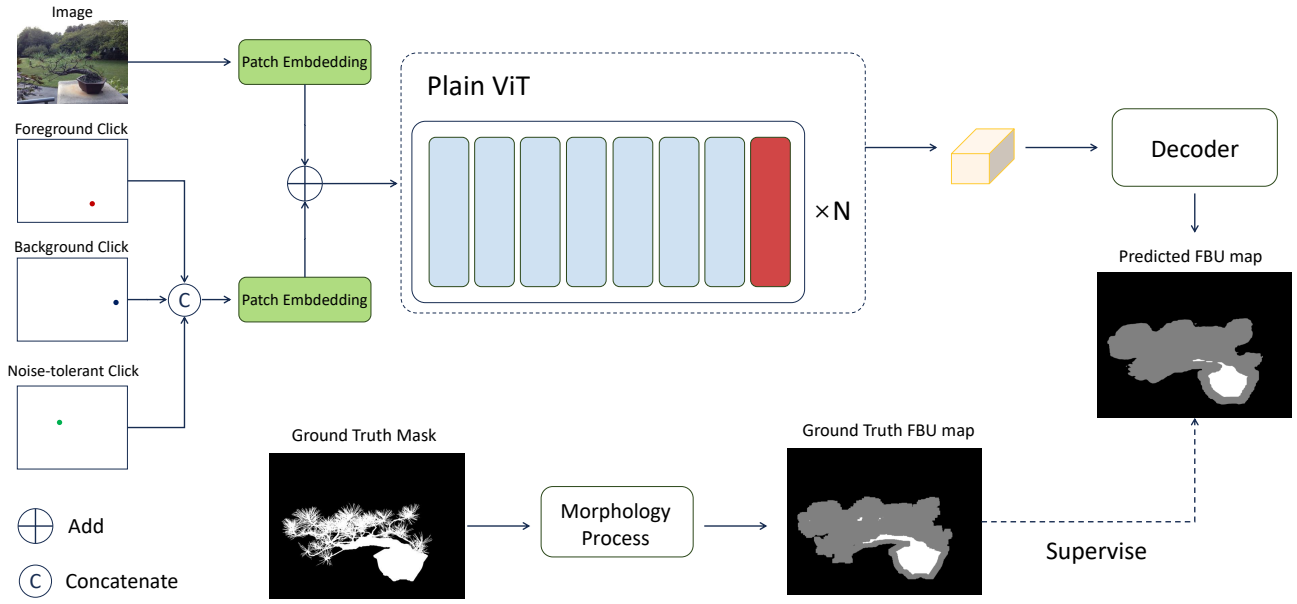

Figure A. Overall stucture of ECP.

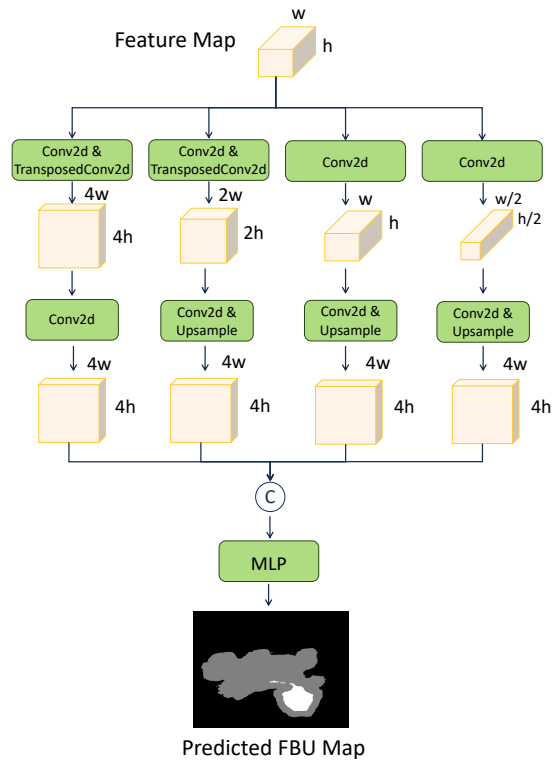

Figure B. Stucture of ECP's decoer.

choice of K impacts the resolution of the images fed into the backbone, as grid attention necessitates that the patches in both the height and width dimensions be divisible by K. Therefore, the image resolution needs to be padded to a multiple of patch size $\times K$. HRR's patch size is 16, so the height and width of input image needs to be a multiple of 128 (16×8).

2.2. Training Details

Training set. HRR was trained on the DIS5K [6] training set, given that HRR is essentially a refinement network and DIS5K features the most complex scenes and the highest annotation quality. Additionally, using less training data reduces the training burden.

Data augmentation. We only use Random Crop as the data augmentation strategy, with a crop size of 2048.

Loss function. HRR is constrained by both the L1 loss and the Gradient loss [8], two widely used loss functions in the field of image matting. Gradient loss is utilized to enhance the learning of edges in this task.

2.3. Inference

During the inference stage, HRR does not resize the image but instead pads it to the nearest multiple of 128 before feeding it into the backbone. This approach avoids the detail loss of RGB image due to downsampling. Essentially, HRR performs inference at a dynamic resolution. For example, an image of 1000×800 would be padded to 1024×896, and an image of 4000×3000 pixels would be padded to 4096×3072. Notably, for images with a long side exceeding 4096, we downsample them to a long side of 4096 while

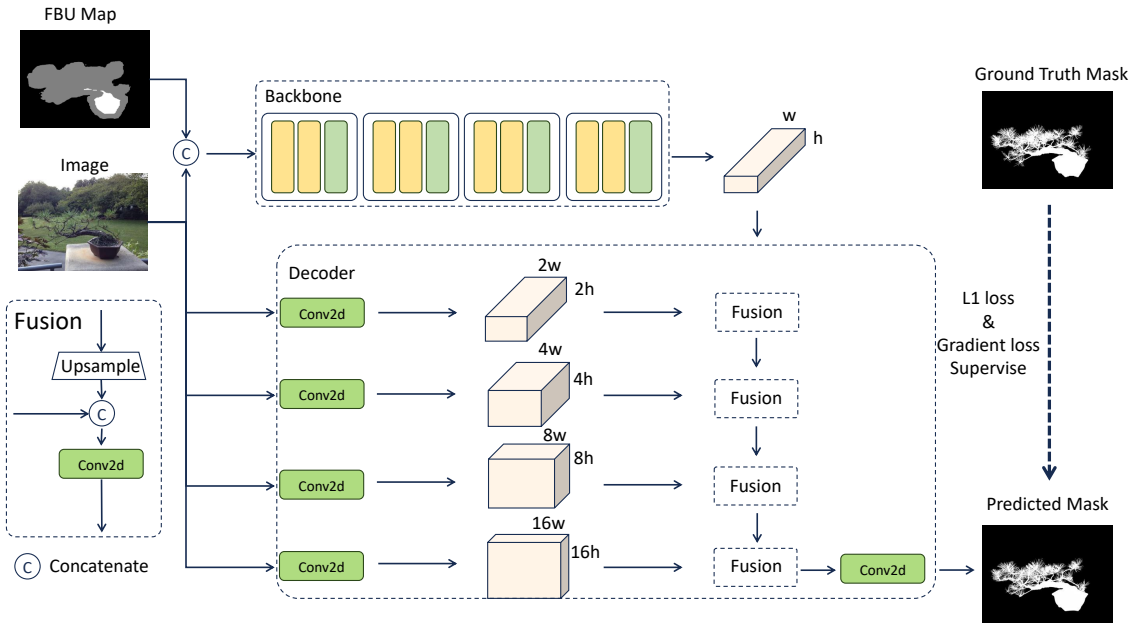

Figure C. Structure of HRR.

maintaining the aspect ratio. This is because we found that images with a resolution greater than 4096 do not typically contain details that require such high resolution to capture, and such details are not reflected in the annotations. Therefore, performing inference at resolutions higher than 4096 offers little benefit and instead adds unnecessary computational overhead.

3. Visualization

We provide more visualizations about interactions, FBU map and predictions in Fig. D,E,F.

4. Ablation Study

4.1. Impact of grid attention interval

Beyond the default interval of 8, we initially selected two additional values, 4 and 16, and retrained the model accordingly. When the interval was set to 4, the insufficient sparsity of global attention led to out-of-memory during inference at a resolution of 4096^2. therefore, We limits the resolution to 2048^2 to obtain results, which showed a slight decrease in accuracy due to the resolution reduction. When the interval was set to 16, the overly sparsification hinders grid attention from effectively establishing long-range dependencies, leading to a slight decrease in accuracy. Overall, an interval of 8 is a more suitable choice.

Interval = 4		Interval = 8		Interval = 16	
NoC@90↓	5-mIoU↑	NoC@90↓	5-mIoU↑	NoC@90↓	5-mIoU↑
7.89	89.09	7.23	89.23	7.32	89.04

Table A. Impact of grid attention interval.

5. Discussion

5.1. Comparison with other interaction forms.

Compared to *scribble-based methods*, our method maintains interaction sparsity even when dealing with objects that contain numerous fine-grained regions. In contrast, scribble-based methods require extensive scribbling, which can be inefficient. Compared to *bbox-based methods*, our method enables continuous refinement, a capability that box-based methods inherently lack. This ability is crucial for precise segmentation tasks. According to existing work [5], it is difficult for *bbox-based methods* to achieve higher segmentation accuracy than *click-based methods*.

5.2. Parameters of morphology process.

Smaller dilation-erosion kernels and fewer iterations result in more narrow uncertain regions in the FBU map, whereas larger kernels and more iterations produce more broad uncertain regions. From the perspective of segmentation accuracy, the fewer uncertain pixels in the FBU map, the higher the quality of the final mask refined by HRR. However, from the perspective of interaction efficiency, we actually prefer

that the model does not learn overly small *uncertain* regions. This is because there is a correspondence between the *uncertain* region and the *noise-tolerant click*. If the *uncertain* region is too narrow, it becomes more challenging to accurately locate the *noise-tolerant click*, which contradicts the original motivation of NTClick. Therefore, we chose a balanced width for the *uncertain* region.

References

[1] Xi Chen, Zhiyan Zhao, Yilei Zhang, Manni Duan, Donglian Qi, and Hengshuang Zhao. Focalclick: Towards practical interactive image segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1300–1309, 2022. 1

[2] Yanghao Li, Hanzi Mao, Ross Girshick, and Kaiming He. Exploring plain vision transformer backbones for object detection. In *European conference on computer vision*, pages 280–296. Springer, 2022. 1

[3] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2117–2125, 2017. 1

[4] Qin Liu, Zhenlin Xu, Gedas Bertasius, and Marc Niethammer. Simpleclick: Interactive image segmentation with simple vision transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 22290–22300, 2023. 1

[5] Qin Liu, Jaemin Cho, Mohit Bansal, and Marc Niethammer. Rethinking interactive image segmentation with low latency high quality and diverse prompts. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3773–3782, 2024. 3

[6] Xuebin Qin, Hang Dai, Xiaobin Hu, Deng-Ping Fan, Ling Shao, and Luc Van Gool. Highly accurate dichotomous image segmentation. In *European Conference on Computer Vision*, pages 38–56. Springer, 2022. 2

[7] Konstantin Sofiiuk, Ilya A Petrov, and Anton Konushin. Reviving iterative training with mask guidance for interactive segmentation. In *2022 IEEE International Conference on Image Processing (ICIP)*, pages 3141–3145. IEEE, 2022. 1

[8] Jingwei Tang, Yagiz Aksoy, Cengiz Oztireli, Markus Gross, and Tunc Ozan Aydin. Learning-based sampling for natural image matting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3055–3063, 2019. 2

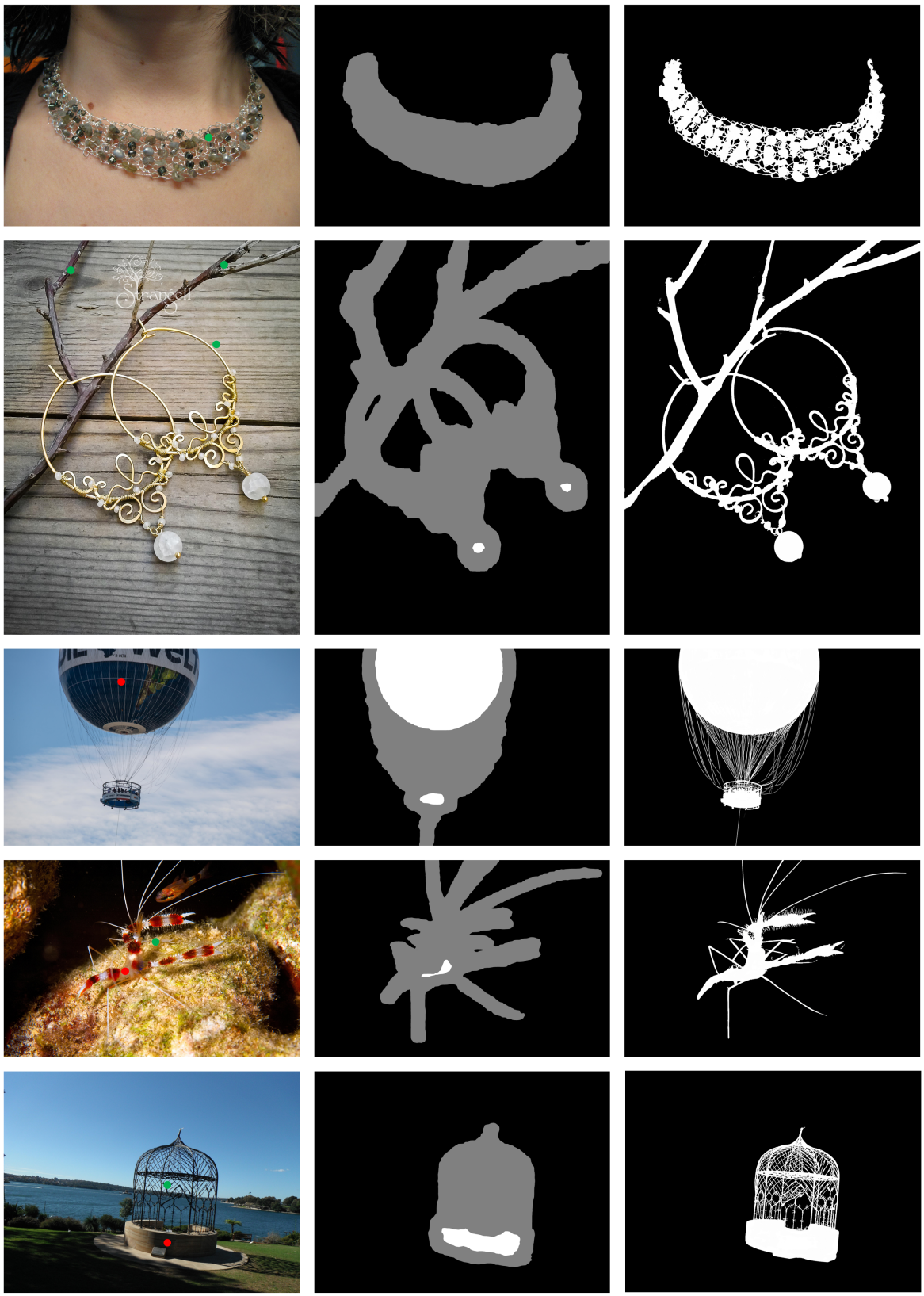

Figure D. •, and • refer to *foreground click* and *background click*, • refers to *noise-tolerant click*.

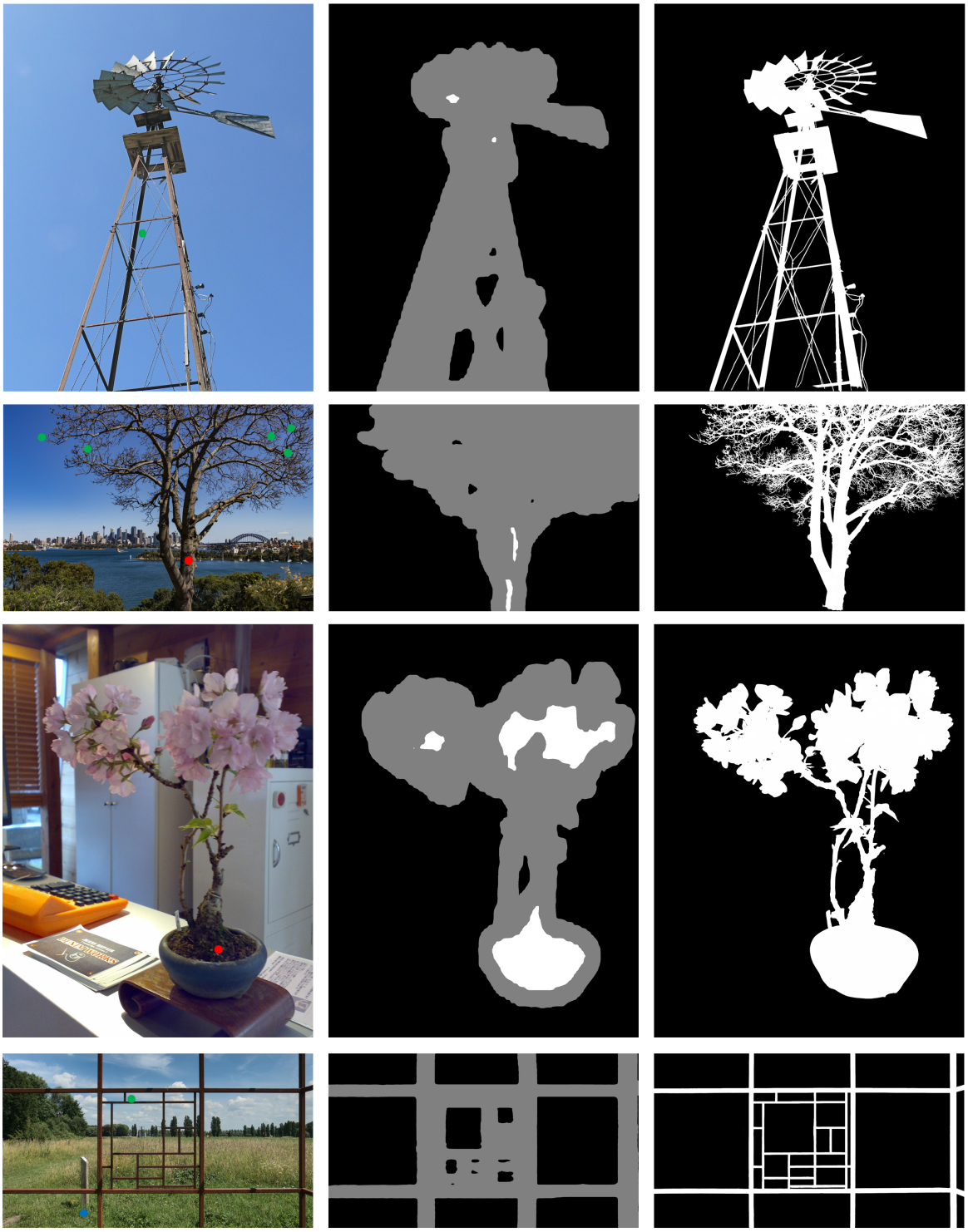

Figure E. •, and • refer to *foreground click* and *background click*, • refers to *noise-tolerant click*.

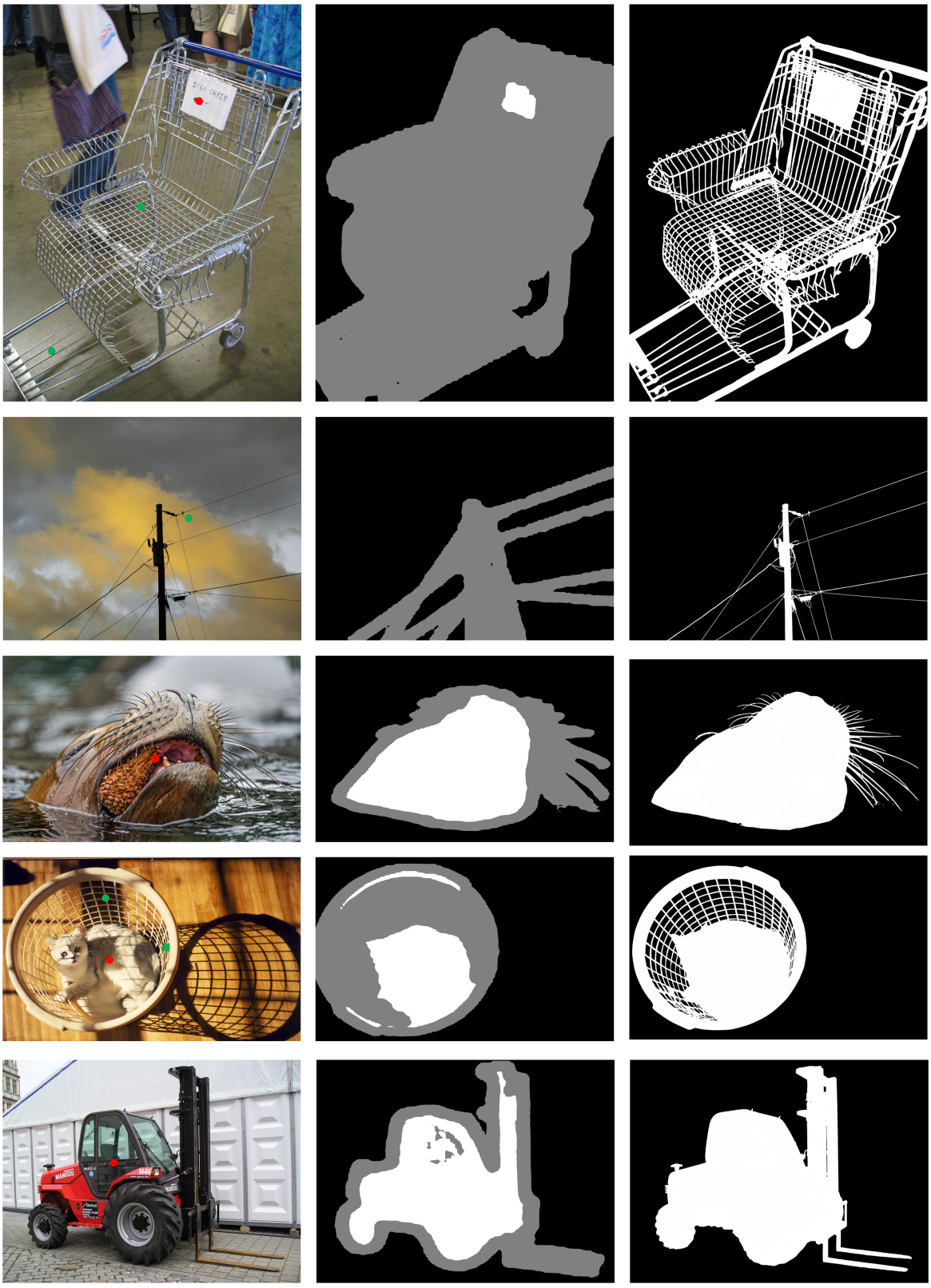

Figure F. •, and • refer to *foreground click* and *background click*, • refers to *noise-tolerant click*.