Supplementary Material for SegRefiner: Towards Model-Agnostic Segmentation Refinement with Discrete Diffusion Process

Anonymous Author(s)
Affiliation
Address
email

1 Implementation Details

In this section, we give a detailed description of the model architecture and training/inference settings. The overall workflow of the training and inference process are provided in Alg. 1 and Alg. 2.

Algorithm 1 Training

Input total diffusion steps T, datasets$D = \{(I, M_{fine}, M_{coarse})^K\}$
repeat
  Sample $(I, M_{fine}, M_{coarse}) \sim D$
  Sample $t \sim Uniform(1, \dots, T)$
  Initialize $m_0 = M_{fine}$, $x_0^{i,j} = [1, 0]$
  $q(x_t^{i,j}|x_0^{i,j}) = x_0^{i,j}\bar{Q}_t$
  Sample $x_t^{i,j} \sim q(x_t^{i,j}|x_0^{i,j})$, get $x_t \in \{0, 1\}^{2\times H\times W}$
  Pixels Transition $m_t = x_t[0] \odot M_{fine} + x_t[1] \odot M_{coarse}$
  Take gradient descent step on $\nabla_\theta \mathcal{L}(f_\theta(I, m_t, t), M_{fine})$
until convergence

Algorithm 2 Inference

Input total diffusion steps T, image and coarse mask (I, M_{coarse})
Initialize $x_T = [0, 1]$, $m_T = M_{coarse}$
for t in $\{T, T-1, \dots, 1\}$ **do**
  $\tilde{m}_{0|t},\ p_\theta(\tilde{m}_{0|t}) = f_\theta(I, m_t, t)$
  $p_\theta(x_{t-1}^{i,j}|x_t^{i,j}) = x_t^{i,j} P_{\theta,t}^{i,j}$
  Sample $x_t^{i,j} \sim p_\theta(x_{t-1}^{i,j}|x_t^{i,j})$, get $x_t \in \{0, 1\}^{2\times H\times W}$
  Pixels Transition $m_{t-1} = x_{t-1}[0] \odot \tilde{m}_{0|t} + x_{t-1}[1] \odot M_{coarse}$
return m_0

Model Architecture Following [9], we use a U-Net with 4-channel input and 1-channel output. Both input and output resolution is set to 256×256. Considering computational load and memory usage, we set the intermediate feature channels to 128 and only conduct *self-attention* in strides 16 and 32.

Submitted to 37th Conference on Neural Information Processing Systems (NeurIPS 2023). Do not distribute.

Training Settings All experiments are conducted on 8 NVIDIA RTX3090 GPUs with Pytorch. During training, we first train the *LR-SegRefiner* on the LVIS dataset [4] with 120k iterations. The AdamW optimizer is used with the initial learning rate of 4×10^{-4}. We use a multi-step learning rate schedule, which decays by 0.5 in steps 80k and 100k. Subsequently, the *HR-SegRefiner* is obtained from 40k-iterations fine-tuning based on the 80k checkpoint of *LR-SegRefiner*. Batch size is set to 8 in each GPU.

Inference Settings In instance segmentation, we use the *LR-SegRefiner* to perform refinement in instance level. For each instance, we extract the bounding box region based on the coarse mask and expand it by 20 pixels on each side. The extracted region is then resized to match the input size of the model. After a complete reverse diffusion process, the output is resized to the original size.

In semantic segmentation and dichotomous image segmentation, because of the high resolution of images, we employ the *HR-SegRefiner* and conduct a global-and-local refinement process. In order to identify the local patches that require refinement, we filter out pixels with low state-transition probabilities from the globally refined mask and use them as the center points for the local patches. We apply Non-Maximum Suppression (NMS, with 0.3 as threshold) to these patches to remove excessive overlapping.

2 More Visual Comparisons

In this section, we provide more visual results in semantic segmentation, instance segmentation, and dichotomous image segmentation. Fig. 1 shows the comparisons of SegRefiner and other models (including instance segmentation models and refinement models) on COCO [8] validation set. Fig. 2 shows more comparisons between the coarse masks and refined masks on COCO validation set. These results demonstrate that the proposed SegRfiner can robustly correct inaccurate predictions in coarse masks. Fig. 3 and Fig. 4 show visual results on BIG dataset [2] and DIS5K dataset [10]. SegRefiner shows a strong capability for capturing extremely fine details on these two high-resolution datasets.

References

[1] Liang-Chieh Chen, Yukun Zhu, George Papandreou, Florian Schroff, and Hartwig Adam. Encoder-decoder with atrous separable convolution for semantic image segmentation. In *ECCV*, pages 801–818, 2018.

[2] Ho Kei Cheng, Jihoon Chung, Yu-Wing Tai, and Chi-Keung Tang. Cascadepsp: Toward class-agnostic and very high-resolution segmentation via global and local refinement. In *CVPR*, pages 8890–8899, 2020.

[3] Yuxin Fang, Shusheng Yang, Xinggang Wang, Yu Li, Chen Fang, Ying Shan, Bin Feng, and Wenyu Liu. Instances as queries. In *ICCV*, pages 6910–6919, 2021.

[4] Agrim Gupta, Piotr Dollar, and Ross Girshick. Lvis: A dataset for large vocabulary instance segmentation. In *CVPR*, pages 5356–5364, 2019.

[5] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *CVPR*, pages 2961–2969, 2017.

[6] Lei Ke, Martin Danelljan, Xia Li, Yu-Wing Tai, Chi-Keung Tang, and Fisher Yu. Mask transfiner for high-quality instance segmentation. In *CVPR*, pages 4412–4421, 2022.

[7] Alexander Kirillov, Yuxin Wu, Kaiming He, and Ross Girshick. Pointrend: Image segmentation as rendering. In *CVPR*, pages 9799–9808, 2020.

[8] 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*, pages 740–755. Springer, 2014.

[9] Alexander Quinn Nichol and Prafulla Dhariwal. Improved denoising diffusion probabilistic models. In *ICML*, pages 8162–8171. PMLR, 2021.

[10] Xuebin Qin, Hang Dai, Xiaobin Hu, Deng-Ping Fan, Ling Shao, and Luc Van Gool. Highly accurate dichotomous image segmentation. In *ECCV*, pages 38–56. Springer, 2022.

[11] Chufeng Tang, Hang Chen, Xiao Li, Jianmin Li, Zhaoxiang Zhang, and Xiaolin Hu. Look closer to segment better: Boundary patch refinement for instance segmentation. In *CVPR*, pages 13926–13935, 2021.

[12] Zhi Tian, Chunhua Shen, and Hao Chen. Conditional convolutions for instance segmentation. In *ECCV*, pages 282–298. Springer, 2020.

[13] Xinlong Wang, Tao Kong, Chunhua Shen, Yuning Jiang, and Lei Li. Solo: Segmenting objects by locations. In *ECCV*, pages 649–665. Springer, 2020.

[14] Yuhui Yuan, Jingyi Xie, Xilin Chen, and Jingdong Wang. Segfix: Model-agnostic boundary refinement for segmentation. In *ECCV*, pages 489–506. Springer, 2020.

[15] Gang Zhang, Xin Lu, Jingru Tan, Jianmin Li, Zhaoxiang Zhang, Quanquan Li, and Xiaolin Hu. Refinemask: Towards high-quality instance segmentation with fine-grained features. In *CVPR*, pages 6861–6869, 2021.

(a) Image

(b) Mask R-CNN [5]

(c) SOLO [13]

(d) QueryInst [3]

(e) CondInst [12]

(f) PointRend [7]

Figure 1: Visual comparisons with other instance segmentation and refinement methods on COCO dataset. Our SegRefiner can robustly correct prediction errors both outside and inside the coarse mask. (Please refer to the next page for the remaining portion of this figure.)

(g) RefineMask [15]

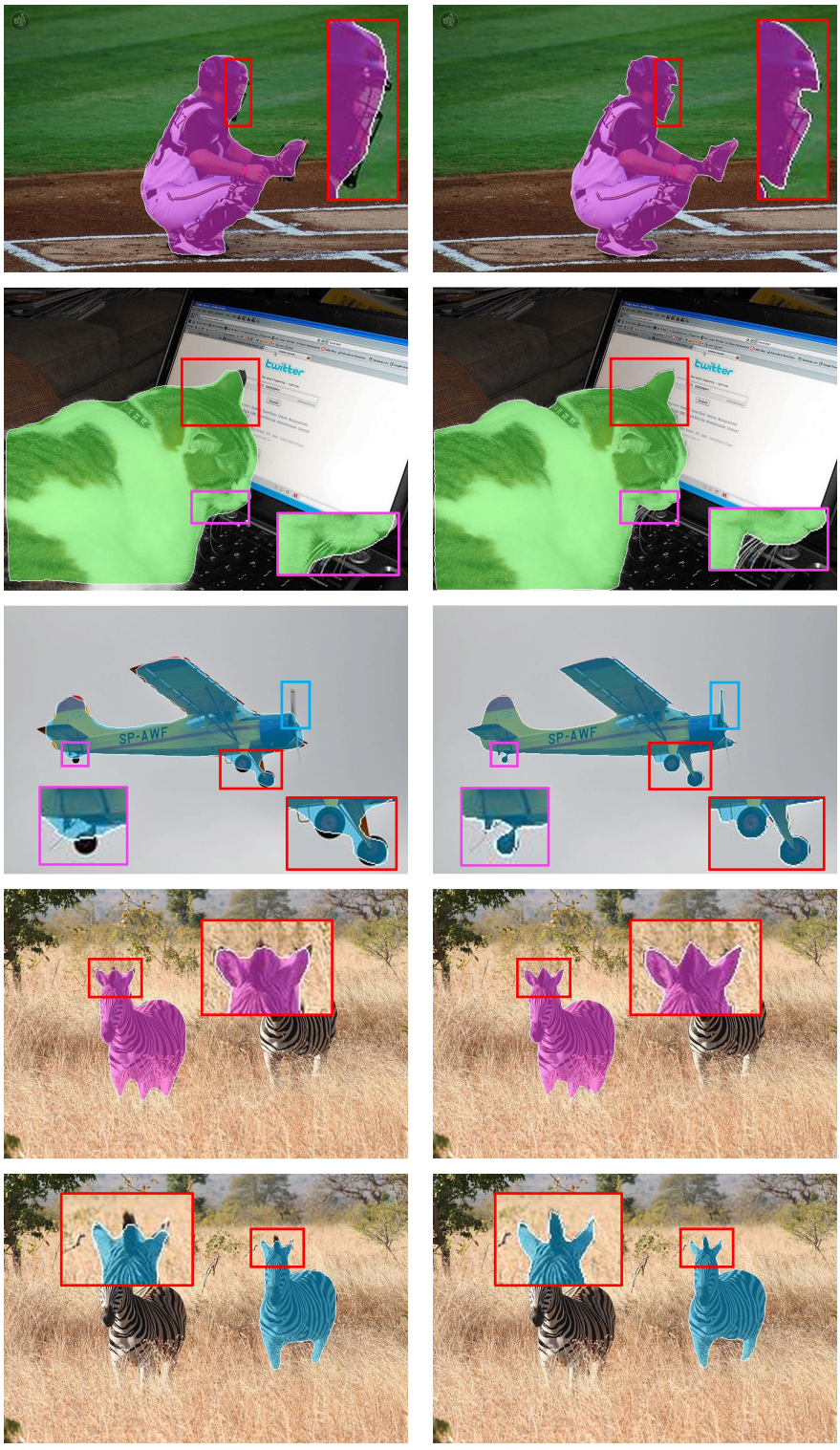
(h) Transfiner [6]

(i) Mask R-CNN + SegFix [14]

(j) Mask R-CNN + BPR [11]

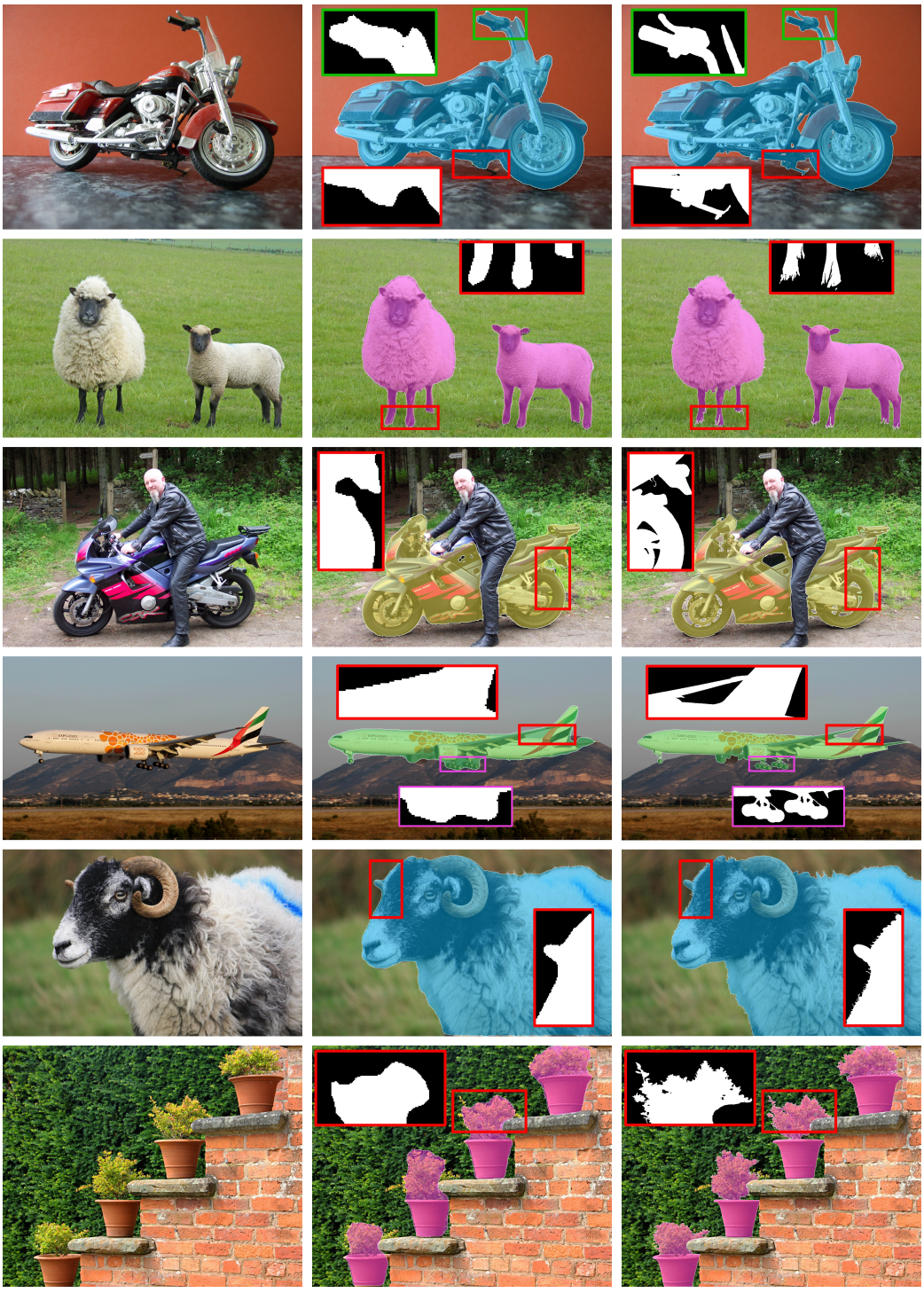
(k) Mask R-CNN + SegRefiner (Ours)

Figure 1: Visual comparisons with other instance segmentation and refinement methods on COCO dataset. Our SegRefiner can robustly correct prediction errors both outside and inside the coarse mask.



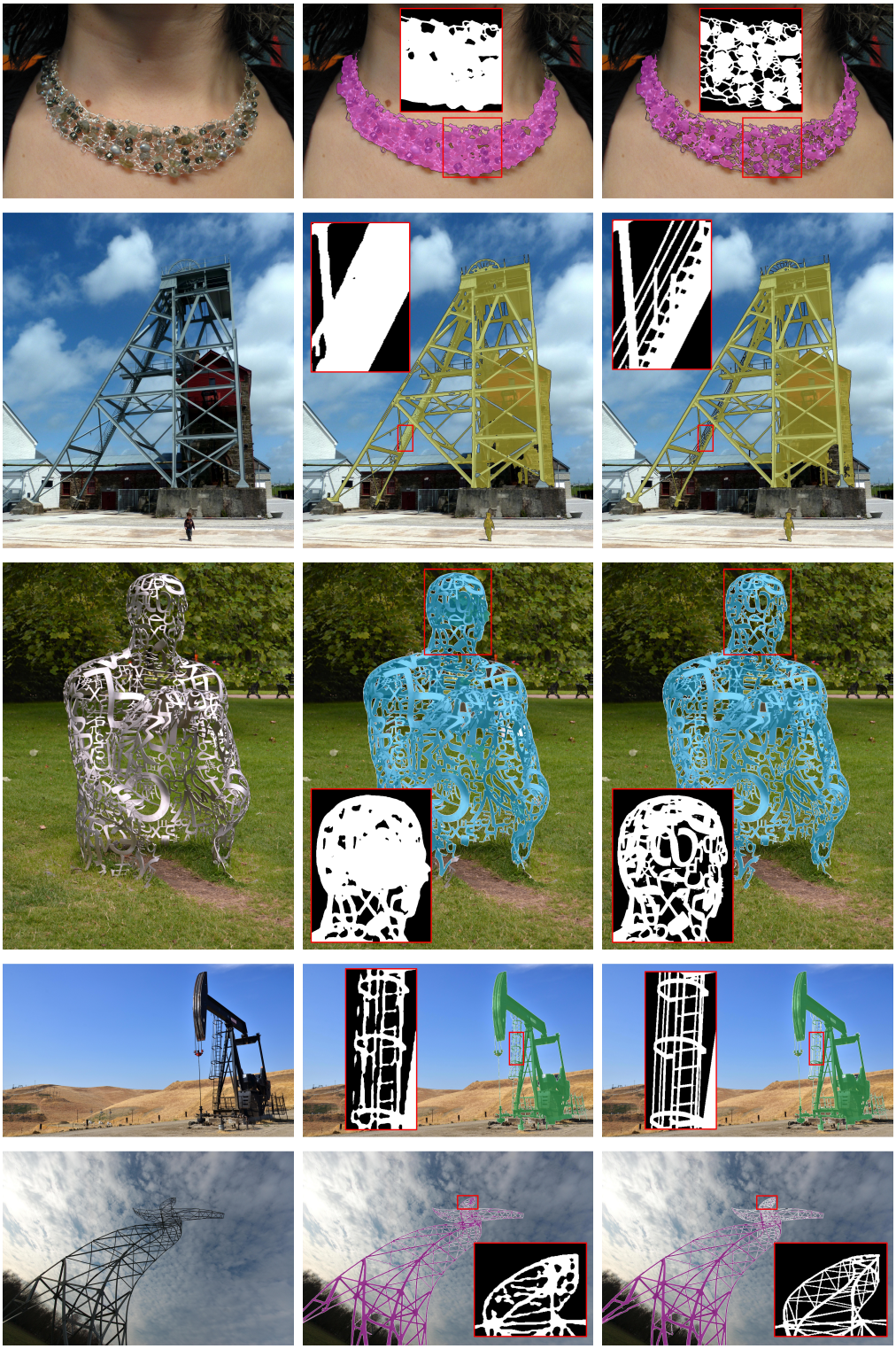
(a) Coarse Mask (b) Refined Mask

Figure 2: More visual results on COCO dataset. Coarse masks are obtained from Mask R-CNN [5]. Our SefRefiner corrects the errors of coarse masks (see Refined Mask).



(a) Image (b) Coarse Mask (c) Refined Mask

Figure 3: More visual results on BIG dataset [2]. Coarse masks are obtained from Deeplab v3+ [1]. Our SefRefiner greatly enhances the mask quality (see Refined Mask). Please kindly zoom in for a better view.



(a) Image (b) Coarse Mask (c) Refined Mask

Figure 4: More visual results on DIS5K dataset [10]. Coarse masks are obtained from ISNet [10]. Our SefRefiner captures extremely fine details (see Refined Mask). Please kindly zoom in for a better view.