OMISSION-FREE INPAINTING: A THREE-STAGE APPROACH TO ENSURE OBJECT GENERATION
- SUPPLEMENTARY MATERIAL -

Hea In Jeong, Boeun Kim, Chung-Il Kim, Saim Shin

Artificial Intelligence Research Center,
Korea Electronics Technology Institute (KETI), South Korea

1. DETAILS OF EQUATION 3

$$
\begin{aligned}
L_{D_{hinge}}(D_1, RN) &= E_{x\sim p_{data}}[max(0, 1 - D_1(x))] \\
&\quad + E[max(0, 1 + D_1(RN(I_{comb})))], \\
L_{G_{hinge}}(D_1, RN) &= -E_{z\sim p_z}[D_1(RN(I_{comb}))], \\
L_{D_{hinge}}(D_2, RN) &= E_{x\sim p_{data}}[max(0, 1 - D_2(x \times m))] \\
&\quad + E[max(0, 1 + D_2(RN(I_{comb}) \times m))], \\
L_{G_{hinge}}(D_2, RN) &= -E[D_2(RN(I_{comb}) \times m)], \\
L_{unmasked} &= \frac{1}{N_{I_{unmasked}}} \|(1 - m) \odot (RN(I_{comb}) - x)\|_1, \\
L_{bg} &= \frac{1}{N_{I_{bg}}} \|(m - B) \odot (RN(I_{comb}) - x)\|_1
\end{aligned}
$$

D_1 and D_2 use hinge loss. I_{comb} is the stitched image of inpainted background of Stage 1 and generated object of Stage 2. m is the binary mask, where 1 indicates the masked region and 0 indicates the unmasked region. RN denotes refinement network and x denotes ground truth image.. $N_{I_{unmasked}}$ is number of unmasked pixels and $N_{I_{bg}}$ is number of generated background pixels from Stage 1. B denotes a blending mask from Stage 2.

2. IMPLEMENTATION DETAILS

For the experimental implementations of the proposed frameworks, the background generator and refinement network are trained with one NVIDIA TITAN RTX GPU. The object generator is trained with two NVIDIA A100 GPUs. All networks are implemented using PyTorch. The weight parameters in Equation 3 are set to $\lambda_1 = 0.5$, $\lambda_2 = 0.1$, $\lambda_3 = 6$ and $\lambda_4 = 6$. The image resolution of input and output for the background generator and refinement network are 256×256. And the input of the object generator is a latent vector with 128 and the output image resolution is a 128×128 image. The ratio of train, validation and test set is 8:1:1.

3. EXAMPLE OF MASKGIT'S CLASS MISMATCH

Fig. S1 demonstrates example of inpainting a "cup". As seen in Fig. S1 (c), MaskGit does generate an object, but the appearance is far from a cup.

4. ADDITIONAL COMPARISONS

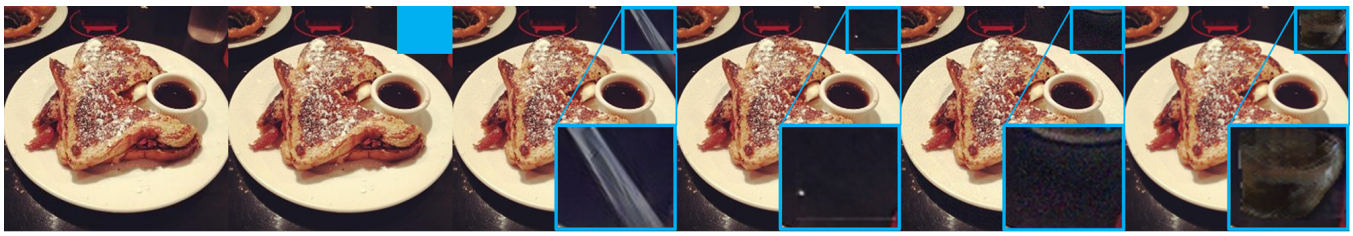
In Fig. S2, more comparison results between the proposed framework and baselines are demonstrated. The baselines are MaskGit [1] GLIDE [2] and BlendedDiff [3]. The results of baseline, (c), (d) and (e) sometimes restores image without the required object. Moreover, BlendedDiff mostly have pixelwises noises in the image and it can clearly be seen in fifth row of Fig. S2 (e).

5. REFERENCES

[1] Huiwen Chang, Han Zhang, Lu Jiang, Ce Liu, and William T Freeman, "Maskgit: Masked generative image transformer," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 11315–11325.

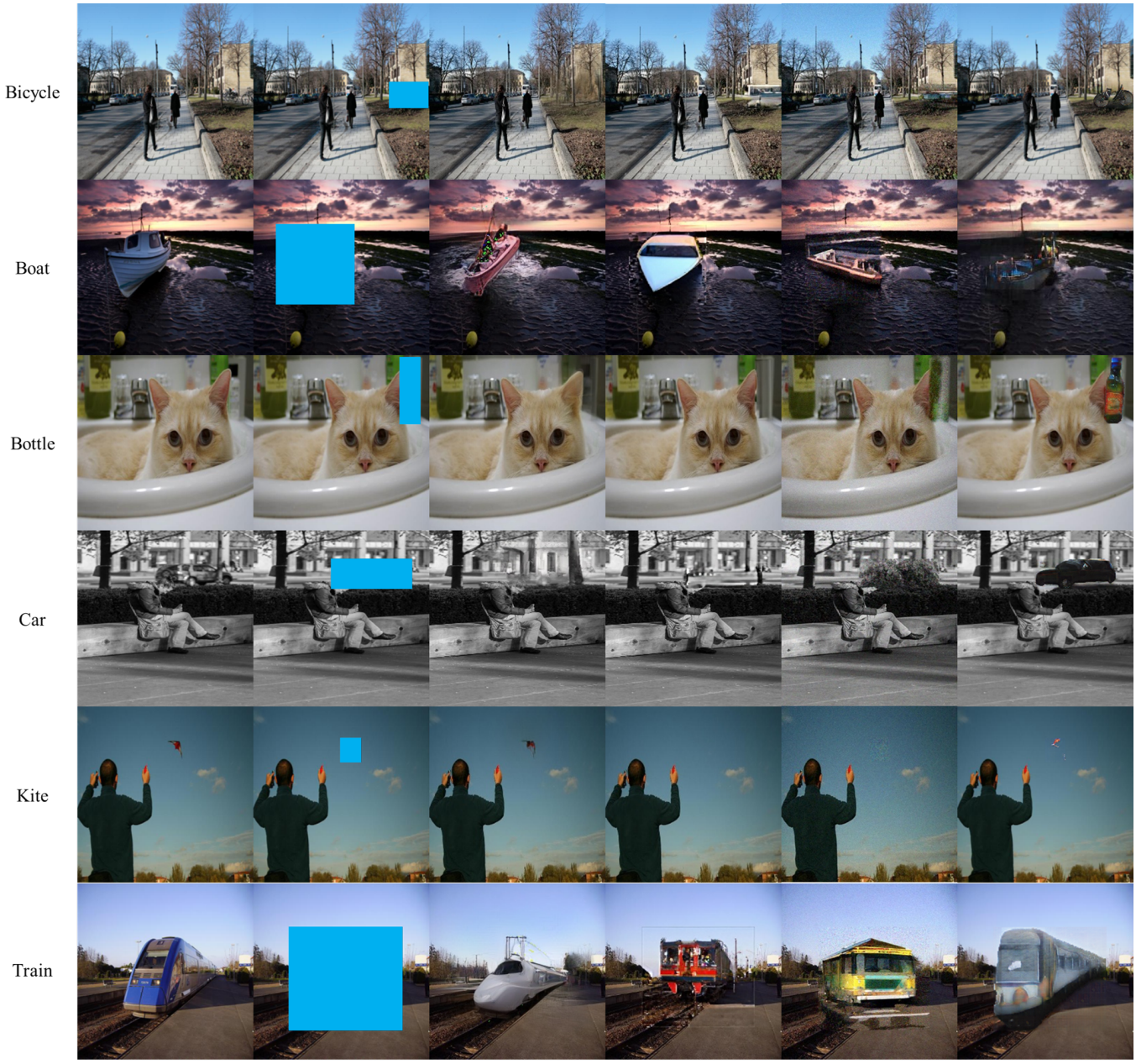
[2] Alex Nichol, Prafulla Dhariwal, Aditya Ramesh, Pranav Shyam, Pamela Mishkin, Bob McGrew, Ilya Sutskever, and Mark Chen, "Glide: Towards photorealistic image generation and editing with text-guided diffusion models," *arXiv preprint arXiv:2112.10741*, 2021.

[3] Omri Avrahami, Dani Lischinski, and Ohad Fried, "Blended diffusion for text-driven editing of natural images," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 18208–18218.



(a) Ground truth (b) Masked image (c) MaskGit [1] (d) GLIDE [2] (e) BlendedDiff [3] (f) Ours

Fig. S1. The comparisons to baselines. The guiding text is “cup”.



(a) Ground truth (b) Masked image (c) MaskGit [1] (d) GLIDE [2] (e) BlendedDiff [3] (f) Ours

Fig. S2. The comparisons to baselines. The texts in the first row indicates guiding text for text-conditional models.