The Supplementary of "LeftRefill: Filling Right Canvas based on Left Reference through Generalized Text-to-Image Diffusion Model"

Chenjie Cao[1,2]*, Yunuo Cai[1], Qiaole Dong[1], Yikai Wang[1], Yanwei Fu[1†]
[1]Fudan University, [2]Alibaba Group

1. Broader Impacts

This paper exploited image synthesis with text-to-image models. Because of their impressive generative abilities, these models may produce misinformation or fake images. So we sincerely remind users to pay attention to it. Besides, privacy and consent also become important considerations, as generative models are often trained on large-scale data. Furthermore, generative models may perpetuate biases present in the training data, leading to unfair outcomes. Therefore, we recommend users be responsible and inclusive while using these text-to-image generative models. Note that our method only focuses on technical aspects. Both images and pre-trained models used in this paper are all open-released.

2. Data Processing and Implementation Details

2.1. Data Processing for Ref-inpainting

Matching-based Masking. For the Ref-inpainting, we find that the widely used irregular mask [2, 18, 19] fails to reliably evaluate the capability of spatial transformation and structural preserving. Therefore, as shown in Figure 1(a), we propose the matching-based masking method. Specifically, we first utilize the scene info provided by MegaDepth [6] to select out the image pairs which have an overlap rate between 40% and 70% Second, for each image pair, we use a feature matching model [14] to detect matching key-points between the images and assign each key-points pair a confidence score. Next, we filter out those pairs with low confidence scores with the threshold of 0.8. Then we randomly crop a 20% to 50% sub-space in the matched region and sample 15 to 30 key points as vertices to be painted across for the final masks. The matching-based mask not only improves the reliability during the evaluation but also facilitates the performance in the training phase as in Table 3.

We split 505 pairs from MegaDepth [6] as the validation, including some manual masks from ETH3D scenes [12]. For the multi-view testing set, we further filter all scenes and retain the ones with at least 4 reference views. Thus there are 482 images in the final multi-view testing set.

2.2. Data Processing for NVS

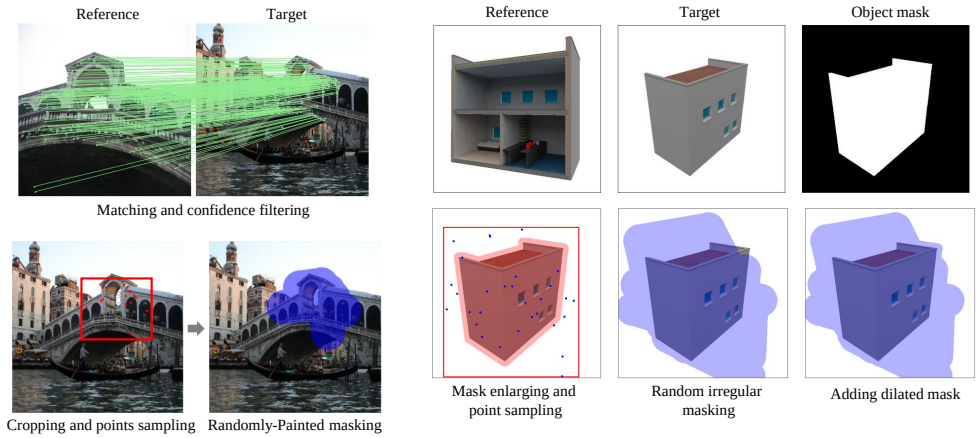
For the NVS, we first dilate the object mask and randomly sample points in the enlarged mask bounding box to paint the irregular mask. Then, we unite the dilated object mask to completely cover target images as in Figure 1(b). We find that local masking is still very important for fast convergence and stable fine-tuning as empirically verified in experiments. For the data processing on Objaverse [1], Zero123 [7] provided images including 800k various scenes with object masks. For each scene, 12 images are rendered in 256×256 with different viewpoints. Following [7], the spherical coordinate system is used to convert the relative pose Δp into the polar angle θ, azimuth angle ϕ, and radius r distanced from the canonical center as $\Delta p = (\Delta\theta, \sin\Delta\phi, \cos\Delta\phi, \Delta r)$, where the azimuth angle is sinusoidally encoded to address the non-continuity. In practice, we calculate the relative pose between the *first* view and the target view for the pose input to LeftRefill. For example, given a group of 4-view stitched input images, we provide relative poses of view 0-to-1, 0-to-2, 0-to-3, and 0-to-4, respectively. For the masking of Objaverse images, we dilate the object mask and related bounding box with 10 to 25 kernel size and 5% to 20% respectively. Then we randomly sample 20 to 45 points to paint the irregular masks.

We select 500 scenes from Objaverse as the validation, while others are used as the training set. Note that there exists an overlap between our validation and Zero123's training set [7], but our method still outperforms the official Zero123 as in the main paper.

2.3. Training Details

We show the training details in Table 1. LeftRefill is efficient in being trained for various tasks. To further demonstrate the effectiveness of LeftRefill, we provide the training log of LeftRefill and Zero123 in Figure 2. Obviously, the contextual inpainting-based LeftRefill enjoys a substantially faster convergence and superior performance.

*Dr. Chenjie Cao is at Alibaba Group. This work was accomplished while Dr. Chenjie Cao was at Fudan University.
†Corresponding author: yanweifu@fudan.edu.cn
Other emails: {cjcao20,yncai20,qldong18,yikaiwang19}@fudan.edu.cn



(a) Matching-based Masking (b) Masking for Objaverse images

Figure 1. The illustration of (a) matching-based masking for Ref-inpainting, and (b) masking strategy used for NVS on Objaverse [1].

Table 1. Training details of LeftRefill. NVS (4-view) and Ref-inpainting (4-view) are trained on $\times 8$ and $\times 4$ A800 GPUs respectively, while others are trained on $\times 2$ A6000 GPUs. NVS (4-view) is fine-tuned based on NVS (1-view).

Task	Batch size	Learning rate		Steps
		Prompt&LoRA	Backbone	
Ref-inpainting (1-view)	16	3e-5	/	6k
Ref-inpainting (2-view)	16	3e-5	/	6k
Ref-inpainting (3-view)	24	3e-5	/	16k
Ref-inpainting (4-view)	64	5e-5	/	16k
NVS-simple (1-view)	48	1e-4	1e-5	80k
NVS (4-view)	512	1e-4	3e-5	110k

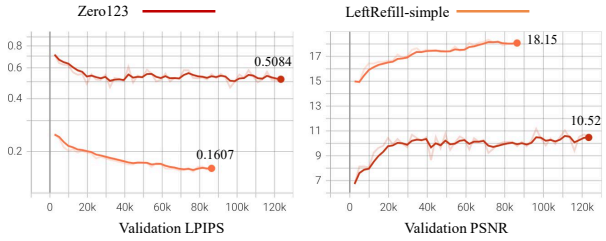


Figure 2. NVS training logs of LeftRefill and Zero123 [7] on Objaverse [1] (batch size 48, learning rate 1e-5).

Table 2. Results of 1-view NVS on Objaverse. Zero123* was re-trained with the same setting as LeftRefill-simple (batch 48).

Methods	PSNR↑	SSIM↑	LPIPS↓	CLIP↑
Zero123* (re-trained)	14.316	0.802	0.3455	0.6549
LeftRefill-simple (prompt tuning)	16.385	0.855	0.2468	0.7107
LeftRefill-simple (LoRA) [4]	19.514	0.869	0.1534	0.7589
LeftRefill-simple (fine-tune)	**20.508**	**0.875**	**0.1288**	**0.7763**

2.4. Differences between Ref-inpainting and NVS

The proposed framework, LeftRefill, serves as a generalized solution catering to both Ref-inpainting and NVS, as detailed in our main paper. However, given the substantial disparities between NVS and Ref-inpainting tasks, we present a comprehensive overview of the minor distinct implementations of LeftRefill tailored for each task. Notably, as the inpainting fine-tuned SD suffers from a large gap in tackling NVS directly, LeftRefill requires slightly more modifications to optimize its performance for NVS. The following key adjustments were identified:

1) NVS needs to be fine-tuned for the whole LDM, while Ref-inpainting only requires prompt tuning. Note that both tasks could be addressed without test-time fine-tuning through LeftRefill.
2) NVS needs another pose FC to encode relative pose information to CLIP-H.
3) To enhance the performance of NVS, positional encoding is added before each self-attention module of LeftRefill. However, our experiments did not reveal significant improvements when positional encoding was applied to Ref-inpainting.
4) The self-attention module of multi-view NVS should be processed with the block casual masking strategy for autoregressive generation. In contrast, multi-view Ref-inpainting does not require autoregressive generation since

Algorithm 1 Pseudo codes for block casual masking.

```
# view: the view number
# length: length of the sequence, usually be h*w

mask = zeros((view, length)) # [view,length]
mask[:, 0] = 1
mask = cumsum(mask.reshape(1, view * length), dim=1) # [1,view*length]
mask = (mask.T >= mask).float()  # [view*length,view*length]
mask = 1 - mask # masked regions are 1, unmasked regions are 0
mask = mask.masked_fill(mask == 1, -inf) # let all masked regions to -inf
```

Algorithm 2 Pseudo codes for the attention visualization.

```
# x: [b,2hw,c], input feature for attention module (left:reference, right:target)
# mask: [b,2hw,1], input 0-1 mask; 1 means masked regions

q, k = matmul(x, Wq), matmul(x, Wk) # [b,2hw,c], project x to query (q) and key (k)
A = matmul(q, k.T) # [b,2hw,2hw], get attenion map
A = mean(A * mask , dim=1) # [b,2hw] get mean scores attended by masked regions
A = A.reshape(b,h,w)[:, :, :w//2] # [b,h,w], show reference attention score only
```

only one view needs to be generated.

Despite these nuanced differences between Ref-inpainting and NVS within the LeftRefill framework, we clarify that it remains a sufficiently generalized model capable of effectively handling reference-based synthesis.

3. Autoregressively Sequential Generation

To verify the generalization of our method, we generate more groups of multi-view images through a single input view as in Figure 16. Moreover, we test several real-world cases with one RGB input in Figure 4. All poses are initialized to $[0.5\pi, 0, 1.5]$ for polar angle, azimuth angle, and radius distance, respectively. The proposed LeftRefill can be well generalized to real-world cases.

3.1. Adaptive Masking

One may ask that the masking strategy used in Figure 1(b) suffers from shape leakages, which lead to unreliable metrics in the main paper. We should clarify that our method can perform well only with the reference mask, which is easy to get by the salient object detection [9]. Specifically, we dilate the reference mask as Figure 1(b). Then, a few DDIM steps [13] are used to generate a rough synthesis in the target view. After that, we detect the foreground mask based on the rough synthesis by [9] and further dilate this mask for the second synthesis with full DDIM steps. The adaptive masking can be well generalized to the NVS as verified in Figure 3. All testing results in this paper without specific descriptions are already based on adaptive masking. Besides, we think that providing target masks according to the distance and direction priors manually is also convincing to address the challenging single-view NVS.

4. Supplemental Experimental Results

We show more impressive results of LeftRefill in Figure 6.

4.1. Supplemental Ablation Studies

Matching-based Masks and Noise Coefficient. On the left of Table 3, we find that the matching-based mask enjoys substantial improvement in the reference-guided inpainting. Besides, setting the noise coefficient $\eta = 1$ achieves consistent improvements in our LeftRefill even sampled as the DDIM [13]. So all LDMs are worked under $\eta = 1$ without special illustrations.

Prompt Initialization. We tried three initialization ways for prompt tuning on the right of Table 3. The random initialization performs worst. Both 'token-wise' and 'token-avgs' leverage text embeddings from a task-specific descriptive sentence listed as follows. For the Ref-inpainting, the description is "The whole image is split into two parts with the same size, they share the same scene/landmark captured with different viewpoints and times". For the NVS, the description is "Left is the reference image, while the right one is the target image with a different viewpoint. The relative pose:". Note that our encoded pose embedding is concatenated to the end of the task description embeddings. 'Token-wise' means repeating descriptive sentences until the prompt length, while each token is initialized for one prompt token. 'Token-avgs' indicates that all prompt tokens are initialized with the average of the descriptive sentence. Meaningful initialization is useful for task-specific prompt tuning.

More Details about CFG. We remove the pose condition with 15% to train the LeftRefill for NVS. Then the CFG coefficient 2.5 is used during the inference. As verified in Table 4 and Figure 8, appropriate CFG could improve the performance with better pose control and shape generation, while high CFG weights suffer from over-saturated issues. Moreover, we find that CFG can also enhance the performance of Ref-inpainting even without training with prompts dropout as in Table 5. The LPIPS initially decreases but then increases as the CFG decreases from 2.5 to 1.0, while the PSNR and the SSIM keep increasing. We consider LPIPS as the most crucial metric, as it aligns with human perception.

Figure 3. Long sequence synthesis from a single image (upper) with adaptive masking (bottom). The leftmost image and mask are the input while others are generated.

Table 3. Ablation studies of Ref-inpainting on MegaDepth. Left: effects of matching-based masks and inference noise η. Right: effects of different prompt initialization.

Configuration	PSNR↑	SSIM↑	LPIPS↓
baseline	20.489	0.829	0.1029
+ Match mask	20.574	0.830	0.1010
+ η=1.0	**20.993**	**0.837**	**0.0951**

Prompt init	PSNR↑	SSIM↑	LPIPS↓
Random	20.810	0.832	0.0998
Token-wise	20.852	0.833	0.1002
Token-avgs	**20.926**	**0.836**	**0.0961**

Table 4. Abaltions of CFG on Objaverse [1] NVS.

CFG training	CFG weight	PSNR↑	SSIM↑	LPIPS↓
×	1.0	20.310	0.872	0.1318
✓	1.0	20.352	0.873	0.1322
✓	1.5	**20.528**	0.874	0.1297
✓	2.5	20.508	**0.875**	**0.1288**
✓	5.0	20.077	0.873	0.1310

Table 5. Abaltions of CFG on MegaDepth [6] Ref-inpainting.

Ref Views	CFG weight	PSNR↑	SSIM↑	LPIPS↓
1	1.0	**21.502**	**0.840**	0.1030
	1.5	21.482	**0.840**	0.0955
	2.0	21.195	0.837	**0.0946**
	2.5	20.761	0.832	0.0969
2	1.0	**21.511**	**0.840**	0.105
	1.5	21.451	**0.840**	0.0977
	2.0	21.092	0.836	**0.0969**
	2.5	20.614	0.830	0.0997
3	1.0	**21.771**	**0.844**	0.0991
	1.5	21.703	**0.844**	0.0912
	2.0	21.356	0.840	**0.0901**
	2.5	20.855	0.834	0.0929
4	1.0	**22.334**	**0.851**	0.0902
	1.5	22.197	0.851	**0.0836**
	2.0	21.779	0.847	0.0839
	2.5	21.125	0.841	0.0894

Hence, when testing our model for Ref-inpainting, we opt to set CFG to 2.0. Furthermore, qualitative CFG results shown in the main paper also prove that 2.0 is a suitable trade-off between geometry and texture.

Table 6. Quantitative Ref-inpainting results compared to Ref-only ControlNet and side-by-side inpainting without prompt tuning.

	PSNR↑	SSIM↑	LPIPS↓
Ref-only	19.95	0.822	0.143
Side-by-Side	20.34	0.827	0.130
LeftRefill	**20.93**	**0.836**	**0.096**

Attention Visualization with Increased References. We visualize the attention map for increased reference views under DDIM step 20 in Figure 9. More reference views help to rectify both inpainted results and attention maps. Note that we also show the result without any reference in Figure 9, which can be seen as vanilla inpainting. The prompt tuning fails to recover correct structures without reliable reference.

4.2. Results of Ref-inpainting

We provide more qualitative and quantitative results of Ref-inpainting* in Figure 11, Figure 12, and Table 7. Since most instances should be defined as object removal tasks without ground truth, quantitative metrics are for reference only. But LeftRefill still outperforms TransFill in FID and LPIPS with perceptually pleasant results. Moreover, as shown in Figure 11, LeftRefill enjoys good generalization in unseen or occluded regions, because it gets rid of the constrained geometric warping.

Ref-only ControlNet and Side-by-Side Inpainting. We further compare our LeftRefill to the popular Reference-only (Ref-only) ControlNet† and side-by-side inpainting (without prompt tuning) as in Figure 10 and Table 6. However, they failed to address Ref-inpainting, retaining lower priority com-

*Since TransFill [19] is not released, we send our images and masks to the authors and take their inpainted results for the evaluation.

†https://github.com/Mikubill/sd-webui-controlnet/discussions/1236.

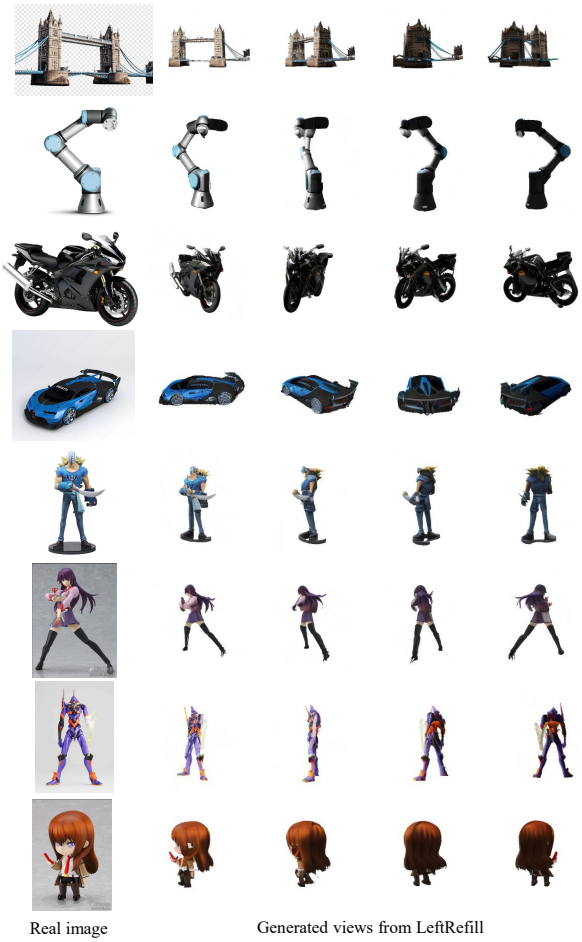


Figure 4. Consistent real-world NVS results generated by LeftRefill.

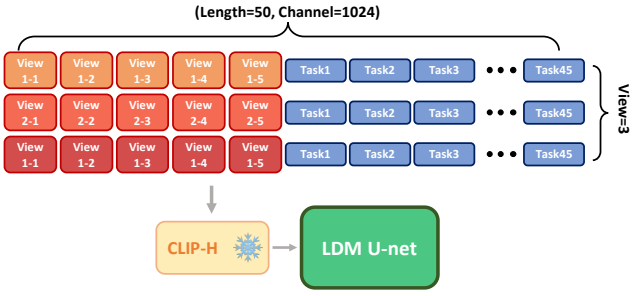


Figure 5. The illustration of task and view prompt tuning. This case shows the situation of view number 3, the length of total prompts, unshared view prompts, and shared task prompts are 50, 5, and 45, respectively.

pared to other competitors in our main paper. Particularly, Ref-only ControlNet just limits attention fields, struggling to learn reasonable correlations. While side-by-side inpainting only stitches reference and target together without explicit instruction to control proper generation.

Table 7. Ref-inpainting results on the real-world set [19].

Method	PSNR↑	SSIM↑	FID↓	LPIPS↓
ProFill [16]	25.550	0.944	71.758	0.0848
TransFill [19]	**26.052**	**0.945**	62.493	0.0757
LeftRefill	25.733	0.942	**61.276**	**0.0756**

Table 8. The out-of-distribution comparison on Google Scanned Objects [3].

Methods	Ref-View	PSNR↑	SSIM↑	LPIPS↓	CLIP↑
Zero123 [7]	1	18.794	0.851	0.1132	0.7270
LeftRefill	1	**21.039**	**0.883**	**0.0909**	**0.7693**
LeftRefill	2	22.090	0.893	0.0729	0.7925
LeftRefill	3	22.917	**0.904**	0.0595	0.8089
LeftRefill	4	**23.169**	**0.904**	**0.0563**	**0.8185**

Table 9. Inference speed of SD under 50 DDIM sampling steps.

Input size	Sec/image	Input size	Sec/image
256×256	2.9172	256×512	2.9395
512×512	3.0715	512×1024	4.0205

4.3. Results of NVS

Besides, we show some diverse NVS on Objaverse [1] in Figure 13. Different random seeds are utilized to process the DDIM sampling. LeftRefill can achieve reasonable results with correct target poses. More qualitative results are in Figure 7 and Figure 17.

Comparison on Google Scanned Objects (GSO). We compare the proposed LeftRefill and zero123 [7] on the out-of-distribution GSO dataset [3] in Table 8 and Figure 18. LeftRefill enjoys good zero-shot generalization, which outperforms zero123 with 1-view inputs. More reference views can further improve the quality of LeftRefill, benefiting from our multi-view-based NVS design and AR training.

5. Inference Speed

We provide the inference speed for different input resolutions in Table 9. All tests are based on one 32GB V100 GPU with 50 DDIM steps. LeftRefill needs to stitch two images together, which would double the input size. But the inference time is not doubled as shown in Table 9. Note that when the image size is smaller than 512, the difference in inference costs is not obvious. Therefore, we think the proposed LeftRefill's inference cost is still acceptable in most real-world applications.

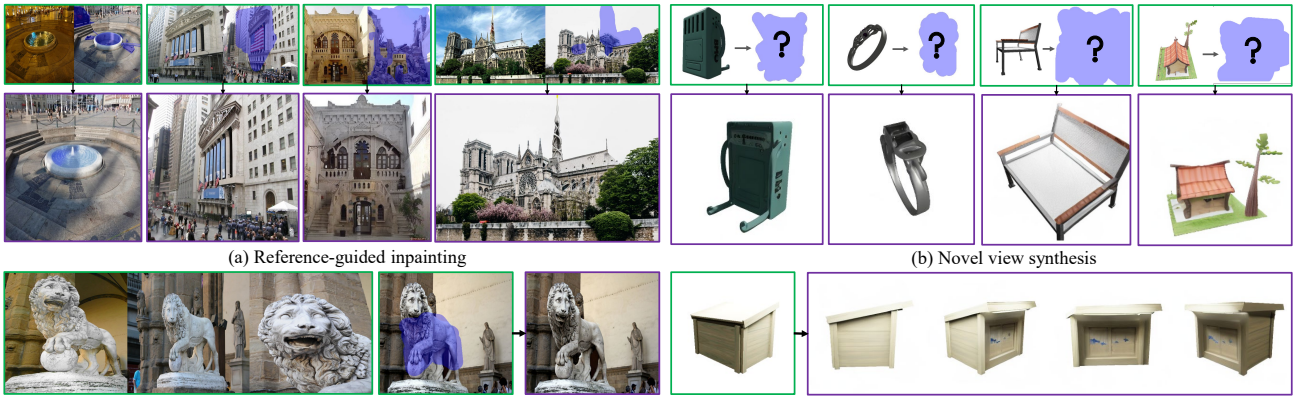


Figure 6. More impressive results of LeftRefill based on (a) Ref-inpainting, (b) NVS, (c) multi-view inpainting, and (d) multi-view synthesis.

Figure 7. NVS on Objaverse [1] from a single reference image.

6. User Study

To evaluate the effectiveness of our LeftRefill in Ref-inpainting. We further test the user study as the human perceptual metric in Figure 14. Formally, 50 masked image pairs are randomly selected from our test set which are compared among SD [11], ControlNet [17]+match [14], Perciver [5], Paint-by-Example [15], TransFill [19], and LeftRefill. Although TransFill was not open-released, we thank TransFill's authors for kindly testing these samples for us. There are 10 volunteers who are not familiar with image generation attending this study. Given masked target images and reference ones, we ask volunteers to vote for the best recovery from the 6 competitors mentioned above. The voting criterion should consider both the faithful recovery according to the reference and natural generations of color and texture. As shown in Figure 14, LeftRefill outperforms other competitors.

7. Limitation

Although the proposed LeftRefill enjoys good performance and geometric consistency in multi-view NVS, it still suffers from the drawback of *error accumulation* as shown in Figure 15. To eliminate this problem, we recommend providing a few more views (2,3,4) for more robust geometric priors. Moreover, the extension to higher resolution and improved efficiency for pre-trained models with superior capacity (SDXL [8]) can be regarded as interesting future work of LeftRefill.

References

[1] Matt Deitke, Dustin Schwenk, Jordi Salvador, Luca Weihs, Oscar Michel, Eli VanderBilt, Ludwig Schmidt, Kiana Ehsani, Aniruddha Kembhavi, and Ali Farhadi. Objaverse: A universe of annotated 3d objects. *arXiv preprint arXiv:2212.08051*, 2022. 1, 2, 4, 5, 6, 7, 10

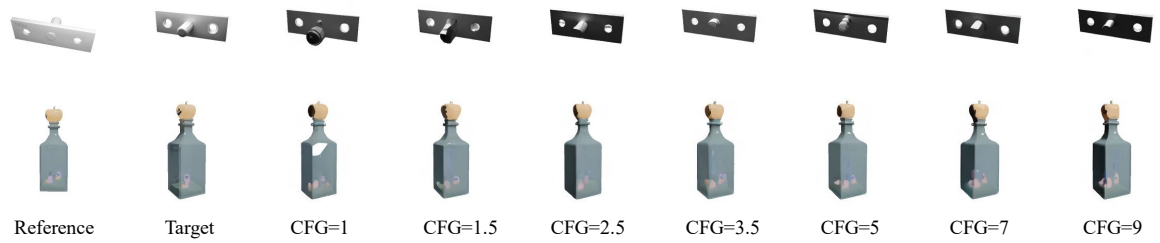


Figure 8. NVS on Objaverse [1] with different CFG weights.

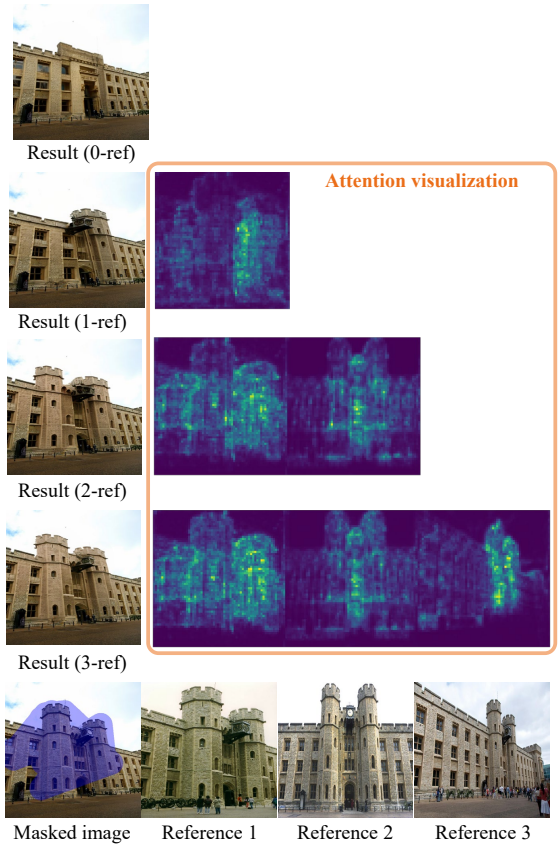


Figure 9. Attention visualization (Algorithm 2) with increased reference views.

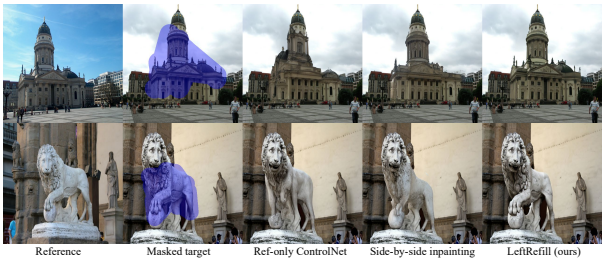


Figure 10. Qualitative Ref-inpainting results compared to Ref-only ControlNet and side-by-side inpainting without prompt tuning.

[2] Qiaole Dong, Chenjie Cao, and Yanwei Fu. Incremental transformer structure enhanced image inpainting with masking positional encoding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11358–11368, 2022. 1

[3] Laura Downs, Anthony Francis, Nate Koenig, Brandon Kinman, Ryan Hickman, Krista Reymann, Thomas B McHugh, and Vincent Vanhoucke. Google scanned objects: A high-quality dataset of 3d scanned household items. In *2022 International Conference on Robotics and Automation (ICRA)*, pages 2553–2560. IEEE, 2022. 5, 13

[4] Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*, 2021. 2

[5] Andrew Jaegle, Felix Gimeno, Andy Brock, Oriol Vinyals, Andrew Zisserman, and Joao Carreira. Perceiver: General perception with iterative attention. In *International conference on machine learning*, pages 4651–4664. PMLR, 2021. 6, 9

[6] Zhengqi Li and Noah Snavely. Megadepth: Learning single-view depth prediction from internet photos. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2041–2050, 2018. 1, 4

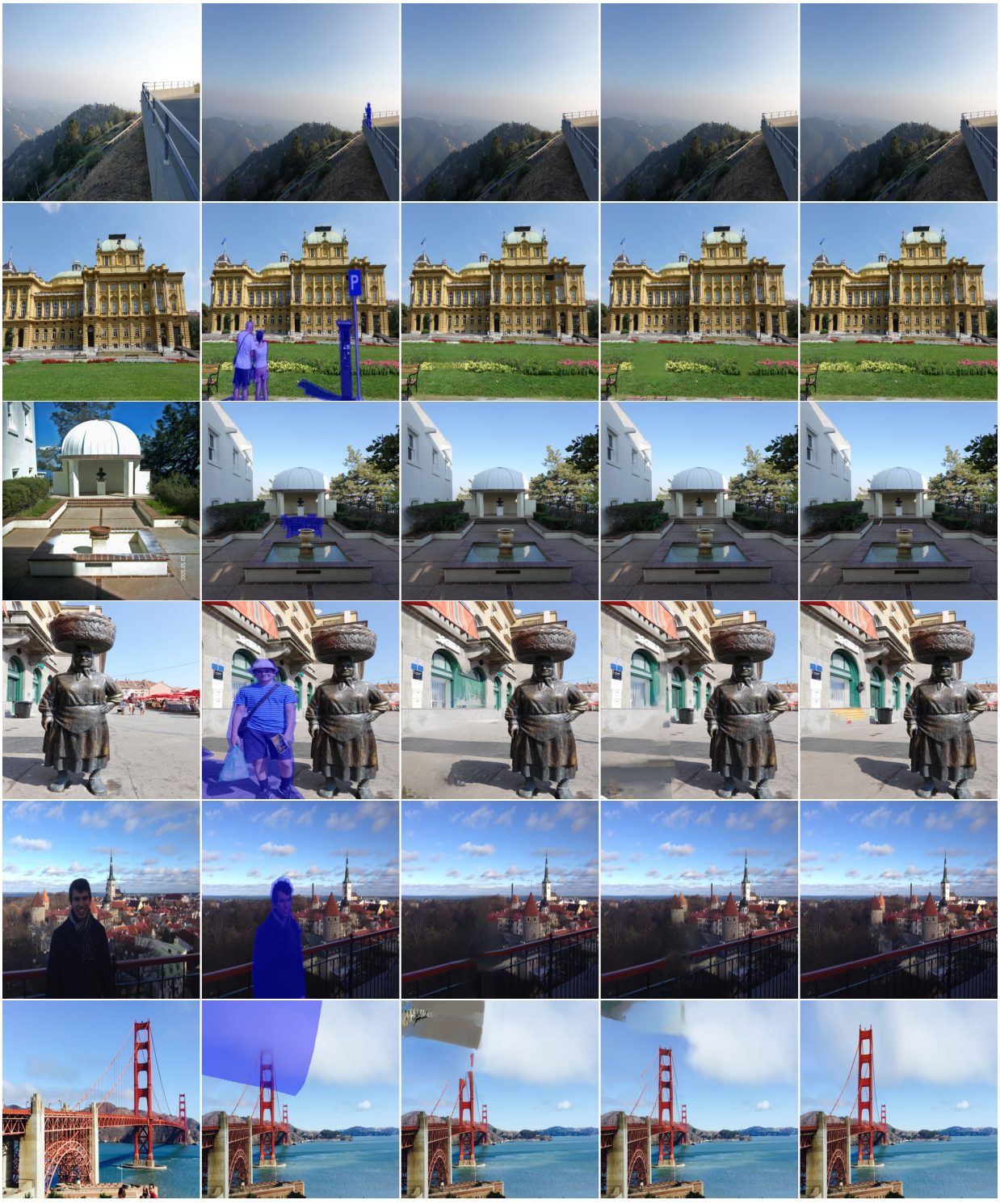
[7] Ruoshi Liu, Rundi Wu, Basile Van Hoorick, Pavel Tokmakov, Sergey Zakharov, and Carl Vondrick. Zero-1-to-3: Zero-shot one image to 3d object. In *Proceedings of the IEEE/CVF international conference on computer vision*, 2023. 1, 2, 5, 11, 12, 13

[8] Dustin Podell, Zion English, Kyle Lacey, Andreas Blattmann, Tim Dockhorn, Jonas Müller, Joe Penna, and Robin Rombach. SDXL: Improving latent diffusion models for high-resolution image synthesis. In *International Conference on Learning Representations (ICLR)*, 2024. 6

[9] Xuebin Qin, Zichen Zhang, Chenyang Huang, Masood Dehghan, Osmar R Zaiane, and Martin Jagersand. U2-net: Going deeper with nested u-structure for salient object detection. *Pattern recognition*, 106:107404, 2020. 3

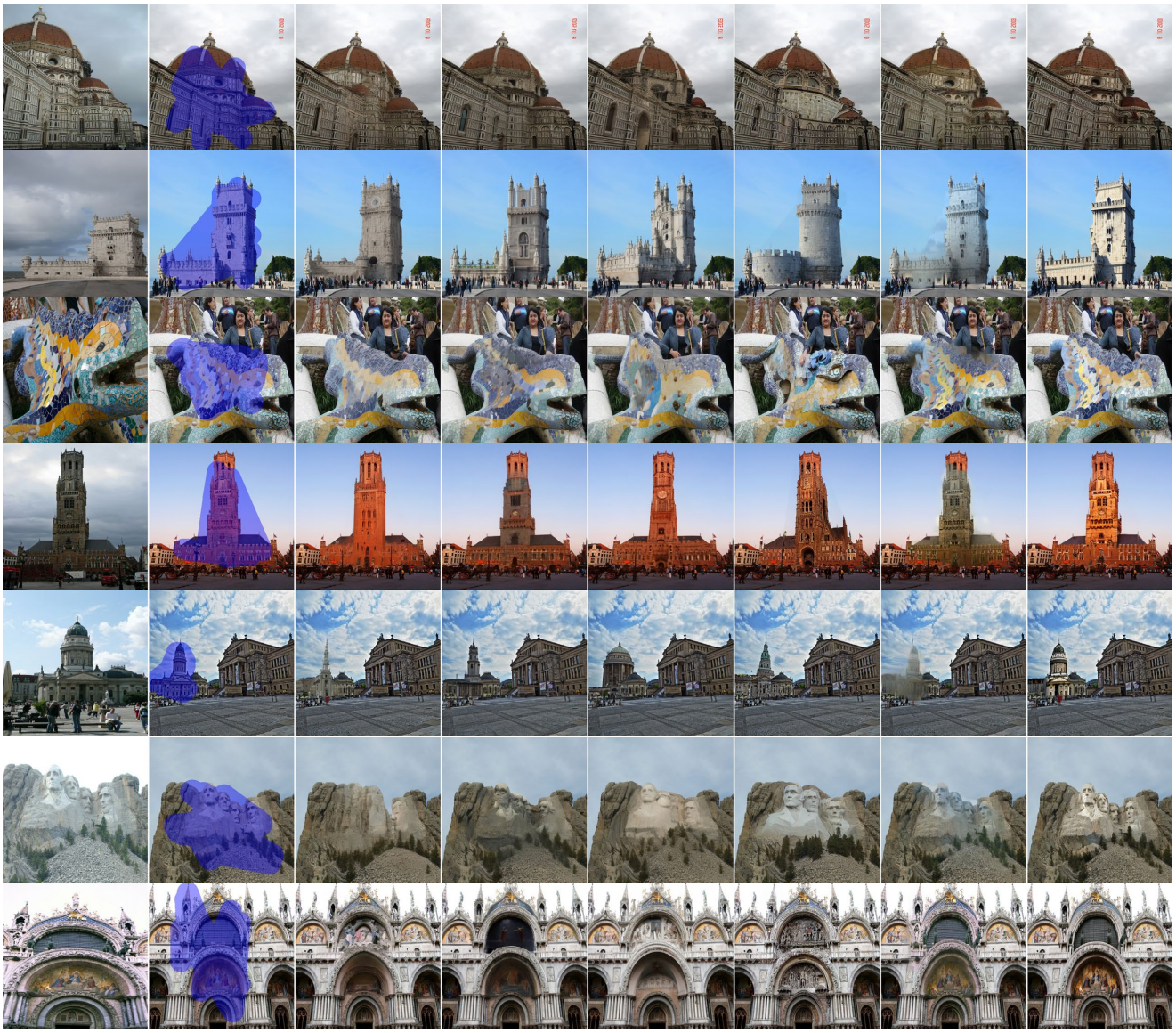
[10] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PMLR, 2021. 9

[11] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image



(a) Reference (b) Masked target (c) ProFill (d) TransFill (e) **LeftRefill**

Figure 11. Qualitative Ref-inpainting results compared with ProFill [16], TransFill [19], LeftRefill on the challenging real set provided by TransFill [19].



(a) Reference (b) Masked target (c) SD (d) Control+Match (e) Perceiver (f) Paint-by-Example (g) TransFill (h) **LeftRefill**

Figure 12. Qualitative Ref-inpainting results on MegaDepth, which are compared among (c) SD [11], (d) ControlNet [17]+Matching [14], (e) Perceiver [5] with ImageCLIP [10], (f) Paint-by-Example [15], (g) TransFill [19], and (I) our LeftRefill. Please zoom in for more details.

synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10684–10695, 2022. 6, 9

[12] Thomas Schops, Johannes L Schonberger, Silvano Galliani, Torsten Sattler, Konrad Schindler, Marc Pollefeys, and Andreas Geiger. A multi-view stereo benchmark with high-resolution images and multi-camera videos. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3260–3269, 2017. 1

[13] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. *arXiv preprint arXiv:2010.02502*, 2020. 3

[14] Shitao Tang, Jiahui Zhang, Siyu Zhu, and Ping Tan. Quadtree attention for vision transformers. *arXiv preprint arXiv:2201.02767*, 2022. 1, 6, 9

[15] Binxin Yang, Shuyang Gu, Bo Zhang, Ting Zhang, Xuejin Chen, Xiaoyan Sun, Dong Chen, and Fang Wen. Paint by example: Exemplar-based image editing with diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18381–18391, 2023. 6, 9

[16] Yu Zeng, Zhe Lin, Jimei Yang, Jianming Zhang, Eli Shechtman, and Huchuan Lu. High-resolution image inpainting with iterative confidence feedback and guided upsampling. In

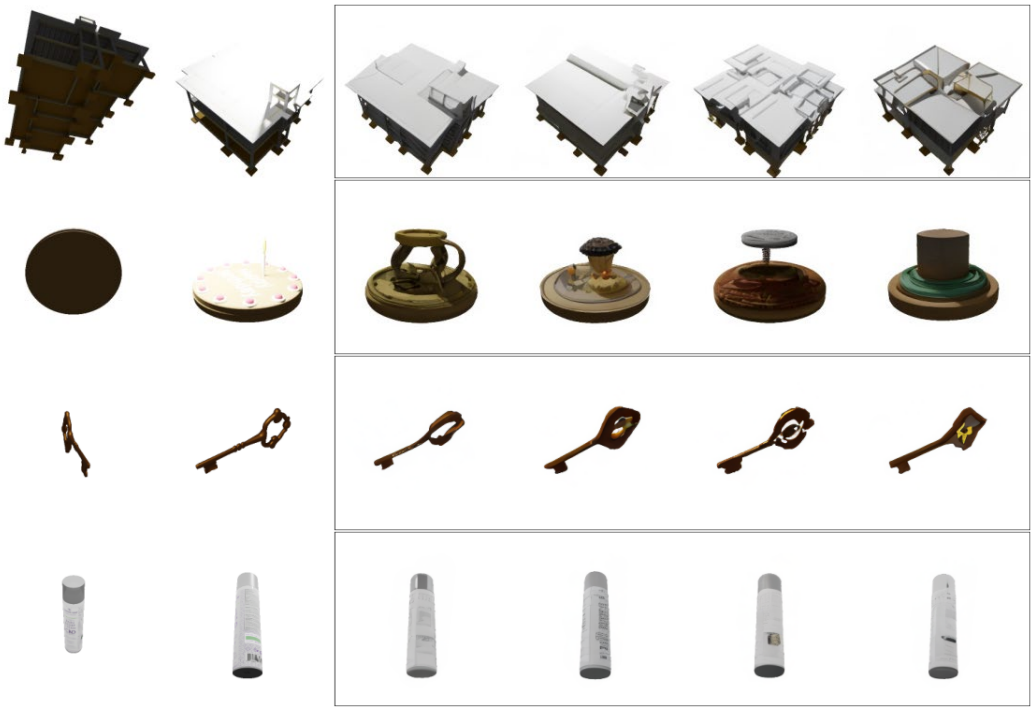


Figure 13. Diversity of the NVS on Objaverse [1] from a single reference image without multi-view guidance.

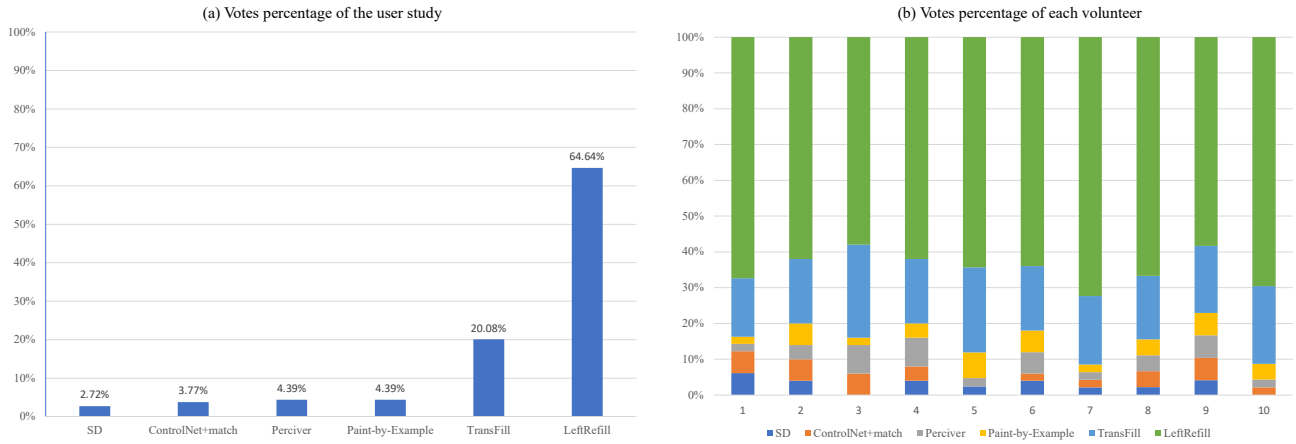


Figure 14. The user study evaluation; (a) the overall voting percentage; (b) the votes of each volunteer.

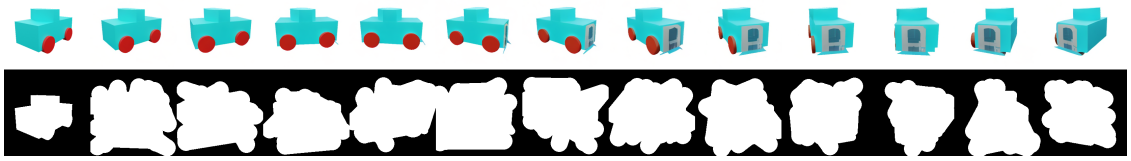


Figure 15. The error accumulation occurred in AR generation. The degraded result is first generated in view 3.

Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XIX 16, pages 1–17. Springer, 2020. 5, 8

[17] Lvmin Zhang and Maneesh Agrawala. Adding conditional control to text-to-image diffusion models. In *Proceedings of the IEEE/CVF international conference on computer vision*,

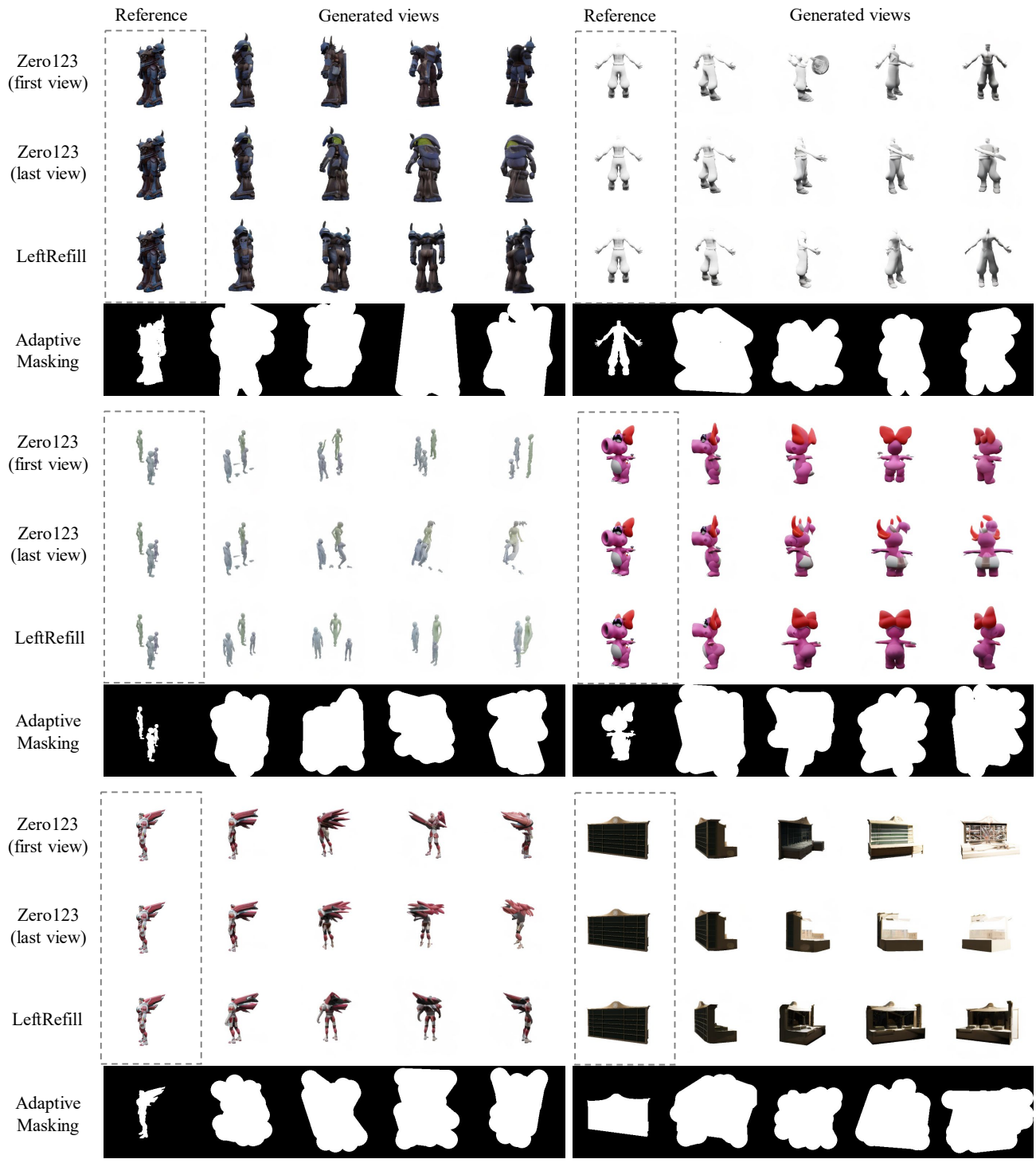


Figure 16. The sequential generative results from a single view. Zero123's [7] results are conditioned on the real reference (first view) and the last generated view (last view) respectively.

2023. 6, 9

[18] Yunhan Zhao, Connelly Barnes, Yuqian Zhou, Eli Shechtman, Sohrab Amirghodsi, and Charless Fowlkes. Geofill: Reference-based image inpainting of scenes with complex

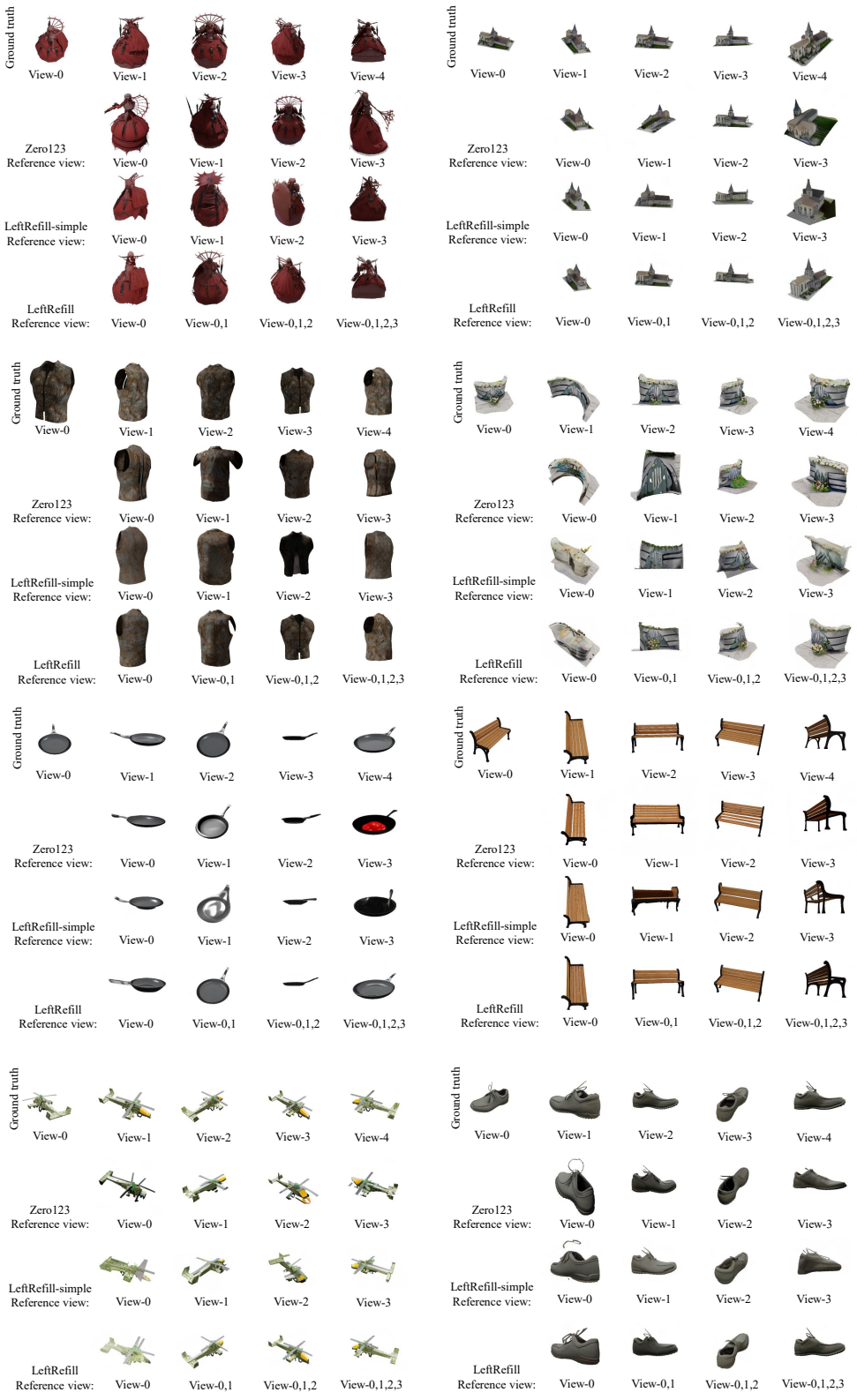


Figure 17. Multi-view NVS results on Objaverse compared among the official Zero123 [7], one-view based LeftRefill-simple, and multi-view based LeftRefill. Please zoom in for details.

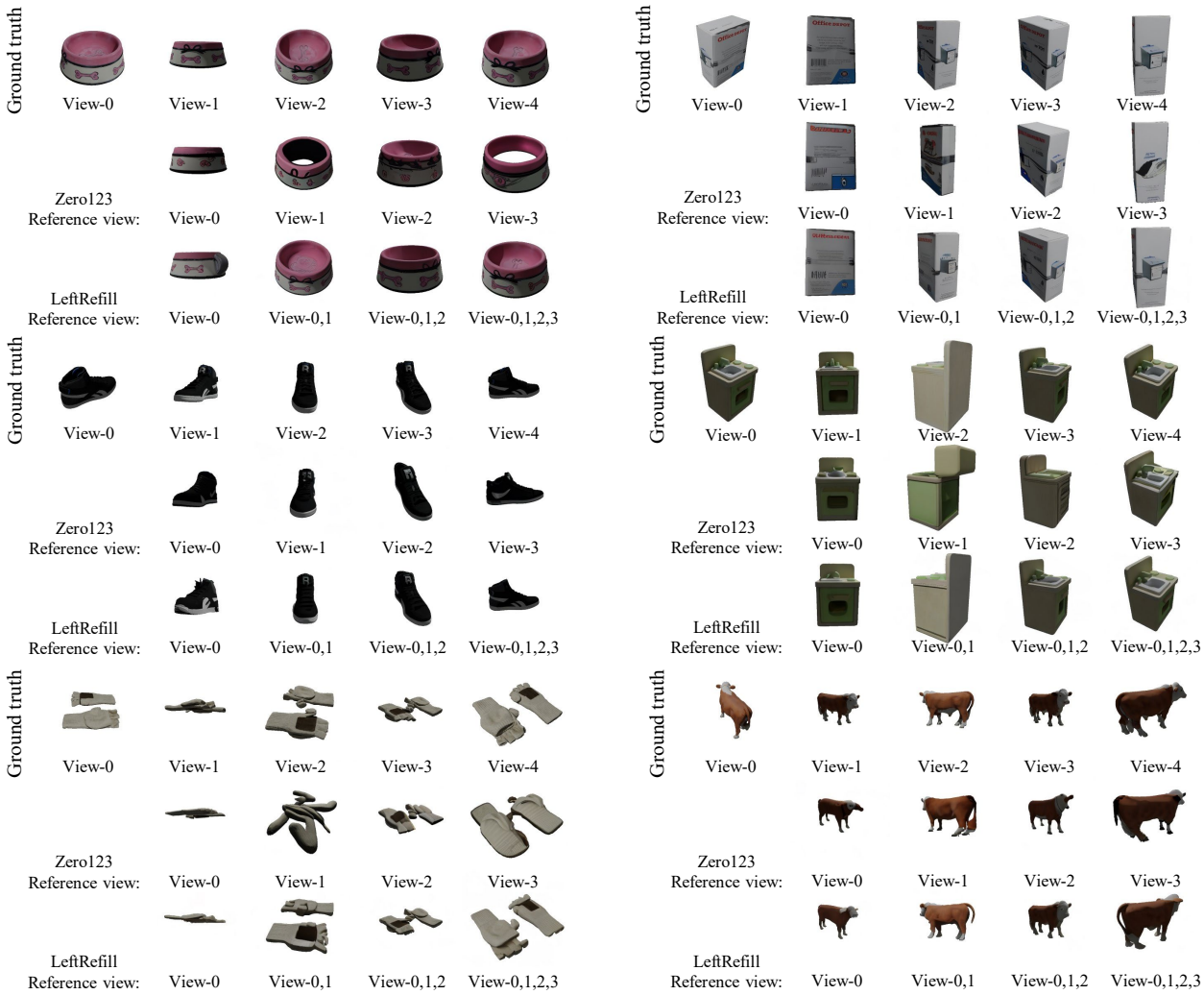


Figure 18. Multi-view NVS results on Google Scanned Objects [3] compared with official Zero123 [7] and multi-view based LeftRefill. Please zoom in for details.

geometry. *arXiv preprint arXiv:2201.08131*, 2022. 1

[19] Yuqian Zhou, Connelly Barnes, Eli Shechtman, and Sohrab Amirghodsi. Transfill: Reference-guided image inpainting by merging multiple color and spatial transformations. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 2266–2276, 2021. 1, 4, 5, 6, 8, 9