Supplementary material (Appendix) for NeRFiller: Completing Scenes via Generative 3D Inpainting

1. Summary of Supplementary Material

The following files are part of the supplementary material:

- This PDF, which describes details related to the main paper and an additional user study for perceptual evaluation.
- A short video overviewing NeRFiller, which is available on YouTube due to upload size limits: `https://youtu.be/AJNDRsS3sGs`
- "multi-view_consistent_inpainting_eval_split" available at `https://youtu.be/dB5a5YNJETs`. We show our NeRF renders for a custom 10-second camera path, for all 10 scenes.
- "3D_consistent_image_inpainting" available at `https://youtu.be/23xdEwi6YzY`. - Results for the 8 NeRF synthetic scenes. On the left, we show the inpaints. On the right, we show the resulting NeRF. We are only showing the 20 evaluation images, while 180 images were used to train the NeRF.

Acknowledgements

This project is supported in part by IARPA DOI/IBC 140D0423C0035. The views and conclusions contained herein are those of the authors and do not represent the official policies or endorsements of IARPA, DOI/IBC, of the U.S. Government. We would like to thank Frederik Warburg, David McAllister, Qianqian Wang, Matthew Tancik, Grace Luo, Dave Epstein, Riley Peterlinz for discussions and technical support. We also thank Ruilong Li, Evonne Ng, Adam Rashid, Alexander Kristoffersen, Rohan Mathur, Jonathan Zakharov for proofreading drafts and providing feedback.

2. NeRF synthetic prompts

We use the following text prompts for each of the 8 NeRF synthetic datasets from the original NeRF paper [6].

- "chair": "a photo of a chair"
- "drums": "a photo of drums"
- "ficus": "a photo of a ficus plant"
- "hotdog": "a photo of a hotdog"
- "lego": "a photo of a lego bulldozer"
- "materials": "a photo of materials"
- "mic": "a photo of a microphone"
- "ship": "a photo of a ship

3. Datasets

Some of our data can be found online with their respective hyperlinks. Others were created by the authors by scanning with Polycam, and one was obtained by modifying a SPIn-NeRF dataset.

- "dumptruck" (Sketchfab)
- "turtle" (author capture with Polycam)
- "drawing" (Sketchfab)
- "backpack" (SPIn-NeRF dataset)
- "bear" (Sketchfab)
- "billiards" (Sketchfab)
- "norway" (Sketchfab)
- "cat" (author capture with Polycam)
- "boot" (Sketchfab)
- "office" (Sketchfab)

3.1. Marking inpaint regions

To mark inpaint regions, we have a variety of approaches depending on the data. For the "office" scene, we mark inpaint regions as any missing region from the original mesh, found here on Sketchfab. For the other scenes, we place a 3D occluder, such as a cube or cylinders. For the "cat" dataset in Figure 1, we use a cylinder to mark an occlusion of the body and a rectangular prism to mark the tag by the ear. It's quite straightforward to load the mesh into Blender and add the occluders. The 3D masks can then be rendered from any view. We leave the automatic placement of 3D occlusions for future work. Uncertainty masks from Bayes' Rays [2] or deleted areas from Nerfbusters [12] could provide masks for our method, but their masks are out-of-distribution for SD (Stable Diffusion), shown in Fig. A1 and discussed in Sec. 6 of this document.

3.2. Creating NeRF datasets

To create NeRF datasets with the marked inpaint regions, we render the mesh from ∼60 images (64 for all except "backpack" which uses 60). For objects, we render around the object and for e.g., large missing rectangles from a room, we render arcs looking toward the region from different angles, at different elevations. We can mark the inpainting regions by doing a depth check between the actual mesh and the manually placed occluders. To create the "backpack" scene, we took the dataset from SPIn-NeRF [8] and modified the original tight mask around the backpack to include the top part. We also dilated the masks to make them less tight. See Figure 2 for an illustration of our changes to convert it for our scene completion setting rather than object deletion.

4. Additional experiment details

4.1. Inpaint implementation details

Image classifier-free guidance. Some percentage of the time during fine-tuning of the Stable Diffusion model for inpainting [9], everything is masked out. This is similar to dropping out the text prompt with some probability to enable classifier-free guidance. Because of this training strategy, we can enable image-guidance with a formulation similar to InstructPix2Pix [1]. We modify the text-conditioned diffusion inpainting model to predict its score estimate in the form:

$$\begin{aligned}\epsilon_\phi(z_t,t,c) = \epsilon_\phi(z_t,t,\{c_I,\emptyset\}) \\ +s_I\cdot(\epsilon_\phi(z_t,t,\{c_I,c_T\})-\epsilon_\phi(z_t,t,\{c_I,\emptyset\})) \\ +s_T\cdot(\epsilon_\phi(z_t,t,\{c_I,\emptyset\})-\epsilon_\phi(z_t,t,\{\emptyset,\emptyset\}))\end{aligned} \tag{1}$$

with z_t as the noisy latent at time t, c_I and c_T as conditioning for image & mask, and text respectively. $\emptyset$ means no text or completely masking out the image. s_I and s_T are image and text guidance scales, respectively. We use $s_I > 0$ and $s_T = 0$ to make the model conditioned only on the image and masked region to inpaint. We note that the popular diffusers[1] library doesn't enable setting $s_I > 0$ *out-of-the-box*, so most works use diffusion inpainting models by using text prompts to describe an inpaint. Trying to describe the scene, however, can be difficult and using image only guidance ("SD Image Cond", where $s_I > 0$ and $s_T = 0$) tends to be more multi-view consistent (Figure 7).

Scheduler details. We use DDIM for our scheduler. We use 20 steps whenever sampling SD [9], regardless of how much noise is added to a current render. This is similar to Instruct NeRF2NeRF [3]'s dataset update procedure. The model we use can be found here as "stabilityai/stable-diffusion-2-inpainting" on Hugging Face.

4.2. NeRF implementation details

For synthetic scenes and objects, we adopt recommended practices for training Nerfacto [11] which is to set the background color to either white or black, disable scene contraction, and turn off the distortion loss. For the "backpack" forward facing scene from SPIn-NeRF [8], we only turn off the distortion loss. Nerfacto is not tuned for LLFF [5] forward-facing scenes, so there may be some artifacts in this dataset compared to our datasets which have larger parallax. For other scenes e.g., those that resemble an indoor room, we train Nerfacto with default settings.

For the baselines besides "Masked NeRF" in Sec. 5.2 "Completing large unknown 3D regions" and Table 2, we train a modified Nerfacto that has losses on patches. Note that "Masked NeRF" is simply Nerfacto, with any modifications as described. The other baselines render 1024 patches of size 32×32 per iteration. An L1 RGB loss and LPIPS [13] loss are applied to these patches, similar to IN2N [3]. Furthermore, we start the dataset update (DU) methods from the result of "Masked NeRF". Each method is trained for 30K iterations, which is typical for Nerfacto.

4.3. Evaluation against inpainted images

We use $t \in [0.4, 1.0]$ noise added to a render before sampling SD and updating a training image. Because $t = 0.4$ is the minimum noise, the inpainter has some freedom to change the inpaint from the current NeRF render. If we added no noise ($t = 0.0$) to the current NeRF render, then the DU (dataset update) methods would be perfect on the NeRF metrics (PSNR, LPIPS, SSIM) because SD would do nothing and the inpainted images would be exactly the NeRF rendered images. Nevertheless, quantifying inpaints without a ground truth result is challenging and our metrics represent an effort to show quantitative evaluation of 3D consistency. We recommend looking at the videos to compare the methods qualitatively.

4.4. Near plane for evaluation metrics

We render with a near plane slightly in front of the training images when reporting our metrics. This is important because NeRFs can cheat by placing "floaters" in front of the training images to explain away any discrepancies in the actual 3D scene compared to the training images. By setting the near plane a bit beyond the camera origin, we can render the true 3D scene without the floaters directly in front of training images.

4.5. Novel-view video metrics

For the *MUSIQ* image quality metric, also used in [7], we take all 300 frames (10 seconds, 30 FPS) of the novel-view videos, downsample each frame by 4x to capture low-frequency structure, and run the MUSIQ [4] model (trained on the AVA dataset) to obtain an image quality score. For the *Corrs* metric, we use LoFTR [10] with a confidence threshold of 0.8. We count the number of correspondences above this confidence threshold for 100 random pairs of frames.

4.6. Metrics for each scene

We provide the individual tables for Table 1 and Table 2 of the paper. See the end of the document for these.

[1] https://github.com/huggingface/diffusers

4.7. Reference-based inpainting

For these experiments, we edit 30 images at a time (instead of 40) and make 10 grids, each with exactly 1 reference inpaint.

4.8. Speed

We report times on a NVIDIA RTX A5000. Inpainting one 512×512 image takes 1.5 sec/img, with M=50 steps of DDIM sampling. Grid Prior and Joint Multi-View Inpainting take 0.375 sec/img because 2×2 grids are used. Monodepth prediction takes 0.25 sec/img. Nerfacto (and Masked NeRF) train for 30K iterations, for a total time of 12.5 minutes. For the other methods, we start from a pre-trained Nerfacto and train for an additional 30K iters. LaMask & SD Image Cond take 1 hour to train. Inpaint DU takes 1.5 hours. Ours w/o depth takes 70 min and Ours with depth takes 75 min. For Inpaint DU and Ours, we update 40 images every 1K iters. Ours is faster than Inpaint DU because we inpaint with 2×2 grids.

5. Perceptual evaluation user study

We include supplementary videos for qualitative assessment, but a perceptual evaluation can sometimes be useful. We conduct a user study on MTurk to complement Sec. 5.2 and Table 2 of the paper. For each dataset and baseline, we asked forced-choice A/B questions. We consider the three most competitive baselines. We ask each dataset×baseline question 40 times for a total of 2400 samples. We report averages over each dataset and report results in the table. We use a consistency check to ensure high-quality responses. In all cases, our method is preferred over the baselines:

Baseline	Preference for "Ours"
LaMask	87%
SD Image Cond	88%
Inpaint + DU	59%

Table R1. **User study to select preferred videos.** The supp. file "3D_consistent_image_inpainting.mp4" contains the videos shown to users. We show the pink mask video and two options, the baseline and our video, and ask which video completes the missing region better.

We use the following repo for these experiments: `https://github.com/ethanweber/pollo`.

6. Inpainting NeRF casual captures

As mentioned in our limitations section (Sec. 6), Stable Diffusion (SD) fails to produce good inpaints when the mask distribution is similar to those produced by Bayes' Rays [2] or Nerfbusters [12]. In Fig. A1, we illustrate this. Please see the caption for details.

	PSNR ↑	SSIM ↑	LPIPS ↓
Masked NeRF	5.91	0.70	0.40
LaMask	21.13	0.93	0.23
SD Text Cond	13.27	0.70	0.40
SD Image Cond	13.71	0.75	0.31
Extended Attention	15.20	0.78	0.30
Grid Prior	14.90	0.82	0.27
Joint Multi-View Inpainting	16.59	0.85	0.24

Table A1. **2D inpainting consistency for data “chair”.**

	PSNR ↑	SSIM ↑	LPIPS ↓
Masked NeRF	5.77	0.66	0.49
LaMask	16.79	0.89	0.39
SD Text Cond	11.17	0.72	0.29
SD Image Cond	12.30	0.76	0.27
Extended Attention	13.23	0.76	0.26
Grid Prior	12.64	0.78	0.27
Joint Multi-View Inpainting	13.25	0.79	0.27

Table A2. **2D inpainting consistency for data “drums”.**

	PSNR ↑	SSIM ↑	LPIPS ↓
Masked NeRF	6.86	0.73	0.37
LaMask	18.50	0.91	0.22
SD Text Cond	13.16	0.74	0.31
SD Image Cond	13.40	0.76	0.31
Extended Attention	14.86	0.77	0.23
Grid Prior	14.35	0.81	0.28
Joint Multi-View Inpainting	17.24	0.85	0.21

Table A3. **2D inpainting consistency for data “ficus”.**

	PSNR ↑	SSIM ↑	LPIPS ↓
Masked NeRF	8.84	0.73	0.36
LaMask	21.29	0.92	0.15
SD Text Cond	12.93	0.75	0.35
SD Image Cond	15.12	0.78	0.31
Extended Attention	15.06	0.78	0.30
Grid Prior	15.62	0.84	0.24
Joint Multi-View Inpainting	16.69	0.86	0.24

Table A4. **2D inpainting consistency for data “hotdog”.**

	PSNR ↑	SSIM ↑	LPIPS ↓
Masked NeRF	8.80	0.73	0.33
LaMask	17.84	0.84	0.18
SD Text Cond	13.01	0.75	0.27
SD Image Cond	14.79	0.79	0.22
Extended Attention	15.42	0.79	0.22
Grid Prior	14.84	0.81	0.21
Joint Multi-View Inpainting	17.89	0.84	0.18

Table A5. **2D inpainting consistency for data “lego”.**

	PSNR ↑	SSIM ↑	LPIPS ↓
Masked NeRF	8.28	0.72	0.33
LaMask	18.23	0.88	0.16
SD Text Cond	12.80	0.74	0.29
SD Image Cond	14.53	0.78	0.25
Extended Attention	12.85	0.77	0.29
Grid Prior	14.25	0.80	0.22
Joint Multi-View Inpainting	14.53	0.80	0.23

Table A6. **2D inpainting consistency for data “materials”.**

	PSNR ↑	SSIM ↑	LPIPS ↓
Masked NeRF	6.94	0.73	0.35
LaMask	21.11	0.92	0.13
SD Text Cond	10.97	0.72	0.31
SD Image Cond	13.40	0.77	0.28
Extended Attention	14.26	0.78	0.27
Grid Prior	12.41	0.80	0.26
Joint Multi-View Inpainting	13.26	0.81	0.27

Table A7. **2D inpainting consistency for data “mic”.**

	PSNR ↑	SSIM ↑	LPIPS ↓
Masked NeRF	10.66	0.69	0.35
LaMask	21.76	0.82	0.18
SD Text Cond	13.14	0.72	0.31
SD Image Cond	15.95	0.74	0.29
Extended Attention	15.69	0.74	0.29
Grid Prior	16.41	0.78	0.27
Joint Multi-View Inpainting	17.64	0.80	0.24

Table A8. **2D inpainting consistency for data “ship”.**

	PSNR ↑	SSIM ↑	LPIPS ↓	MUSIQ ↑	Corrs ↑
Masked NeRF	12.45	0.81	0.30	3.82	330
LaMask	27.43	0.95	0.03	3.72	179
SD Image Cond	17.42	0.88	0.15	3.55	305
Inpaint + DU	25.42	0.95	0.06	3.96	260
Ours w/o depth	29.80	0.96	0.03	3.91	218
Ours	29.71	0.96	0.03	3.92	213

Table A9. **Quantitative NeRF baselines for data "cat".**

	PSNR ↑	SSIM ↑	LPIPS ↓	MUSIQ ↑	Corrs ↑
Masked NeRF	19.16	0.90	0.15	3.65	164
LaMask	31.66	0.96	0.03	3.66	140
SD Image Cond	21.96	0.89	0.12	3.62	182
Inpaint + DU	28.40	0.94	0.07	3.78	151
Ours w/o depth	29.76	0.94	0.06	3.65	154
Ours	29.74	0.93	0.07	3.69	146

Table A10. **Quantitative NeRF baselines for data "turtle".**

	PSNR ↑	SSIM ↑	LPIPS ↓	MUSIQ ↑	Corrs ↑
Masked NeRF	14.20	0.70	0.34	3.95	1083
LaMask	25.45	0.76	0.09	4.03	1040
SD Image Cond	24.23	0.77	0.12	4.00	1024
Inpaint + DU	25.15	0.76	0.13	3.97	1041
Ours w/o depth	26.68	0.82	0.12	4.01	1019
Ours	26.49	0.81	0.13	4.02	1038

Table A11. **Quantitative NeRF baselines for data "drawing".**

	PSNR ↑	SSIM ↑	LPIPS ↓	MUSIQ ↑	Corrs ↑
Masked NeRF	11.68	0.79	0.25	3.81	214
LaMask	24.50	0.95	0.05	4.02	195
SD Image Cond	16.45	0.89	0.12	3.66	196
Inpaint + DU	20.42	0.90	0.10	3.80	228
Ours w/o depth	28.76	0.96	0.03	3.84	211
Ours	29.37	0.96	0.03	3.86	176

Table A12. **Quantitative NeRF baselines for data "boot".**

	PSNR ↑	SSIM ↑	LPIPS ↓	MUSIQ ↑	Corrs ↑
Masked NeRF	16.86	0.93	0.10	3.82	245
LaMask	27.41	0.96	0.02	3.71	200
SD Image Cond	24.58	0.95	0.04	3.70	261
Inpaint + DU	27.92	0.95	0.03	3.70	245
Ours w/o depth	28.20	0.96	0.03	3.72	259
Ours	28.13	0.96	0.03	3.72	264

Table A13. **Quantitative NeRF baselines for data "bear".**

	PSNR ↑	SSIM ↑	LPIPS ↓	MUSIQ ↑	Corrs ↑
Masked NeRF	11.96	0.75	0.29	3.80	193
LaMask	27.73	0.97	0.04	3.84	148
SD Image Cond	19.07	0.87	0.15	3.54	235
Inpaint + DU	28.76	0.95	0.05	3.69	210
Ours w/o depth	27.75	0.95	0.04	3.63	245
Ours	27.48	0.95	0.05	3.62	232

Table A14. **Quantitative NeRF baselines for data "dumptruck".**

	PSNR ↑	SSIM ↑	LPIPS ↓	MUSIQ ↑	Corrs ↑
Masked NeRF	16.23	0.70	0.32	3.65	934
LaMask	27.25	0.86	0.09	3.84	865
SD Image Cond	21.58	0.81	0.14	3.98	852
Inpaint + DU	27.75	0.87	0.10	3.83	812
Ours w/o depth	29.54	0.90	0.07	3.65	980
Ours	29.11	0.88	0.07	3.69	1059

Table A15. **Quantitative NeRF baselines for data "norway".**

	PSNR ↑	SSIM ↑	LPIPS ↓	MUSIQ ↑	Corrs ↑
Masked NeRF	16.41	0.80	0.19	3.46	1766
LaMask	24.82	0.84	0.04	3.41	1833
SD Image Cond	22.48	0.83	0.07	3.45	1769
Inpaint + DU	24.04	0.84	0.06	3.47	1805
Ours w/o depth	24.38	0.85	0.04	3.39	1847
Ours	23.93	0.83	0.05	3.40	1924

Table A16. **Quantitative NeRF baselines for data "backpack".**

	PSNR ↑	SSIM ↑	LPIPS ↓	MUSIQ ↑	Corrs ↑
Masked NeRF	13.37	0.69	0.37	3.50	981
LaMask	25.63	0.84	0.11	3.59	1053
SD Image Cond	24.31	0.83	0.12	3.66	1023
Inpaint + DU	27.13	0.83	0.13	3.69	1078
Ours w/o depth	28.87	0.88	0.09	3.66	1120
Ours	28.78	0.88	0.09	3.66	1137

Table A17. **Quantitative NeRF baselines for data "billiards".**

	PSNR ↑	SSIM ↑	LPIPS ↓	MUSIQ ↑	Corrs ↑
Masked NeRF	14.74	0.75	0.31	3.65	839
LaMask	32.02	0.93	0.03	3.80	779
SD Image Cond	28.26	0.90	0.07	3.64	799
Inpaint + DU	31.05	0.92	0.05	3.70	766
Ours w/o depth	30.35	0.93	0.05	3.68	764
Ours	30.08	0.93	0.05	3.69	768

Table A18. **Quantitative NeRF baselines for data "office".**

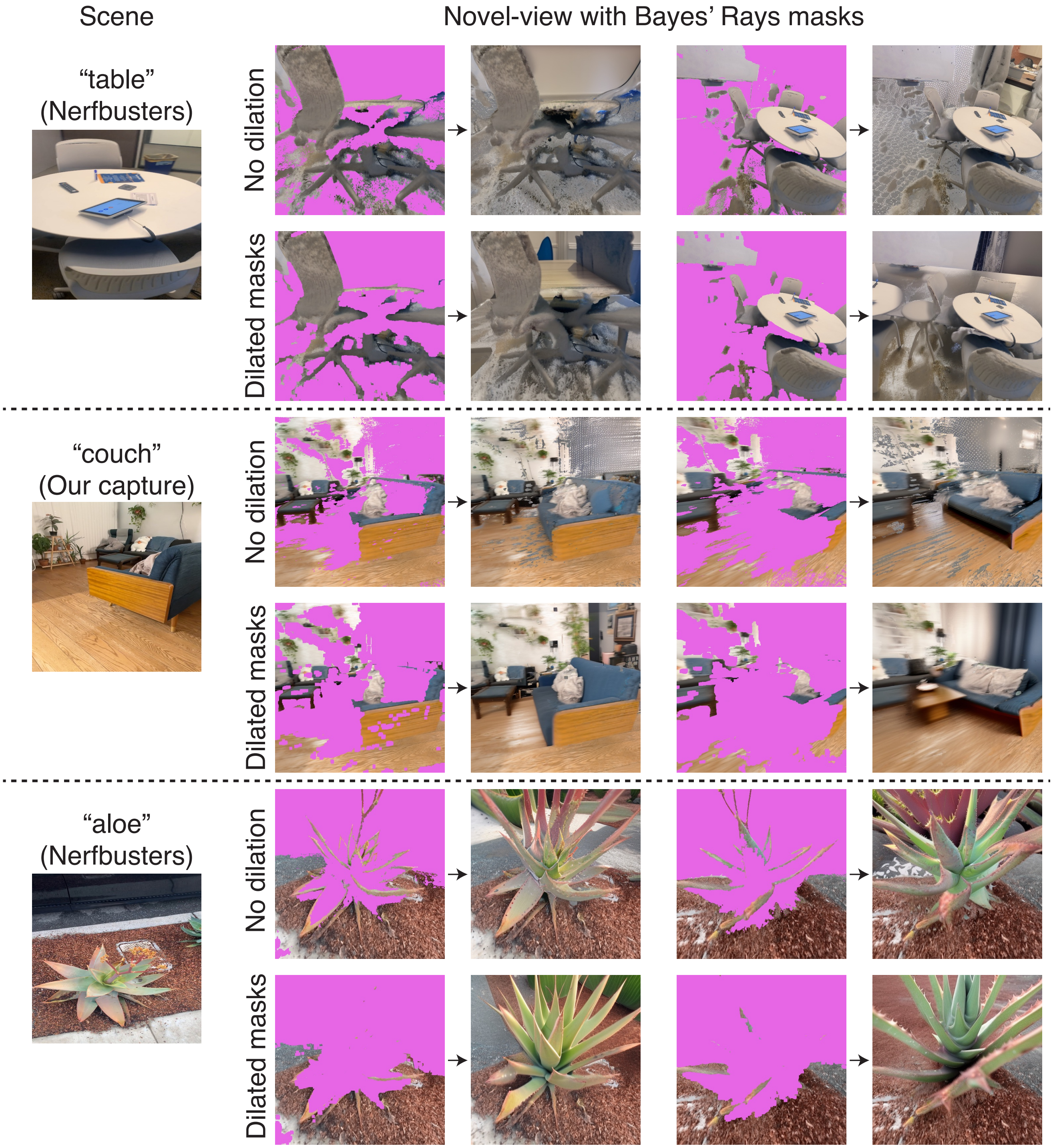

Figure A1. **Out-of-distribution inpaints from Stable Diffusion when applied to casual captures.** On the left, we show a training image from a casually captured scene. On the right, we show masks obtained by rendering a novel-view that has occlusions, and then running Bayes' Rays [2] to delete areas with high uncertainty (marked in pink). We then inpaint these regions with image conditioning. When the mask is not dilated (top rows), the inpaints have many artifacts such as ripple patterns and gray stretches. When the masks are dilated (bottom rows), the inpaints get slightly better but are no longer consistent with the known parts of the scene. This is a challenging setting to address in future work. Retraining SD with masks of this distribution could alleviate the problem, but this is costly and out-of-scope of using an *off-the-shelf* model, as done in our work.

References

[1] Tim Brooks, Aleksander Holynski, and Alexei A. Efros. Instructpix2pix: Learning to follow image editing instructions. In *CVPR*, 2023. 2

[2] Lily Goli, Cody Reading, Silvia Sellán, Alec Jacobson, and Andrea Tagliasacchi. Bayes' Rays: Uncertainty quantification in neural radiance fields. In *arXiv*, 2023. 1, 3, 6

[3] Ayaan Haque, Matthew Tancik, Alexei Efros, Aleksander Holynski, and Angjoo Kanazawa. Instruct-nerf2nerf: Editing 3d scenes with instructions. In *ICCV*, 2023. 2

[4] Junjie Ke, Qifei Wang, Yilin Wang, Peyman Milanfar, and Feng Yang. Musiq: Multi-scale image quality transformer. In *ICCV*, 2021. 2

[5] Ben Mildenhall, Pratul P. Srinivasan, Rodrigo Ortiz-Cayon, Nima Khademi Kalantari, Ravi Ramamoorthi, Ren Ng, and Abhishek Kar. Local light field fusion: Practical view synthesis with prescriptive sampling guidelines. In *SIGGRAPH*, 2019. 2

[6] Ben Mildenhall, Pratul P Srinivasan, Matthew Tancik, Jonathan T Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. In *ECCV*, 2020. 1

[7] Ashkan Mirzaei, Tristan Aumentado-Armstrong, Marcus A. Brubaker, Jonathan Kelly, Alex Levinshtein, Konstantinos G. Derpanis, and Igor Gilitschenski. Reference-guided controllable inpainting of neural radiance fields. In *ICCV*, 2023. 2

[8] Ashkan Mirzaei, Tristan Aumentado-Armstrong, Konstantinos G. Derpanis, Jonathan Kelly, Marcus A. Brubaker, Igor Gilitschenski, and Alex Levinshtein. SPIn-NeRF: Multiview segmentation and perceptual inpainting with neural radiance fields. In *CVPR*, 2023. 2

[9] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *CVPR*, 2022. 2

[10] Jiaming Sun, Zehong Shen, Yuang Wang, Hujun Bao, and Xiaowei Zhou. Loftr: Detector-free local feature matching with transformers. In *CVPR*, 2021. 2

[11] Matthew Tancik, Ethan Weber, Evonne Ng, Ruilong Li, Brent Yi, Terrance Wang, Alexander Kristoffersen, Jake Austin, Kamyar Salahi, Abhik Ahuja, et al. Nerfstudio: A modular framework for neural radiance field development. In *SIGGRAPH Conference Proceedings*, 2023. 2

[12] Frederik Warburg*, Ethan Weber*, Matthew Tancik, Aleksander Hołyński, and Angjoo Kanazawa. Nerfbusters: Removing ghostly artifacts from casually captured nerfs. In *ICCV*, 2023. 1, 3

[13] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *CVPR*, 2018. 2