SIGNeRF: Scene Integrated Generation for Neural Radiance Fields

Supplementary Material

This appendix offers supplementary information on SIGNeRF. It delves into more intricate aspects of our implementation (Sec. A), presents insights into the emergence of multiview consistency (Sec. B), outlines the specialized tools we have engineered (Sec. C), and elaborates on the generated datasets (Sec. D).

Additional Material We wish to emphasize the inclusion of two videos alongside this paper. Given the three-dimensional nature of our results, these videos serve as the most effective medium for their evaluation. We strongly encourage viewing:

- Explanation Video — A comprehensive guide through our methods and pipeline.
- Results Video — A presentation of our results, including comparative analyses.

A. Implementation Details

Image Diffusion In our approach, we integrate an inpainting version of ControlNet [15] with Stable Diffusion XL [8] to synthesize images based on the mask and depth map inputs. We have customized the implementation of the publicly available SD WebUI API [1], which builds upon the Diffusers library [13]. Notably, SIGNeRF is a general approach that also works with previous versions of Stable Diffusion. For example, the 'field' scene with the generated cows (Fig. 3), which was generated with Stable Diffusion 1.5 [9] and ControlNetInpaint [10].

Training SIGNeRF necessitates a pre-trained NeRF scene, which we acquire by employing the nerfacto model from Nerfstudio [12], undergoing $30,000$ iterations of training for each scene. Subsequently, the SIGNeRF pipeline for scene-integrated generation is utilized to create an edited dataset for the NeRF scene (Sec. 3.3). It was discovered that for optimal efficacy, a reference sheet comprising 5 images strikes an effective balance between preserving image quality and providing adequate scene context. Following the creation of the edited dataset, we opt to either fine-tune the existing NeRF scene with the updated dataset or initiate training of a new NeRF scene, depending on the choice of selection method. We observe better results for object generation when training the NeRF scene from scratch. Conversely, for the task involving generative editing, fine-tuning the pre-trained NeRF scene has shown to be more effective. In such cases, optimizers are re-initialized, and the LPIPS [16] loss is applied to enhance scene consistency [5].

Generation	Denoising S.	ControlNet S.	Guidance S.
Person - Sport	0.9	0.4	7.0
Person - Pirate	[0.95, 0.5]	[0.4, 0.8]	7.5
Person - Batman	[0.95, 0.5]	[0.4, 1.0]	6.0
Plushy - Ironman	0.9	1.0	7.0
Plushy - Tiger	0.6	1.0	7.0
Plushy - Gold	0.95	1.0	7.0
Bear - Grizzly	0.9	0.95	7.0
Bear - Polar	0.9	0.95	7.0
Bear - Panda	0.9	0.95	7.0
Bear - Rabbit	0.95	1.0	7.0
Field - Cow	(SD 1.5)	0.7	7.5
Urban - House	1.0	0.8	7.0

Table A1. **Scene List** – Parameters used for each scene for the image diffusion model, with denoising strength, ControlNet scale, and guidance scale. Rows with multiple values indicate the need for a second iteration, as discussed in Sec. 3.3. For the 'field' scene, we used Stable Diffusion 1.5 [9], which does not have a denoising strength parameter.

Generation Information The generational edit of the NeRF scene can be controlled by several parameters provided by the image diffusion model, such as the denoising strength, the ControlNet condition and the guidance scale. We provide the parameters used for each scene in Tbl. A1.

Metrics For the quantitative evaluation of our results (Sec. 4.4), we use two metrics, the first one is the CLIP text-to-image similarity score [4] and the second one is a new background preservation metric. This background preservation metric measures the background difference between a render $\boldsymbol{I}_{\text{org}}$ from the original NeRF and a render $\boldsymbol{I}_{\text{edit}}$ from the edited NeRF. We use the corresponding mask $\boldsymbol{M}$ provided by the picked selection method (Sec. 3) and compute the background preservation as:

$$\text{PSNR}_{\text{bg}} = \text{PSNR}(\boldsymbol{I}_{\text{org}} \cdot (\neg \boldsymbol{M}), \boldsymbol{I}_{\text{edit}} \cdot (\neg \boldsymbol{M})) \quad (1)$$

$$\text{SSIM}_{\text{bg}} = \text{SSIM}(\boldsymbol{I}_{\text{org}} \cdot (\neg \boldsymbol{M}), \boldsymbol{I}_{\text{edit}} \cdot (\neg \boldsymbol{M})) \quad (2)$$

B. Multiview Consistency

We want to clarify that ControlNet [15] was not fine-tuned in any way. We found the multi-view consistency property

Figure B1. **Inpainting Ablation & LoFTR Features** – grid inpainting generation capabilities with & w/o ControlNet under certain denoising levels (label top right). LoFTR keypoint correspondences reflect the consistency between the sub images (last row). Zoom in for details.

empirically by exploration. Initially, combining two images, and later discovering how to reach greater consistency using a grid. The reason why diffusion models show this multi-view consistency isn't entirely clear, but we have some hypotheses. The grid-based inpainting can achieve a degree of multi-view consistency without ControlNet, although the results become less consistent at higher denoising levels (Fig. B1), therefore only allowing subtle edits. This suggests that StableDiffusion [9] inherently supports some level of multi-view consistency and 3D understanding as also shown by [3, 14]. We assume that attention plays a crucial role, further enforced by the inpainting mechanism focusing on the relevant parts and ignoring scene elements by masking. The depth condition in ControlNet further acts as a reference for the model to infer orientation/shape/scale, even for high denoising levels as the condition is not noised. Fine-tuning to achieve even better results is interesting future work, but we think that one of the most attractive properties of our approach is the possibility of achieving high-quality generation with off-the-shelf image diffusion models.

Further there is no direct measurement of the quality of the multiview consistency within the reference sheet. This is also nontrivial to measure, else we could optimize for it in an self supervised manner. We experimented with LoFTR [11] correspondences between original, reference-sheet generated and independent generated images. Averaged results show that independent generation has 70% less correspondences than the original images, while the reference sheet generated images achieve 15% less correspondences. Nevertheless, we believe the supplementary videos we provided substantiate our claims of achieving multiview consistency. The feasibility of conducting 3D reconstructions from the generated images serves as a testament to this consistency. It is important to acknowledge that while the images may not exhibit perfect consistency, the underlying NeRF counteracts that by aligning 3D consistent regions and discarding outliers.

Figure C2. **Viewer** – Outlining the placement of a proxy object within an existing NeRF scene. The viewer is divided into a 3D viewport at the center, a scene and selection column on the left and a selection specific control column on the right.

C. Viewer

Our development includes a NeRF viewer designed to examine the trained NeRF scenes and facilitate the placement of proxy objects or the selection of scene elements for modification. The viewer's backend is based on Nerfstudio [12], while the frontend is entirely novel, incorporating a suite of advanced features. A depiction of the viewer interface is presented in Fig. C2. Constructed as a React [7] application, it leverages React Three Fiber [6], a React renderer for Three.js [2], to facilitate 3D scene manipulation. The Nerfstudio backend streams the NeRF data to the frontend, where it is displayed centrally in the viewport. We have augmented the backend to transmit not just the scene render but also the corresponding depth map. This data enables us to utilize a shader that merges the NeRF render with Three.js's native rendering, allowing for the visualization of occluded elements within the NeRF scene in real-time.

Enhancements to the viewer include a scene hierarchy and intuitive manipulation tools, which simplify the process of placing proxy objects or delineating bounding boxes to select regions of interest. Additionally, a bridge to SIGNeRF was written, such that we can generate the reference sheet and dataset with the information provided by the viewer. Consequently, the viewer serves as a user-friendly platform for performing scene-integrated generation on NeRF scenes. We plan to release this viewer as open-source software, providing the community with a powerful tool for NeRF scene editing.

D. Dataset Generation

SIGNeRF generates an updated NeRF image dataset with the provided reference sheet to edit a NeRF scene. Our generative pipeline (Fig. 2) demonstrates this with a selection of modified dataset images, but a completely revised dataset is not showcased. In Fig. D3, we exhibit a fully edited dataset

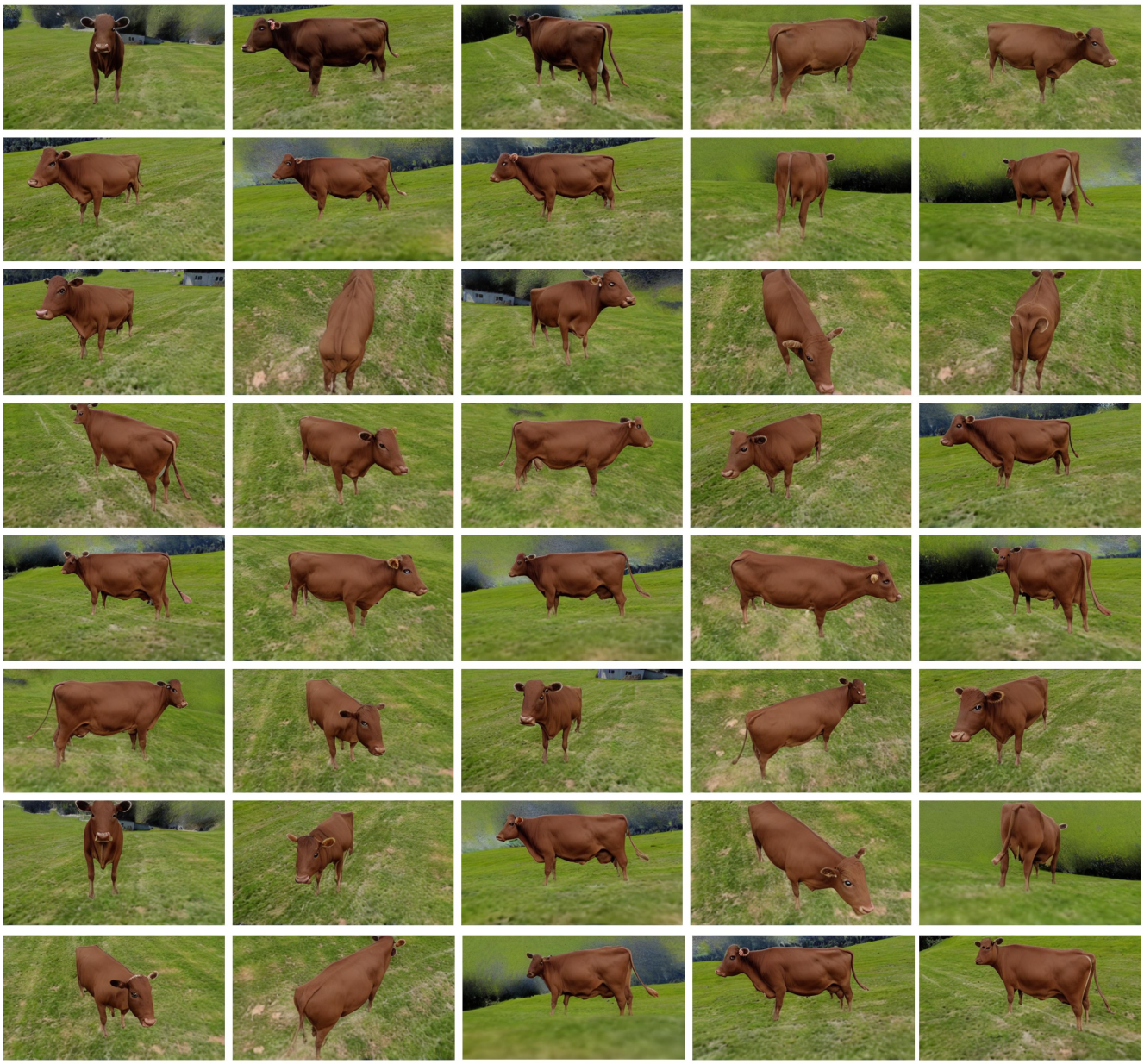

Figure D3. **Dataset Generation** – Illustrating the consistency within edited views of an original NeRF dataset. The first row shows images from the reference sheet, the rest are the views generated with the reference sheet.

of the 'field' scene (Fig. 3). Consistency is maintained across the generated images; however, some outliers might occur. These inconsistencies are minor over the dataset and are generally resolved by the NeRF optimization process, which learns to ignore such outliers. For complex scenes, the use of an LPIPS [16] loss is beneficial to further improve the consistency during the optimization.

References

[1] AUTOMATIC1111. Stable diffusion webui. https://github.com/AUTOMATIC1111/stable-diffusion-webui, 2022. 1

[2] Ricardo Cabello. *Three.js*, 2010. 2

[3] Yida Chen, Fernanda Viégas, and Martin Wattenberg. Beyond surface statistics: Scene representations in a latent diffusion model. *arXiv preprint arXiv:2306.05720*, 2023. 2

[4] Rinon Gal, Or Patashnik, Haggai Maron, Gal Chechik, and Daniel Cohen-Or. Stylegan-nada: Clip-guided domain adap-

tation of image generators. *ACM Transactions on Graphics (TOG)*, 41(4):1–13, 2021. 1

[5] Ayaan Haque, Matthew Tancik, Alexei A. Efros, Aleksander Holynski, and Angjoo Kanazawa. Instruct-nerf2nerf: Editing 3d scenes with instructions. 2023. 1

[6] Paul Henschel. *React three fiber*, 2019. 2

[7] Meta. *React*, 2013. 2

[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. *arXiv preprint arXiv:2307.01952*, 2023. 1

[9] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. pages 10684–10695, 2021. 1, 2

[10] Neelay Shah, Tommaso De Rossi, and Mikolaj Czerkawski. Controlnetinpaint: Inpaint images with controlnet. https://github.com/mikonvergence/ControlNetInpaint, 2023. GitHub repository. 1

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

[12] Matthew Tancik, Ethan Weber, Evonne Ng, Ruilong Li, Brent Yi, Justin Kerr, Terrance Wang, Alexander Kristoffersen, Jake Austin, Kamyar Salahi, Abhik Ahuja, David McAllister, and Angjoo Kanazawa. Nerfstudio: A modular framework for neural radiance field development. *arXiv preprint arXiv:2302.04264*, 2023. 1, 2

[13] Patrick von Platen, Suraj Patil, Anton Lozhkov, Pedro Cuenca, Nathan Lambert, Kashif Rasul, Mishig Davaadorj, and Thomas Wolf. Diffusers: State-of-the-art diffusion models. https://github.com/huggingface/diffusers, 2022. 1

[14] Guanqi Zhan, Chuanxia Zheng, Weidi Xie, and Andrew Zisserman. What does stable diffusion know about the 3d scene? *arXiv preprint arXiv:2310.06836*, 2023. 2

[15] Lvmin Zhang and Maneesh Agrawala. Adding conditional control to text-to-image diffusion models. *arXiv preprint arXiv:2302.05543*, 2023. 1

[16] Richard Zhang, Phillip Isola, Alexei A. Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. pages 586–595, 2018. 1, 3