-Supplementary Material-
Segment Anything in 3D with NeRFs

Anonymous Author(s)
Affiliation
Address
email

A Appendix

In this appendix, we provide the following:

1. Implementation details (Section A.1);
2. A segmentation refinement strategy (Section A.2);
3. The scribble to point strategy for the evaluation of NVOS (Section A.3);
4. An analysis about vanilla NeRF [5] used in SA3D (Section A.4);
5. More information about the object filtering of Replica [8] (Section A.5);
6. A further illustration of the self-prompting strategy (Section A.6);
7. More visualization results with different kinds of input prompts (Section A.7).

A.1 Implementation Details

We implement SA3D using PyTorch [6] with reference to the code provided by DVGOv2 [9]. The SA3D model is built and trained on a single Nvidia Geforce RTX3090 GPU. For our NeRF model, we primarily employ TensoRF [2], utilizing the VM-48 representation to store the radiance latent vectors. The radiance fields are pre-trained for most datasets with 40,000 iterations. For the LLFF dataset [4] and the 360 dataset [1], the radiance fields are trained with 20,000 iterations.

A.2 Refinement with A Two-pass Segmentation Mechanism

SAM may produce segmentation masks containing undesired parts. The IoU-aware view rejection is hard to handle this issue when the mis-classified region gradually expands.

We propose a two-pass segmentation mechanism to further refine the segmentation result. After completing 3D segmentation introduced in the main manuscript, we get a 3D mask $\mathbf{V}$. To detect the mis-classified region from $\mathbf{V}$, we re-render the 2D segmentation mask $\mathbf{M}^u$ of the user-specific reference view and compare it with the original SAM segmentation result $\mathbf{M}^u_{\texttt{SAM}}$.

Subsequently, we reset the original 3D mask $\mathbf{V}$ to be a zero tensor and introduce another 3D mask $\mathbf{V}' \in \mathbb{R}^3$ that specifically indicates the mis-classified regions. The 3D segmentation process is then repeated, with the key difference being the incorporation of negative prompt points during the self-prompting phase. In other words, the prompts obtained from $\mathbf{V}'$ serve as negative prompts for $\mathbf{V}$, and vice versa. This incorporation of negative prompts enables SAM to gain a better understanding of the user's requirements and refine the segmentation accordingly (shown in Figure A1). It is important to note that while this two-pass segmentation mechanism holds promise, it was not utilized in our main experiments due to considerations of efficiency.

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

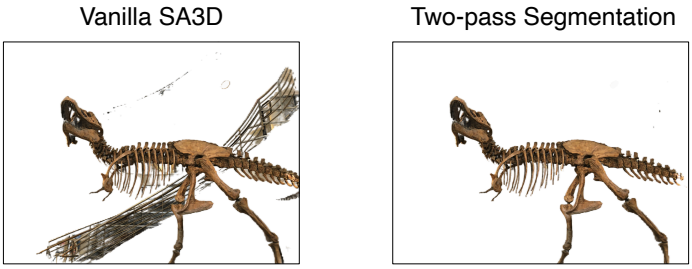

Figure A1: The effect of the two-pass segmentation refinement.

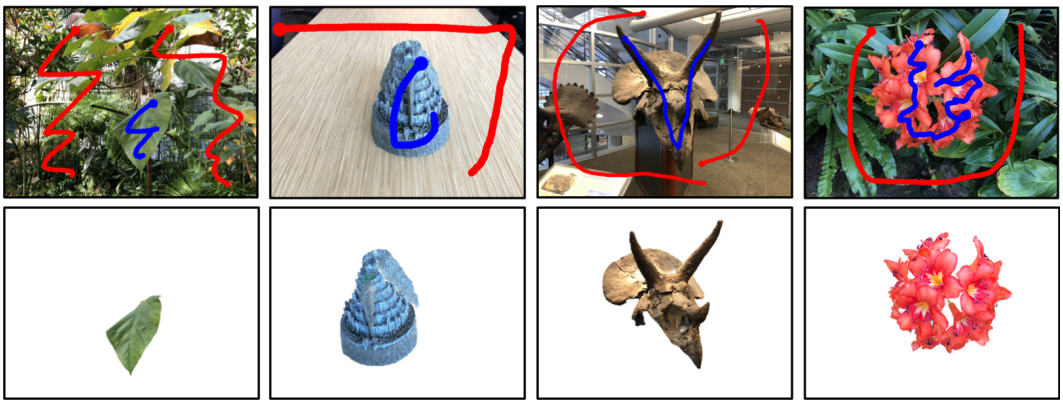

Figure A2: Some scribbles on the reference views provided by NVOS and the corresponding segmentation results of SA3D. Blue scribbles are positive and red scribbles are negative.

A.3 The Scribble to Points Strategy for The Evaluation of NVOS

The NVOS [7] dataset provides a reference view and the corresponding scribbles for each scene (shown in Figure A2). In practice, since the scribbles usually contain tens of thousands of dense points, SAM [3] cannot directy take such scribbles as input. The abundance of points in the scribbles hinders SAM's performance when directly utilizing them as prompts, which is an inherent limitation of SAM.

For fair comparison, we extract positive and negative prompt points from the provided positive and negative scribbles, respectively. For input scribbles, we first skeletonize them and then select 2% points from the skeletonized positive scribbles as the positive prompts and 0.5% points from the skeletonized negative scribbles as the negative prompts.

A.4 The Effect of Different NeRFs Used in SA3D

We adapt SA3D to the vanilla NeRF [5] to showcase its generalizability. We present visualization results on the LLFF dataset. As illustrated in Figure A3, SA3D with the vanilla NeRF exhibits excellent performance without the need for additional modifications.

A.5 Object Filtering for The Replica Dataset

The Replica dataset contains many objects in each scene. However, it is important to note that many of these objects exhibit low quality, as depicted in Figure A4, making them unsuitable for evaluating 3D segmentation. Generally, these instances exhibit the following issues: some instances are not present in the training frames provided by Zhi *et al.* [10]; some instances are too small to be effectively segmented, such as thin slits in doors; and some instances consist of unrecognizable, low-resolution pixels, such as blurred tags, which are not suitable for accurate instance segmentation. Accordingly, we carefully select approximately 20 representative instances from each scene for the evaluation. The list of instance IDs for each scene can be found in Table A2. We have also included

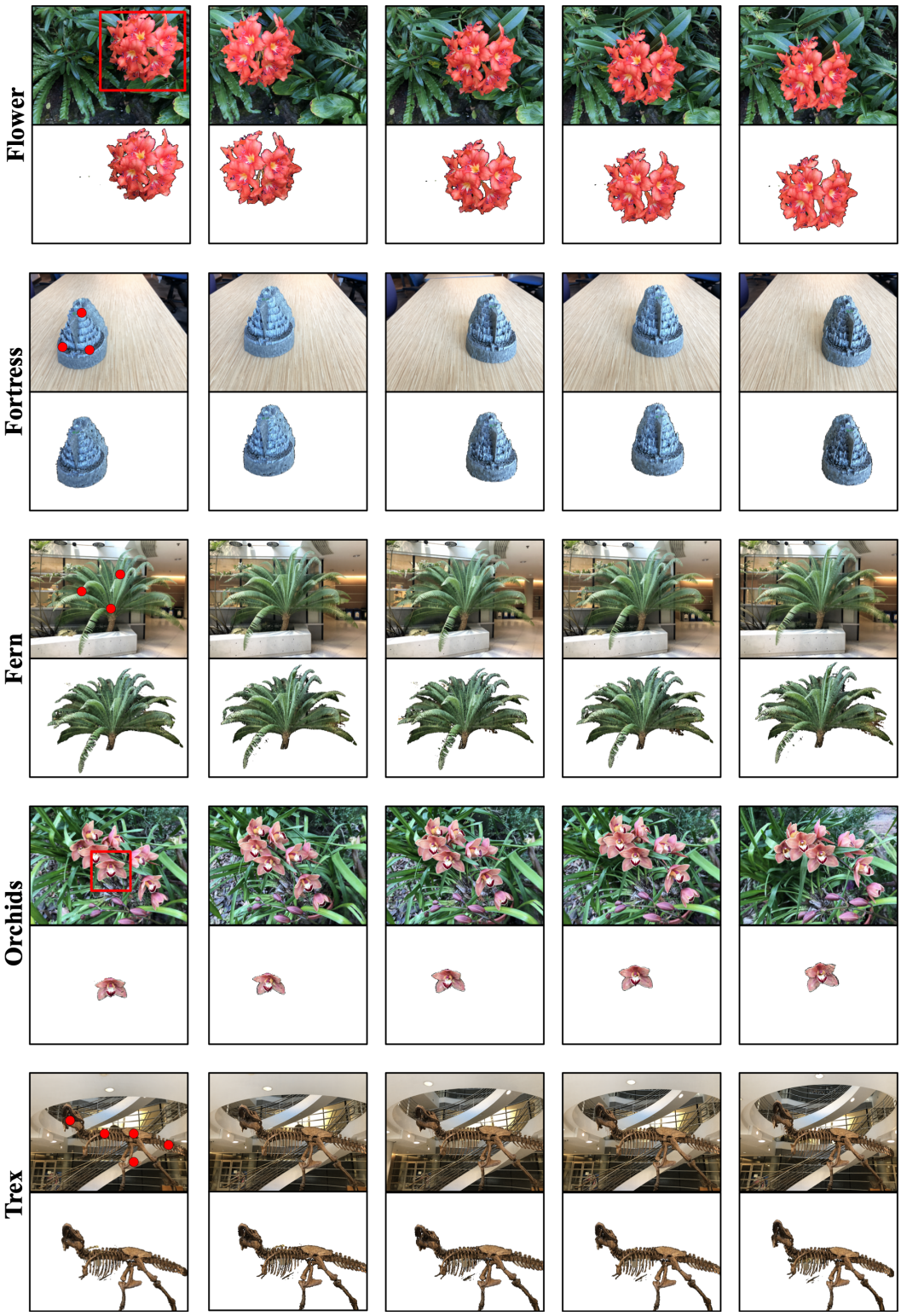

Figure A3: 3D Segmentation results based on vanilla NeRF on the LLFF dataset.

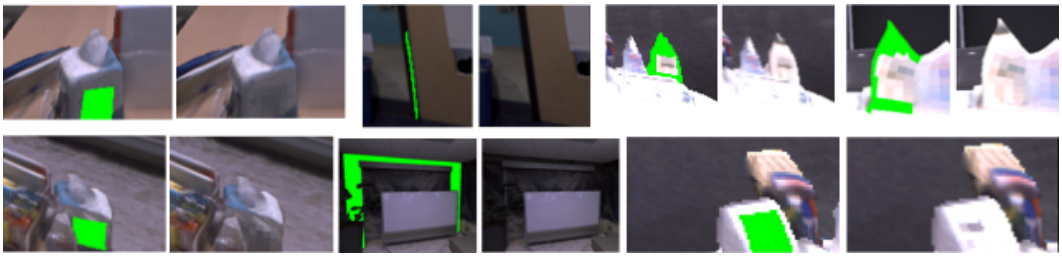

Figure A4: Some ground-truth masks (shown in green) and their corresponding instances from the Replica dataset. These unreasonable segmentation targets are filtered out in evaluation.

the quantitative results without object filtering in Table A1. Even without object filtering, SA3D demonstrates improvements compared to the single-view baseline.

Table A1: Quantitative results on Replica (mIoU) without object filtering.

Scenes	office0	office1	office2	office3	office4	room0	room1	room2	mean
Single view	58.8	53.1	61.2	54.2	56.7	51.0	58.6	58.3	56.49
SA3D (ours)	**65.1**	**59.8**	**69.7**	**61.4**	**63.8**	**56.8**	**68.4**	**72.2**	**64.65**

Table A2: The selected id lists of Replica.

Scenes	ID list
office0	3,4,7,8,9,10,12,14,17,19,21,23,26,28,29,30,36,37,40,42,44,46,54,55,57,58,61
office1	3,7,9,11,13,14,15,17,23,24,29,32,33,36,37,39,42,44,45,46
office2	0,2,8,9,13,14,17,19,23,27,40,41,47,49,51,54,58,60,65,67,70,71,72,73,78,85,90,92,93
office3	3,8,11,14,15,18,19,25,29,30,32,33,38,39,43,51,54,55,61,65,72, 76,78,82,87,91,95,96, 101,111
office4	1,2,6,7,9,11,17,22,23,26,33,34,39,47,49,51,52,53,55,56
room0	5,6,7,10,13,14,16,25,32,33,35,46,51,53,55,60,64,67,68,83,86,87,92
room1	1,2,4,6,7,9,10,11,16,18,24,28,32,36,37,44,48,52,54,56
room2	3,5,6,7,8,9,11,12,16,18,22,26,27,37,38,39,40,43,49,55,56

A.6 An Illustration for The Proposed Self-prompting Strategy

We offer an illustration (Figure A5) to assist readers in gaining a clearer understanding of the self-prompting strategy.

In the self-prompting strategy, prompt points $\mathcal{P}_s$ are derived from an incomplete 2D rendered mask $\mathbf{M}^{(n)}$, which is represented as a confidence score map. Initially, the selected prompt points set $\mathcal{P}_s$ is empty, and the first prompt point $\mathbf{p}_0$ is selected as the one with the highest confidence score in the mask $\mathbf{M}^{(n)}$. For subsequent points, square regions centered around existing prompt points are masked out on $\mathbf{M}^{(n)}$. The depth $z(\mathbf{p})$, estimated by the pre-trained NeRF, helps convert 2D pixel $\mathbf{p}$ into a 3D point $\mathcal{G}(\mathbf{p})$. The new prompt point is expected to have a high confidence score while being close to existing prompt points. Hence, a distance-aware decay term is introduced to compute the confidence score. The remaining point with the highest decayed score is added to the prompt set. This selection process is repeated until either the number of prompts $|\mathcal{P}_s|$ reaches a predefined threshold n_p or the maximum value of the remaining points is less than 0. Please refer to Section 3.4 of the main manuscript for more details.

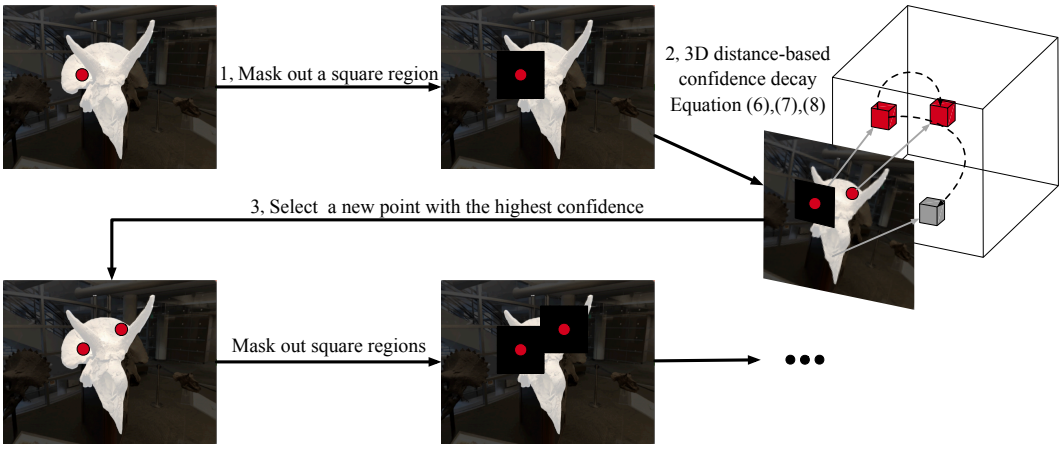

Figure A5: An illustration of the self-prompting strategy.

A.7 More Visualization Results

We present additional visualization results in Figure A6 and Figure A7, showcasing the effectiveness of SA3D across various input prompts. Additionally, we provide rendered videos (see "SA3D_visualization.mp4") that showcase the segmented 3D objects.

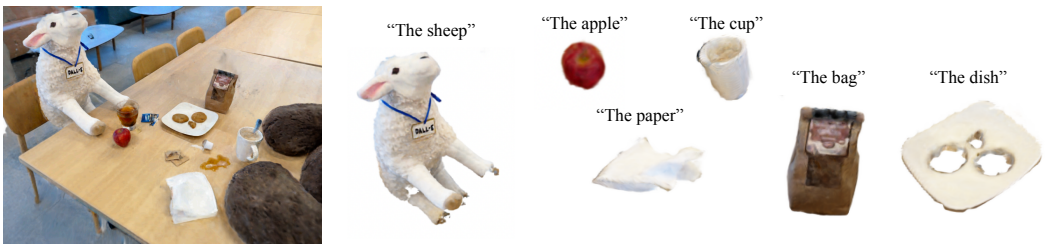

Figure A6: Text prompt based visualization results on the LERF figurines dataset.

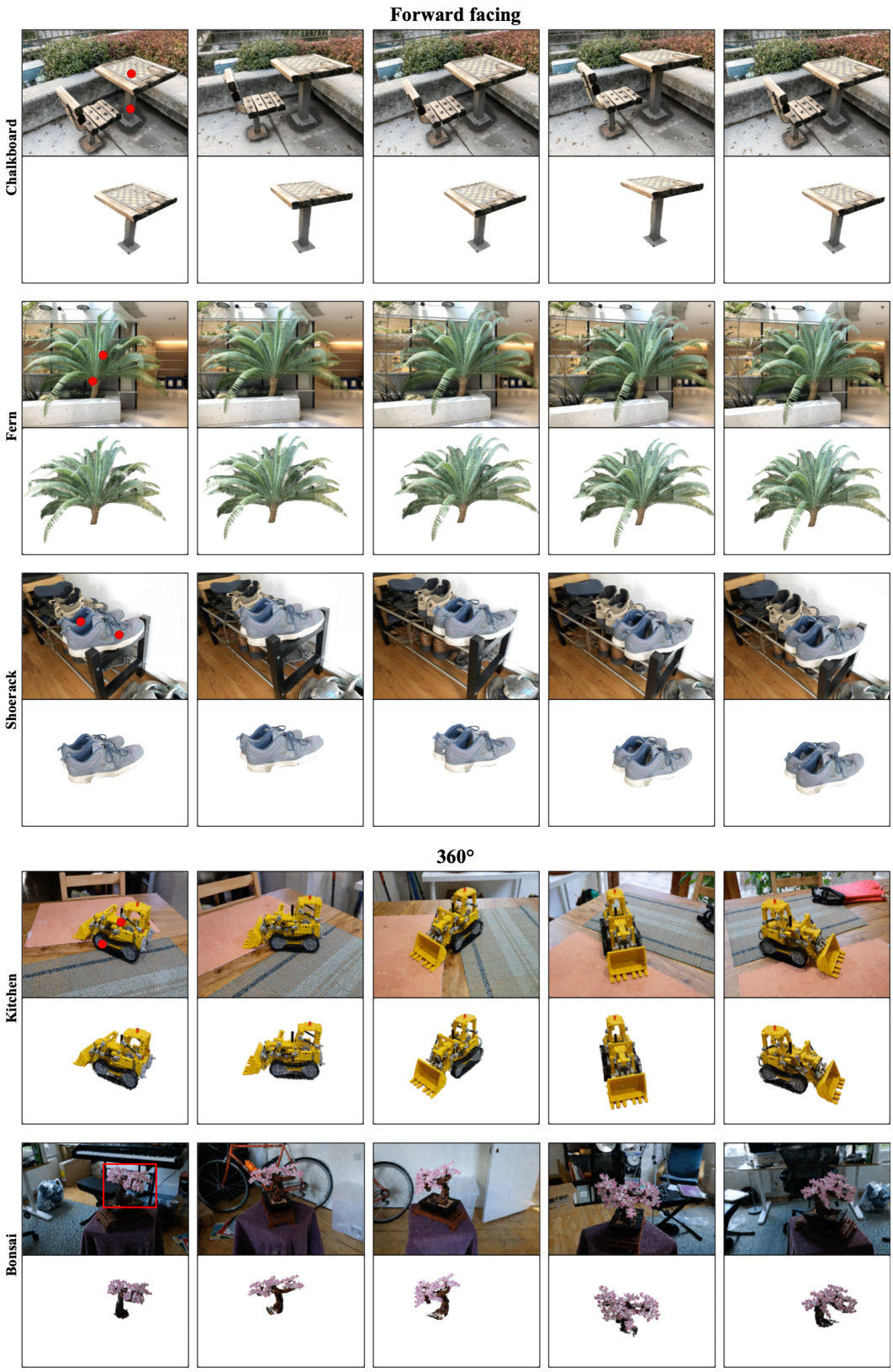

Figure A7: More visualization results on the LLFF dataset and the 360 dataset (based on point and box prompts).

References

[1] Jonathan T. Barron, Ben Mildenhall, Dor Verbin, Pratul P. Srinivasan, and Peter Hedman. Mip-nerf 360: Unbounded anti-aliased neural radiance fields. In *CVPR*, 2022.

[2] Anpei Chen, Zexiang Xu, Andreas Geiger, Jingyi Yu, and Hao Su. Tensorf: Tensorial radiance fields. In *ECCV*, 2022.

[3] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C Berg, Wan-Yen Lo, et al. Segment anything. *arXiv preprint arXiv:2304.02643*, 2023.

[4] 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. *ACM Trans. Graph.*, 2019.

[5] 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.

[6] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Köpf, Edward Z. Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In *NeurIPS*, 2019.

[7] Zhongzheng Ren, Aseem Agarwala, Bryan C. Russell, Alexander G. Schwing, and Oliver Wang. Neural volumetric object selection. In *CVPR*, 2022.

[8] Julian Straub, Thomas Whelan, Lingni Ma, Yufan Chen, Erik Wijmans, Simon Green, Jakob J Engel, Raul Mur-Artal, Carl Ren, Shobhit Verma, et al. The replica dataset: A digital replica of indoor spaces. *arXiv preprint arXiv:1906.05797*, 2019.

[9] Cheng Sun, Min Sun, and Hwann-Tzong Chen. Improved direct voxel grid optimization for radiance fields reconstruction. *arXiv preprint arXiv:2212.13545*, 2022.

[10] Shuaifeng Zhi, Tristan Laidlow, Stefan Leutenegger, and Andrew J. Davison. In-place scene labelling and understanding with implicit scene representation. In *ICCV*, 2021.