Supplementary Materials for "NeRFPrior: Learning Neural Radiance Field as a Prior for Indoor Scene Reconstruction"

Wenyuan Zhang[1], Emily Yue-ting Jia[1], Junsheng Zhou[1], Baorui Ma[1], Kanle Shi[2],
Yu-Shen Liu[1*], Zhizhong Han[3]
School of Software, Tsinghua University, Beijing, China[1]
Kuaishou Technology, Beijing, China[2]
Department of Computer Science, Wayne State University, Detroit, USA[3]
zhangwen21@mails.tsinghua.edu.cn, jiaemily120@gmail.com, zhou-js24@mails.tsinghua.edu.cn
mabaorui2014@gmail.com, shikanle@kuaishou.com, liuyushen@tsinghua.edu.cn, h312h@wayne.edu

1. Additional Comparison Results

Table 1. Additional results on ScanNet dataset bewteen Neuralangelo, Go-Surf and our method.

Methods	CD ↓	NC ↑	Prec ↑	Recall ↑	F1 ↑
NeuralAngelo[4]	0.245	0.272	0.274	0.311	0.292
Ours	**0.133**	**0.120**	**0.439**	**0.429**	**0.433**
GO-Surf[9]	0.048	0.021	0.880	0.894	0.887
Ours (+depth)	**0.027**	**0.020**	**0.931**	**0.928**	**0.930**

1.1. Comparing with Neuralangelo and GO-Surf

To demonstrate the priority of our method, we further compare our method with Neuralangelo [4] on ScanNet dataset, as shown in Table 1 and Fig. 1. Neuralangelo is a state-of-the-art reconstruction method which combines SDF optimization from NeuS [10] and hash encoding from Instant-NGP [7]. However, it shows poor performance on ScanNet dataset, because it is difficult for Neuralangelo to optimize the multiscale feature grids in indoor scenes. Additionally, it takes a long time (about 16 hours) for Neuralangelo to optimize a single scene.

Our method is able to reconstruct high-quality surfaces without additional supervision from other datasets. On the other hand, given stronger prior constraints, our method is able to achieve better performance. To verify this, we evaluate our method on the condition of ground truth depth supervision on ScanNet dataset and compare our method with GO-Surf [9], which uses RGBD data to reconstruct indoor scenes, as shown in Table 1 and Fig. 1. Our method outperforms Go-Surf under all metrics, which further justifies the superiority of our method. Visual comparisons in Fig. 1 show that our method is able to reconstruct more complete and smooth surfaces and captures more scene details. GO-Surf can only reconstruct the surfaces which are visible in the training views, while seriously degenerates at the invisible areas. Our method can complete the invisible areas and reconstruct more consistent and smooth surfaces, thus achieving better visual effect.

1.2. Comparison on Object-centered Scenes

Previous methods like NeuS [10] and HF-NeuS [11] work well with object-centered scenes, but struggle to reconstruct plausible geometry in indoor scenes, unless depth or other priors are provided. Indoor scenes contain complex topologies, various types of objects with different scales, and lack densely surrounded perspectives, which makes the reconstruction task challenging. Therefore in this work, we focus on reconstructing indoor scenes and propose NeRFPrior. It gives both density and color priors, provides a novel perspective for indoor scene reconstruction and does not have a generalization issue. We further justify that our method can extend to object-centered scenes such as DTU and BlendedMVS. The comparison results on the two datasets are shown in Tab. 2 and Fig. 2.

Table 2. Numerical comparison of Chamfer Distance (CD) between SOTAs and our method on DTU dataset.

Scan	24	37	40	55	63	97	110	Mean
NeuS[10]	1.37	1.21	0.73	0.40	1.20	1.16	1.69	1.11
HF-NeuS[11]	0.76	1.32	0.70	**0.39**	**1.06**	1.12	1.22	0.94
MonoSDF	1.04	1.16	0.71	0.42	1.25	1.18	1.58	1.05
Ours	**0.68**	**1.10**	**0.55**	0.44	1.15	**1.05**	**1.19**	**0.88**

1.3. More Visualization Comparison

We provide additional visualization comparisons between our method and baselines on ScanNet dataset, BlendSwap

[*] The corresponding author is Yu-Shen Liu. This work was partially supported by Deep Earth Probe and Mineral Resources Exploration—National Science and Technology Major Project (2024ZD1003405), and the National Natural Science Foundation of China (62272263).

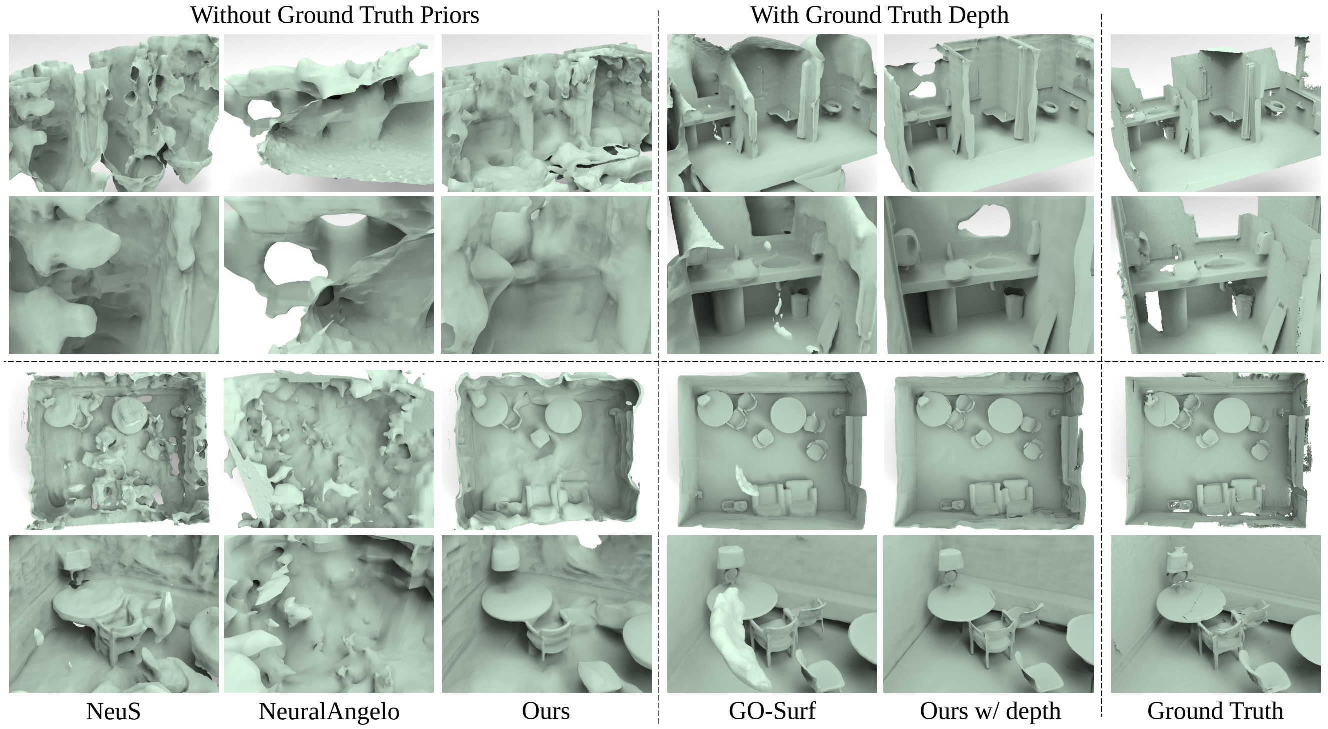


Figure 1. Visualization Comparison on ScanNet Dataset.

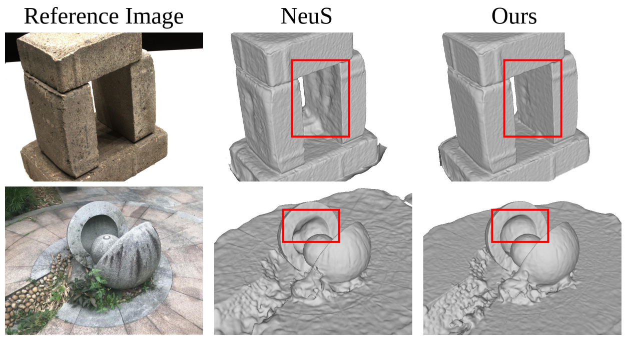


Figure 2. Visual comparison on DTU and BlendedMVS.

dataset and Replica dataset, as shown in Fig. 1, Fig. 3 and Fig. 4, separately. Our method significantly outperforms existing methods without data-driven priors, and achieves high accuracy. Note that we didn't display the visualization results of Manhattan-SDF [3] because the performance of MonoSDF [13] has been proved to be better than Manhattan-SDF. Therefore, we only compare our method with MonoSDF instead of Manhattan-SDF.

2. Additional Implementation Details

2.1. Chosen of Backbones

To train a neural radiance field as our NeRF prior, we adopt the grid-based architecture of TensoRF [2]. We clarify that our innovation lies in leveraging the prior neural radiance field to provide geometry and color clues for SDF network, instead of proposing a novel grid-based NeRF structure. We choose grid-based NeRF as our prior backbone because of its rapid training speed and its ability to capture high-frequency geometry details. To provide a clear illustration, we replace the structure of the prior network from TensoRF [2] to Instant-NGP [7] and vanilla NeRF [6] and report the comparisons, as shown in Fig. 5. While it takes a long time to train an vanilla MLP-based NeRF, the raw NeRF is able to bring an improvement serving as prior. Instant-NGP, which adopts a backbone of feature grids, shows similar result to TensoRF. This experiment demonstrates that our prior backbone is not confined to a grid-based structure. The key improvement comes from pre-training a different prior network from SDF network to provide additional clues.

2.2. Chosen of Hyper-Parameters

We train the prior NeRF for each scene in 30k iterations, which takes about 30 minutes per scene. For our implicit surface function, we adopt the architecture of NeuS [10], where the signed distance function and color function are modeled by an MLP with 8 and 4 hidden layers, respectively. We train our implicit surface function for 200k iterations in total. The multi-view consistency constraint is applied after 100k iterations and the depth consistency loss is applied after 150k iterations. We adopt such strategy based on the observation that the multi-view consistency and depth loss may mislead the network at the early training stage when the surface is noisy and ambiguous. We set $t_0 = 0.02$ in Eq. 1, $t_1 = 0.04$ and $t_2 = 0.1$ in Eq. 2 in the original paper,

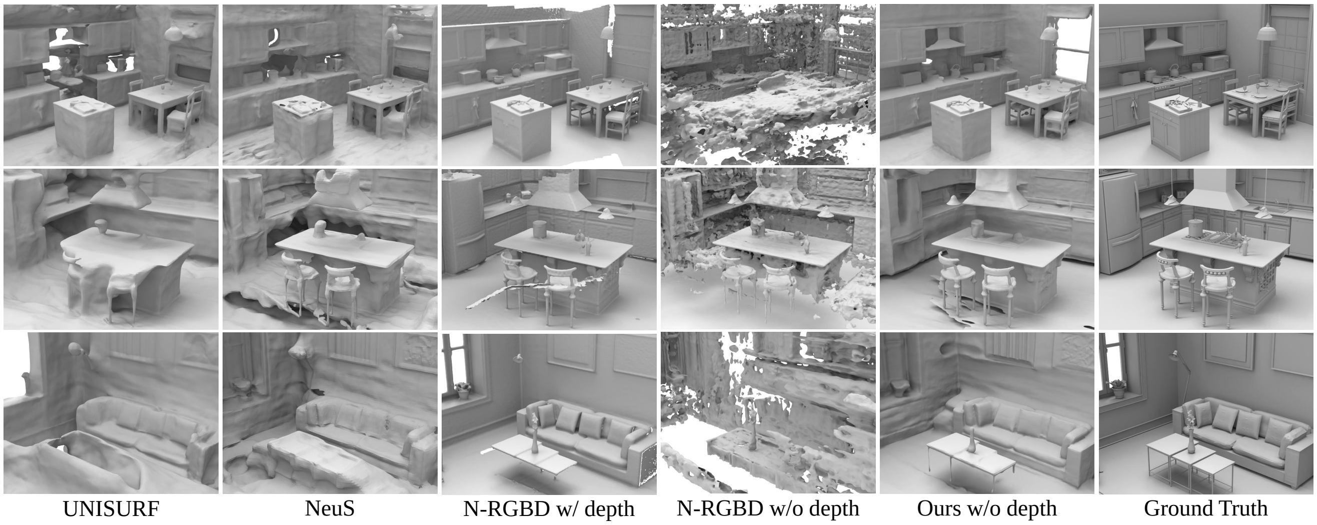


Figure 3. Visualization Comparison on BlendSwap Dataset.

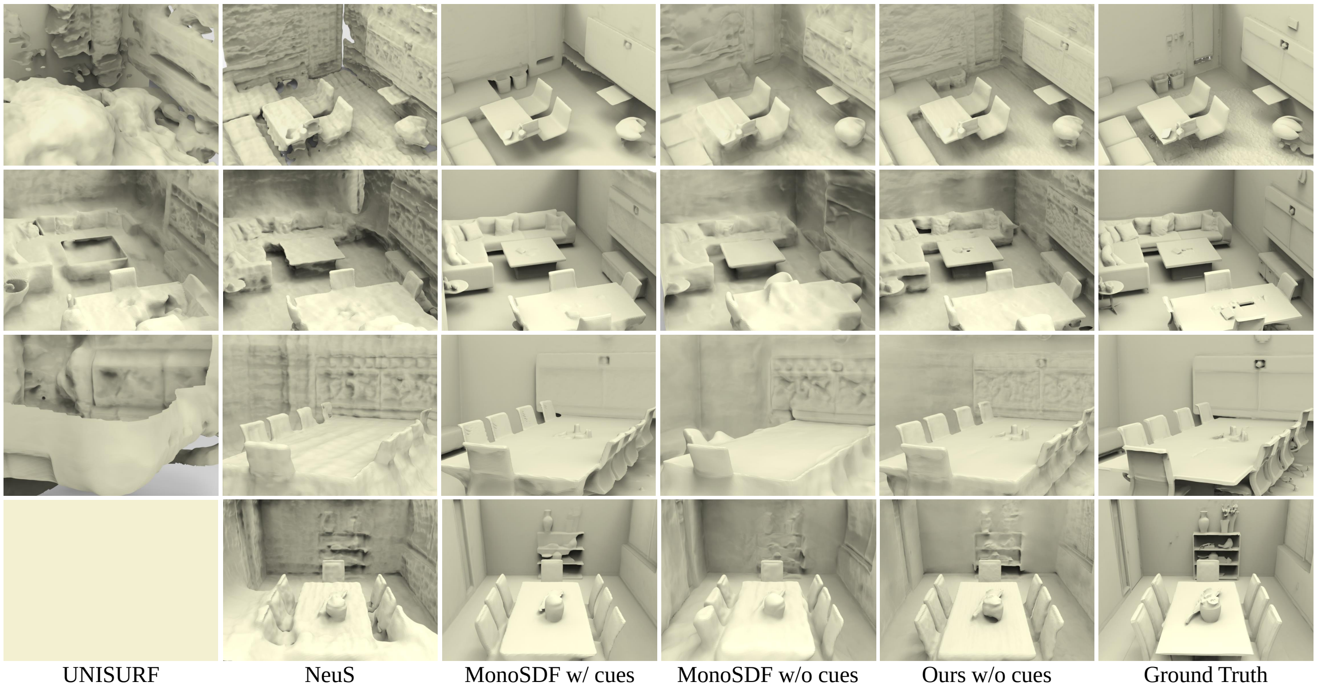


Figure 4. Visualization Comparison on Replica Dataset.

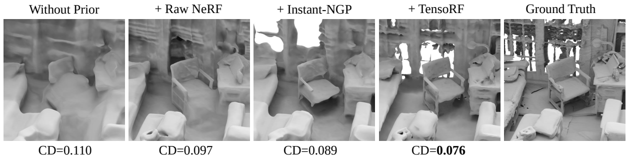


Figure 5. Choice of different NeRF priors.

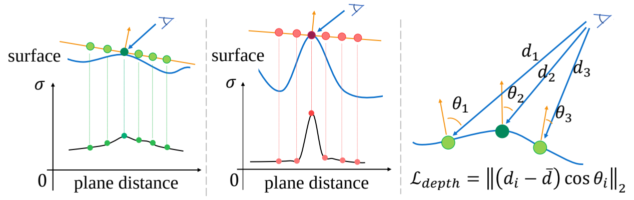
$\lambda_1 = \lambda_2 = 0.1$ and decreases exponentially to 0, $\lambda_3 = 0.05$ and $\lambda_4 = 0.5$ in Eq. 3. All the experiments are conducted on a single NVIDIA RTX 3090Ti GPU.

$$\mathbf{p}^* = \begin{cases} \text{visible} & |\mathbf{c}_s^* - \mathbf{c}_s^{proj}| < t_0 \\ \text{invisible} & |\mathbf{c}_s^* - \mathbf{c}_s^{proj}| \geq t_0 \end{cases} \tag{1}$$

$$\begin{gathered} \text{sgn}_c = \begin{cases} 1 & \text{var}(\mathbf{c}^{proj}) < t_1 \\ 0 & \text{var}(\mathbf{c}^{proj}) \geq t_1 \end{cases} \\ \text{sgn}_\sigma = \begin{cases} 1 & \text{var}(\sigma(\mathbf{p}^*)) < t_2 \\ 0 & \text{var}(\sigma(\mathbf{p}^*)) \geq t_2 \end{cases} \end{gathered} \tag{2}$$

$$\mathcal{L} = \mathcal{L}_{rgb} + \lambda_1 \mathcal{L}_\sigma + \lambda_2 \mathcal{L}_c + \lambda_3 \mathcal{L}_{reg} + \lambda_4 \mathcal{L}_{depth}, \tag{3}$$

Regarding the selection of t_0, we find that the color loss usually converges between 0.03 and 0.06 at the end of train-



(a) Plane Detected **(b) Plane Not Detected** **(c) Depth Consistency Loss**

Figure 6. An illustration of our depth consistency. This is the same as Figure 5 in the original paper.

ing, so we choose a t_0 that is less than the average convergence loss. Regarding t_1, we visualized the effect of different t_1 on segmenting flat areas on images, similar to the paper Super-Plane NeRF [12] Figure 9. We choose a t_1 that can divide planes as much as possible without including more erroneous areas. The selection of t_2 is similar to t_1, where we visualize small planes in 3D space and find a suitable t_2 so that these points are roughly distributed on the same plane.

2.3. Calculation of the Plane Confidence

In subsection "Depth Consistency Loss" in the paper, we impose depth consistency loss on surface points to improve the smoothness and completeness of scene surfaces. Here we discuss the calculation of confidence in detail. Our goal is to judge whether a surface point p^* is on a plane. To this end, we first calculate the normal vector $\mathbf{n}_{p^*}$ at p^*. Then we can get the tangent plane at p^*. Assume that the two basis vector of this plane are $\mathbf{v}_1$ and $\mathbf{v}_2$, and then we calculate the 8 corner coordinates on a square neighborhood of p^*: $p_1 = \mathbf{v}_1, p_2 = \mathbf{v}_1 + \mathbf{v}_2, p_3 = \mathbf{v}_2, p_4 = -\mathbf{v}_1 + \mathbf{v}_2, p_5 = -\mathbf{v}_1, p_6 = -\mathbf{v}_1 - \mathbf{v}_2, p_7 = -\mathbf{v}_2, p_8 = \mathbf{v}_1 - \mathbf{v}_2$. We query the prior density at the 9 locations (p^* and $p_1 \sim p_8$), and calculate the variance of the density of the 9 points, which is equivalent to $\mathrm{var}(\sigma(\mathbf{p}^*))$ in Eq. (8) in the paper.

2.4. Sampling Strategy for Depth Loss

There are two prerequisites before imposing depth consistency loss: (i) the intersection and its neighboring points have similar colors on the projection view, (ii) the intersection and its neighboring points are nearly on a plane. If the two prerequisites are both met, we then constrain the neighborhood points to maintain the same depth on their normal directions. Specifically, while generating a batch of training data, we sample several 3*3 patch of pixels to form a batch. For each patch of pixels, we emit rays and calculate the estimated depth of the rays using volume rendering. For every 9 rays, we calculate the variation of the depths on normal directions of the 9 intersections, and constrain the variation to zero. In this way, we push the surface points on textureless planes to have the same depth, thus improving the smoothness and completeness of textureless areas.

2.5. Prior Filtering Threshold

In subsection "Neural Radiance Field Prior" in the original paper, we mentioned that the prior density field is usually noisy, which may mislead the neural implicit network. Therefore, we filter out the fuzzy density value and apply supervision only if the density value is convincing. As shown in Fig. 6, (a) and (b) are two cases of detected planes and not detected planes. We notice that if the network is not confident about the reconstructed surfaces, it tends to learn a fuzzy distribution of density field, where 3D points near the surface have an medium-sized density, as shown in Fig. 6 (a). On the contrary, if the network is confident about the reconstructed surfaces, it tends to learn a significant boundary, where 3D points near the surface have a very small density as 0 or very large density as more than 100, as shown in Fig. 6 (b). Through this way, we can use the value of density itself as a confidence of whether the prior is convincing. Practically, we use thresholds $t_{upper} = 7$ and $t_{lower} = 1\mathrm{e}-5$ to filter density field. The supervision of density and color is applied only if the density is larger than $t_{upper} = 7$ or smaller than $t_{lower} = 1\mathrm{e}-5$.

3. Evaluation Metrics

Following Neural RGB-D [1] and MonoSDF [13], we adopt Accuracy (Acc), Completeness (Comp), Chamfer Distance-L1 (CD), Normal Consistency (NC), Precision (Prec), Recall and F1-score (F1) as our evaluation metrics. The definitions of the metrics are listed in Tab. 3.

To reconstruct surfaces for scenes, we use the marching cubes algorithm [5]. For ScanNet dataset, since implicit networks can reconstruct artifacts in unobserved regions which will be penalized in evaluation, we render depth maps from the predicted mesh and refuse them using TSDF fusion [8] following [3]. For synthetic dataset BlendSwap and Replica, following [1], we firstly subdivide all the meshes to have a maximum edge length of below 1.5 cm. Then we use the ground truth trajectory of the ground truth depth maps to detect the vertices which are visible by at least one camera. Triangles which have no visible vertices, either due to not being in any of the view frustrum or due to being occluded by other surfaces, are culled. The point cloud to be evaluated is sampled on the culled mesh with a density of 1 point per square centimetre. All metrics are evaluated between the predicted point cloud and ground truth point cloud.

4. Additional Visualization Results

4.1. More Visualization of Ablation Study on Depth Loss

We provide additional visualization results of the ablation study on depth consistency loss, as shown in Fig. 7. The results show that our depth consistency loss further improves

Table 3. Definitions of evaluation metrics. P and Q are point clouds sampled from predicted mesh and ground truth mesh, respectively. n_p is the normal vector at point p.

Metric	Definition
Acc	$\underset{p\in P}{\text{mean}}(\underset{p^*\in Q}{\min} \\|p-p^*\\|_1)$
Comp	$\underset{p\in Q}{\text{mean}}(\underset{p^*\in P}{\min} \\|p-p^*\\|_1)$
Chamfer Distance-L1	$\frac{\text{Acc}+\text{Comp}}{2}$
Normal-Acc	$\underset{p\in P}{\text{mean}}(n_p^T n_{p^*})$ s.t. $p^* = \underset{p^*\in Q}{\text{argmin}} \\|p-p^*\\|_1$
Normal-Comp	$\underset{p\in Q}{\text{mean}}(n_p^T n_{p^*})$ s.t. $p^* = \underset{p^*\in P}{\text{argmin}} \\|p-p^*\\|_1$
Normal Consistency	$\frac{\text{NormalAcc}+\text{NormalComp}}{2}$
Precision	$\underset{p\in P}{\text{mean}}(\underset{p^*\in Q}{\min} \\|p-p^*\\|_1 < 0.05)$
Recall	$\underset{p\in Q}{\text{mean}}(\underset{p^*\in P}{\min} \\|p-p^*\\|_1 < 0.05)$
F1-score	$\frac{2\times\text{Precision}\times\text{Recall}}{\text{Precision}+\text{Recall}}$

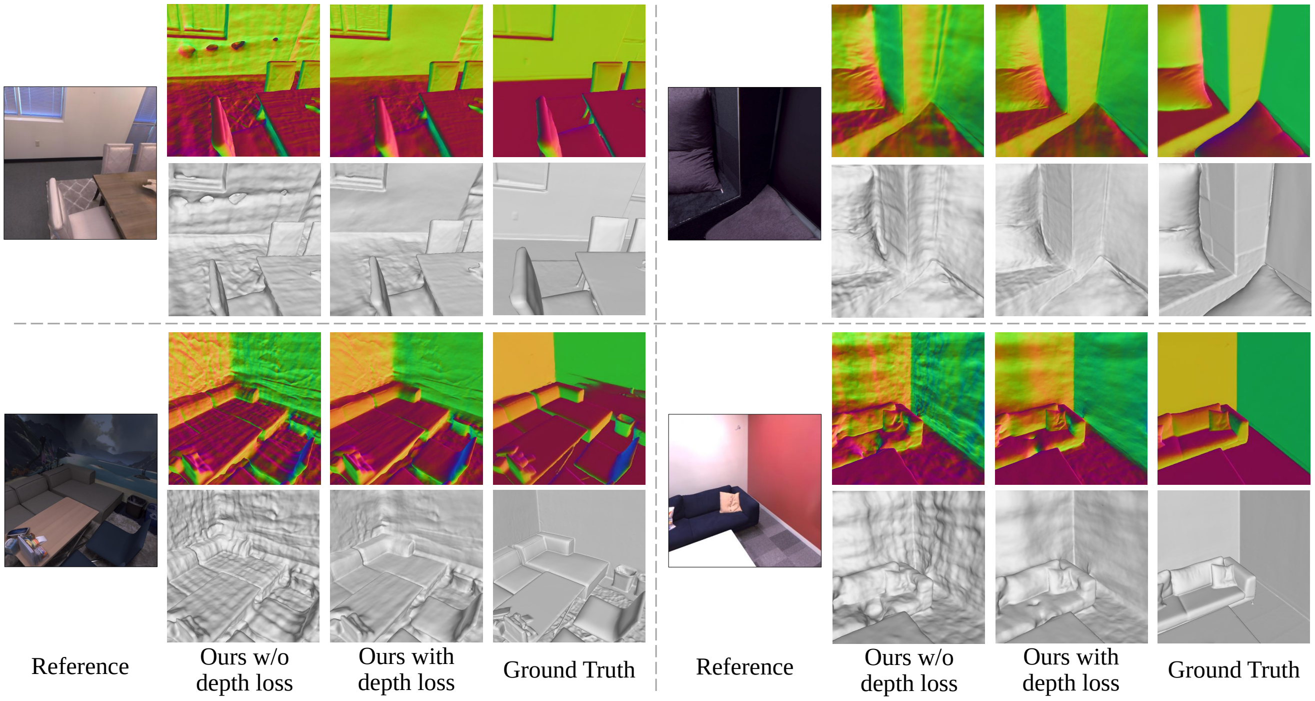


Figure 7. Visualization Results of Ablation Study on Depth Consistency Loss.

the smoothness and completeness of the reconstructed surfaces.

4.2. Scene Reconstruction Display

We made a video in the supplementary to provide additional examples of the reconstructed indoor scenes (Section 1 in the video). The results show that our method is able to obtain smooth, complete and high fidelity surfaces of scenes. Please refer to the video for more details.

4.3. Visualization of Multi-view and Depth Loss

In our video, we additionally provide a visualization of our depth loss (Section 2) and multi-view constraint (Section 3). In Section 2 of the video, we firstly find out the textureless planes using color variance and density variance. For the textureless planes, we visualize the sampled ray positions

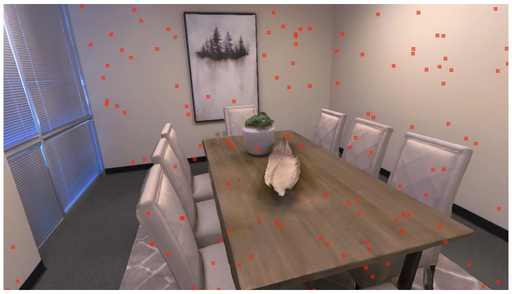


Figure 8. Visualization of pixel positions constrained by depth loss.

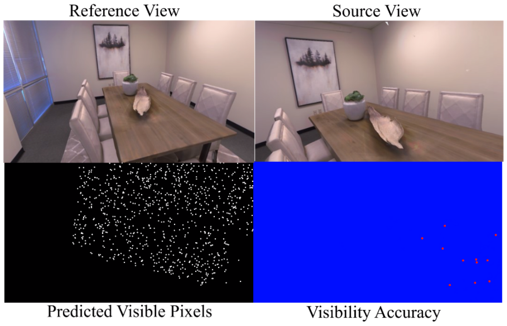
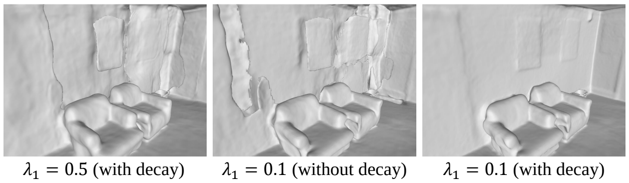


Figure 9. Visualization of projection pixels which are visible predicted by our method in source view.



λ_1 = 0.5 (with decay) λ_1 = 0.1 (without decay) λ_1 = 0.1 (with decay)

Figure 10. Visualization of different supervision weight and whether using weight decay strategy or not. Not using weight decay or assigning a too large weight both leads to bad reconstruction results.

in the input view in every training epoch, colored in red points. As shown in the video and Fig. 8, our depth loss is accurately imposed to textureless areas in the room, such as walls, floors and floors. In Section 3 of the video, we randomly emit rays from reference view and project the intersections into the source view. As shown in Fig. 9, all the non-occluded intersections are visualized as white points in the left-bottom image. The accuracy of visibility check is shown in the right-bottom image (red point represents wrong and blue point represents correct). Almost all of the visibility of the intersections in the source view are correctly predicted. Comparing to the traditional MVS method which depends on projection color to judge occlusion, our method of using local-prior volume rendering achieves better accuracy of the visibility check.

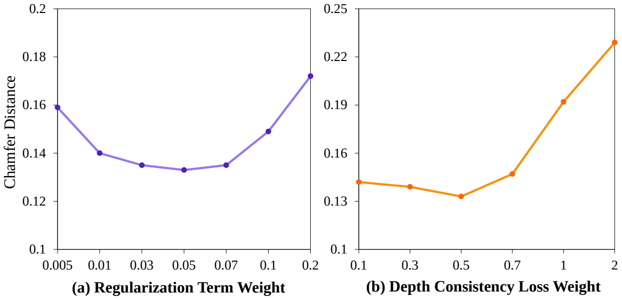
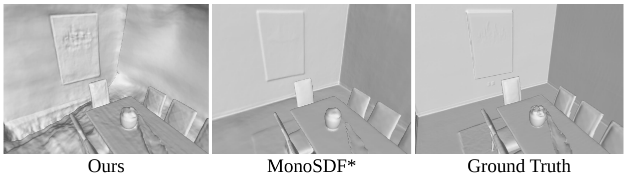


Figure 11. Ablation on the weights of regularization term and depth consistency loss.



Ours MonoSDF* Ground Truth

Figure 12. Failure case of our method comparing with MonoSDF*. This is due to the weak-observation of the input few views.

5. Ablation Study

5.1. Loss Function

Our loss function can be written as

$$\mathcal{L} = \mathcal{L}_{rgb} + \lambda_1 \mathcal{L}_{\sigma} + \lambda_2 \mathcal{L}_{c} + \lambda_3 \mathcal{L}_{reg} + \lambda_4 \mathcal{L}_{depth}, \tag{4}$$

where $\mathcal{L}_{rgb}$ is the error between rendered color and ground truth pixel color, $\mathcal{L}_{\sigma}$ and $\mathcal{L}_{c}$ are the supervision of density and color from our prior field, $\mathcal{L}_{reg}$ is the regularization of SDF field, $\mathcal{L}_{depth}$ is our depth consistency loss. For prior supervision, we set $\lambda_1 = \lambda_2 = 0.1$ and decrease exponentially to 0, similar to the decreasing strategy in [13]. Fig. 10 provides an ablation study on the weight of prior supervision. Not using weight decay or assigning a too large weight both leads to worse reconstruction results, because the prior field shows poor performance in textureless areas and may mislead the network. We also conduct an ablation study on the chosen of λ_3 and λ_4, as shown in Fig. 11. Too large or too small weights will both cause significant degeneration of performance. We also visualize the training curve of each loss term, as shown in Fig. 13. Metion that depth consistency loss is not added at the beginning of traning because it may mislead the network at the early training stage when the surface is noisy and ambiguous. Each term of the loss function smoothly decreases with the training progress.

5.2. Convergence Speed

In Fig. 14, we show the reconstruction results of MonoSDF and our method in the early training stage. It can be seen that under the same epochs, our method is able to highlight the scene details much faster than MonoSDF due to the hint of the prior field, which justifies the superiority of our method.

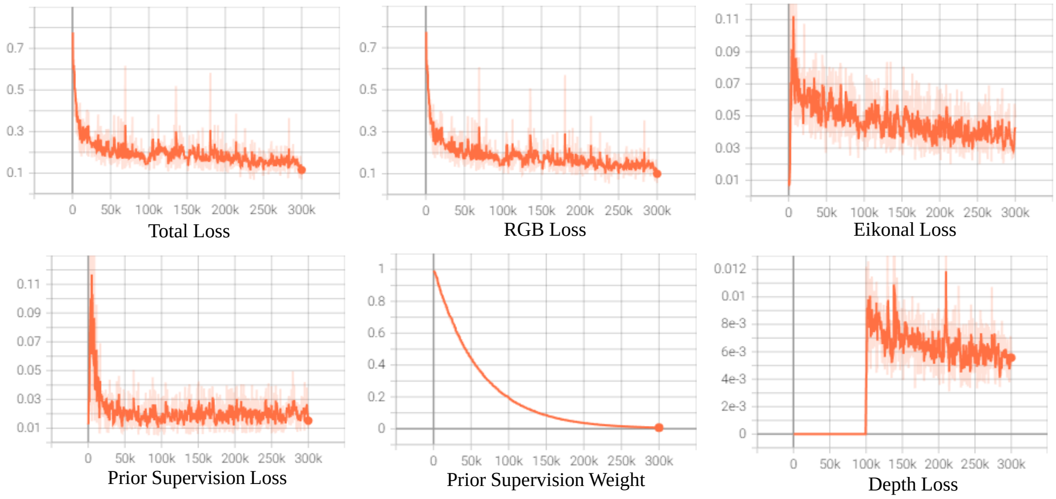


Figure 13. Visualization of the training curve of each loss function term.

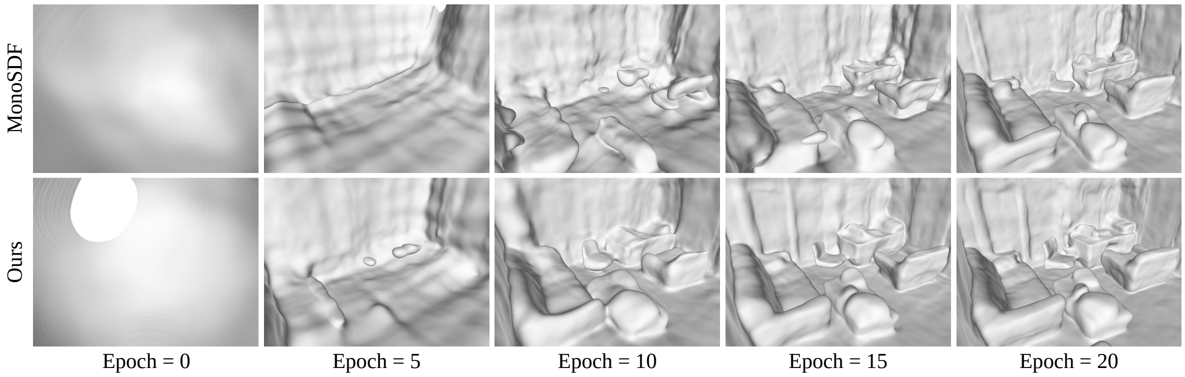


Figure 14. Reconstruction results under different epochs. Our method shows significantly faster convergence speed.

5.3. Failure Case

Our method achieves better visual effects than MonoSDF [13] in most scenes without using data-driven priors. Fig. 12 provides an example of failure case of our method (without depth and normal cues) comparing to MonoSDF* (MonoSDF with depth and normal cues). The failure case is because there are some weak-observed areas in the dataset, such as the walls blocked by chairs, the carpet under the tables. These areas are visible in only a few input views, and the NeRF model can not inference the geometry well through sparse input views. However, MonoSDF* can recover the weak-observed areas better using the additional data-driven depth priors and normal priors.

References

[1] Dejan Azinović, Ricardo Martin-Brualla, Dan B Goldman, Matthias Nießner, and Justus Thies. Neural RGB-D surface reconstruction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6290–6301, 2022. 4

[2] Anpei Chen, Zexiang Xu, Andreas Geiger, Jingyi Yu, and Hao Su. Tensorf: Tensorial radiance fields. In *European Conference on Computer Vision (ECCV)*, pages 333–350. Springer, 2022. 2

[3] Haoyu Guo, Sida Peng, Haotong Lin, Qianqian Wang, Guofeng Zhang, Hujun Bao, and Xiaowei Zhou. Neural 3D scene reconstruction with the Manhattan-world assumption. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5511–5520, 2022. 2, 4

[4] Zhaoshuo Li, Thomas Müller, Alex Evans, Russell H Taylor, Mathias Unberath, Ming-Yu Liu, and Chen-Hsuan Lin. Neuralangelo: High-fidelity neural surface reconstruction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8456–8465, 2023. 1

[5] William E. Lorensen and Harvey E. Cline. Marching cubes: A high resolution 3D surface construction algorithm. *Computer Graphics*, 21(4):163–169, 1987. 4

[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 *European Conference on Computer Vision (ECCV)*, pages 405–421. Springer, 2020. 2

[7] Thomas Müller, Alex Evans, Christoph Schied, and Alexander Keller. Instant neural graphics primitives with a multireso-

lution hash encoding. *ACM Transactions on Graphics (ToG)*, 41(4):1–15, 2022. 1, 2

[8] Richard A Newcombe, Shahram Izadi, Otmar Hilliges, David Molyneaux, David Kim, Andrew J Davison, Pushmeet Kohi, Jamie Shotton, Steve Hodges, and Andrew Fitzgibbon. Kinectfusion: Real-time dense surface mapping and tracking. In *2011 10th IEEE international symposium on mixed and augmented reality*, pages 127–136. Ieee, 2011. 4

[9] Jingwen Wang, Tymoteusz Bleja, and Lourdes Agapito. Go-Surf: Neural feature grid optimization for fast, high-fidelity rgb-d surface reconstruction. In *2022 International Conference on 3D Vision (3DV)*, pages 433–442. IEEE, 2022. 1

[10] Peng Wang, Lingjie Liu, Yuan Liu, Christian Theobalt, Taku Komura, and Wenping Wang. NeuS: Learning neural implicit surfaces by volume rendering for multi-view reconstruction. *Advances in Neural Information Processing Systems*, 34:27171–27183, 2021. 1, 2

[11] Yiqun Wang, Ivan Skorokhodov, and Peter Wonka. HF-NeuS: Improved Surface Reconstruction Using High-Frequency Details. *Advances in Neural Information Processing Systems*, 35:1966–1978, 2022. 1

[12] Botao Ye, Sifei Liu, Xueting Li, and Ming-Hsuan Yang. Self-supervised super-plane for neural 3d reconstruction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 21415–21424, 2023. 4

[13] Zehao Yu, Songyou Peng, Michael Niemeyer, Torsten Sattler, and Andreas Geiger. MonoSDF: Exploring monocular geometric cues for neural implicit surface reconstruction. *Advances in neural information processing systems*, 35:25018–25032, 2022. 2, 4, 6, 7