# Supplementary Material for: Ray-Distance Volume Rendering for Neural Scene Reconstruction

Ruihong Yin[1], Yunlu Chen[2], Sezer Karaoglu[1,3], and Theo Gevers[1,3]

[1] University of Amsterdam, Amsterdam, The Netherlands
[2] Carnegie Mellon University, Pittsburgh, The United States
[3] 3DUniversum, Amsterdam, The Netherlands
{r.yin, Th.Gevers}@uva.nl, s.karaoglu@3duniversum.com,
yunluche@andrew.cmu.edu

This supplementary document provides additional details and experimental results of our approach. The definitions of evaluation metrics are given in Sec. A. Sec. B details loss functions employed in our method. The implementation details are discussed in Sec. C. Sec. D shows additional qualitative and quantitative results. Sec. E offers a complementary analysis of the efficacy of our approach.

## A Evaluation Metrics

The definitions of metrics are given in Tab. S1, including accuracy (acc), completeness (comp), precision (prec), recall, F-score, normal consistency (normal c.), chamfer L1 distance (Chamfer-L1), and peak signal-to-noise ratio (PSNR). Lower values denote superior performance for accuracy, completeness, and chamfer L1 distance, whereas higher values are preferable for precision, recall, F-score, normal consistency, and PSNR.

## B Details of Loss Function

This section provides the computation for the RGB loss $\mathcal{L}_c$, the depth loss $\mathcal{L}_d$, the normal loss $\mathcal{L}_n$, the smooth loss $\mathcal{L}_s$, and the Eikonal loss $\mathcal{L}_e$.

To enhance the prediction of geometry and appearance, the RGB loss comprises the color loss with density derived from SRDF, denoted as $\mathcal{L}_c^{\text{SRDF}}$, and another color loss with density transformed from SDF, denoted as $\mathcal{L}_c^{\text{SDF}}$. They are calculated as follows:

$$\begin{aligned}\mathcal{L}_c &= \mathcal{L}_c^{\text{SRDF}} + \mathcal{L}_c^{\text{SDF}} \\ \mathcal{L}_c^{\text{SRDF}} &= \frac{1}{N_s}\sum_{\mathbf{r}\in\mathcal{S}^2} \parallel \mathbf{C}^{\text{SRDF}} - \bar{\mathbf{C}} \parallel_1 \\ \mathcal{L}_c^{\text{SDF}} &= \frac{1}{N_s}\sum_{\mathbf{r}\in\mathcal{S}^2} \parallel \mathbf{C}^{\text{SDF}} - \bar{\mathbf{C}} \parallel_1\end{aligned} \tag{S-1}$$

where $\mathcal{S}^2$ represents the set of rays in a minibatch. $\bar{\mathbf{C}}$ denotes the ground truth of color. $\mathbf{C}^{\text{SRDF}}$ is the rendered color with density derived from SRDF as detailed in

**Table S1:** Definitions of evaluation metrics. $\mathbf{p}$ is the sampled point cloud from the predicted mesh, while $\bar{\mathbf{p}}$ is from the ground truth. $\mathbf{n}_{\mathbf{p}}$ and $\mathbf{n}_{\bar{\mathbf{p}}}$ represent normal vectors computed from the prediction and ground truth respectively. $\mathbf{C}^n_{pred}$ / $\mathbf{C}^n_{gt}$ is the $n$-th predicted/ground truth image. $N_I$ is the number of images for evaluation.

| Metric | Definition |
|---|---|
| Acc | $\text{mean}_{\mathbf{p}\in P}(\text{min}_{\bar{\mathbf{p}}\in\bar{P}}\lvert\lvert\mathbf{p}-\bar{\mathbf{p}}\rvert\rvert_1)$ |
| Comp | $\text{mean}_{\bar{\mathbf{p}}\in\bar{P}}(\text{min}_{\mathbf{p}\in P}\lvert\lvert\mathbf{p}-\bar{\mathbf{p}}\rvert\rvert_1)$ |
| Chamfer-L1 | $\frac{\text{Acc+Comp}}{2}$ |
| Prec | $\text{mean}_{\mathbf{p}\in P}(\text{min}_{\bar{\mathbf{p}}\in\bar{P}}\lvert\lvert\mathbf{p}-\bar{\mathbf{p}}\rvert\rvert_1 < .05)$ |
| Recall | $\text{mean}_{\bar{\mathbf{p}}\in\bar{P}}(\text{min}_{\mathbf{p}\in P}\lvert\lvert\mathbf{p}-\bar{\mathbf{p}}\rvert\rvert_1 < .05)$ |
| F-score | $\frac{2\times\text{Prec}\times\text{Recall}}{\text{Prec+Recall}}$ |
| Normal-Acc | $\text{mean}_{\mathbf{p}\in P}\left(\mathbf{n}_{\mathbf{p}}^T\mathbf{n}_{\bar{\mathbf{p}}}\right)$ s.t. $\bar{\mathbf{p}} = \underset{\bar{\mathbf{p}}\in\bar{P}}{\text{argmin}}\,\lVert\mathbf{p}-\bar{\mathbf{p}}\rVert_1$ |
| Normal-Comp | $\text{mean}_{\bar{\mathbf{p}}\in\bar{P}}\left(\mathbf{n}_{\mathbf{p}}^T\mathbf{n}_{\bar{\mathbf{p}}}\right)$ s.t. $\mathbf{p} = \underset{\mathbf{p}\in P}{\text{argmin}}\,\lVert\mathbf{p}-\bar{\mathbf{p}}\rVert_1$ |
| Normal Consistency | $\frac{\text{Normal-Acc+Normal-Comp}}{2}$ |
| PSNR | $\text{mean}_{n\in[1,N_I]}(20\cdot\log_{10}(1/\sqrt{\text{MSE}(\mathbf{C}^n_{pred},\mathbf{C}^n_{gt})}))$ |

Eq. (5), while $\mathbf{C}^{\text{SDF}}$ represents the color rendered with the density that is derived from SDF according to Eq. (3). $N_s$ refers to the number of points sampled per minibatch.

The depth and normal losses, serving to constrain the 3D geometry, are defined as follows:

$$\begin{aligned}
D &= \sum_{i=1}^{N} T_i\alpha_i z_i, \mathbf{N} = \sum_{i=1}^{N} T_i\alpha_i \mathbf{n}_i \\
\mathcal{L}_d &= \frac{1}{N_s}\sum_{\mathbf{r}\in\mathcal{S}^2} \parallel (wD+q) - \bar{D} \parallel_2 \\
\mathcal{L}_n &= \frac{1}{N_s}\sum_{\mathbf{r}\in\mathcal{S}^2} (\parallel \mathbf{N} - \bar{\mathbf{N}} \parallel_1 + \parallel 1 - \mathbf{N}^T\bar{\mathbf{N}} \parallel_1)
\end{aligned} \tag{S-2}$$

where $\bar{D}$ and $\bar{\mathbf{N}}$ correspond to the ground truth of depth and normal, predicted by the pre-trained Omnidata model [3]. $D$ and $\mathbf{N}$ denote rendered depth and normal, respectively. $T_i$ and $\alpha_i$ are computed by Eq. (1) with density derived from SRDF. To address potential depth ambiguities, a scale parameter $w$ and a shift parameter $q$ are utilized to ensure predicted depth $D$ matches the ground truth $\bar{D}$, as utilized by MonoSDF.

The smooth loss, defined in Eq. (S-3), is computed by considering the gradients of the SDF between adjacent points. This encourages a smoother reconstructed surface.

$$\mathcal{L}_s = \frac{1}{N_{sm}}\sum_{\mathbf{p}\in\mathcal{X}} \parallel \nabla d_{\Omega}(\mathbf{p}) - \nabla d_{\Omega}(\mathbf{p}+\varepsilon) \parallel_2 \tag{S-3}$$

where $\varepsilon$ is a small perturbation applied to the point. $\mathcal{X}$ is the set of sampled points. $d_{\Omega}(\mathbf{p})$ refers to the SDF output produced by the geometry MLP. Additionally, $N_{sm}$ indicates the number of points for computing the smooth loss.

The Eikonal loss [4] serves as a regularization term for SDF prediction, detailed in Eq. (S-4).

$$\mathcal{L}_e = \frac{1}{N_{ei}} \sum_{\mathbf{p} \in \mathcal{X}} (\parallel \nabla d_{\Omega}(\mathbf{p}) \parallel_2 - 1)^2 \tag{S-4}$$

where $N_{ei}$ refers to the number of sampled points for the Eikonal loss.

# C Implementation Details

The MLP representation employs a geometry MLP comprising 8 layers with a hidden dimension of 256, while the grid representation utilizes two layers. Both the color MLP and SRDF MLP consist of two layers each, with an intermediate output comprising 256 channels. Softplus is used as the activation layer in the geometry MLP and SRDF MLP, while ReLU is applied in the color MLP. The network utilizes geometric initialization [1] and is optimized over 200,000 iterations with Adam optimizer [6]. The resolution of 2D output is set to 384 × 384. Each iteration includes the sampling of 1024 rays. The initial learning rate is configured as 5e-4 for geometry MLP and color MLP, 1e-5 for the SRDF branch, and 1e-2 for feature grids. The hyperparameter $k$ in the SRDF-SDF consistency loss is set to 12. Loss weights $\{\lambda_d, \lambda_n, \lambda_e, \lambda_s, \lambda_{con}, \lambda_{vis}\}$ are {0.1, 0.05, 0.05, 0.005, 1.0, 0.001}. Following MonoSDF, the ground truth for monocular geometric cues is predicted by the pre-trained Omnidata model [3]. The optimization is performed on one NVIDIA A100 GPU.

# D Additional Results

## D.1 Additional Ablation Study

**Impact of SRDF Branch:** Additional ablation study, shown in Tab. S3, evaluates the importance of SRDF branch. The experiments include: (a) *Baseline*: MonoSDF with MLP representation. (b) *SDF to SRDF*: this setup generates SRDF from the predicted SDF, eliminating the need for the SRDF branch. The SRDF is then used to calculate volume density. (c) *SRDF branch:* SRDF is predicted by the SRDF branch. In Tab. S3, Row (b) performs much worse than the baseline. The reason is that the SDF prediction for a single point is affected by all nearby objects, which can result in inaccuracies, generating false surfaces and imprecise SRDF. Thus, without the SRDF branch, SRDF-based volume rendering may yield inaccurate weights, encountering the same issues as SDF-based volume rendering. Moreover, since the surface is interpolated by the SDF of near-surface points, SRDF derived from SDF lacks adequate supervision for SDF during training, which can result in poorer geometry. Hence, generating SRDF

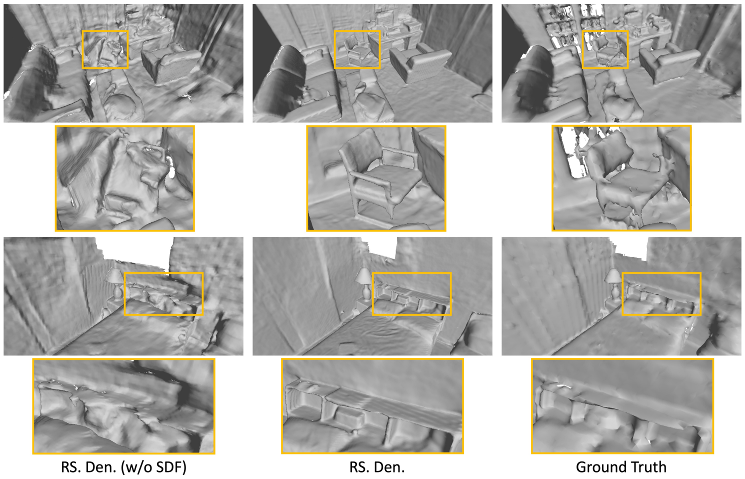

**Fig. S1: Visualization of structures with/without SDF on ScanNet.** With SDF representing the surface geometry, our method (*RS. Den.*) generates smoother and more accurate surfaces.

**Table S2:** Ablation study for the impact of SDF in our method on ScanNet.

| Method | Acc ↓ | Comp ↓ | Prec ↑ | Recall ↑ | F-score ↑ |
|---|---|---|---|---|---|
| RS. Den. (w/o SDF) | 0.061 | 0.079 | 0.600 | 0.540 | 0.568 |
| RS. Den. | 0.040 | 0.044 | 0.772 | 0.720 | 0.745 |

from the SDF branch is not optimal. Instead, Row (c) shows that the structure with an SRDF branch can avoid the influence of inaccurate SDF and achieve better results. Additionally, the joint optimization of SRDF and SDF branches can integrate SRDF information into the SDF branch, thereby improving SDF prediction. In conclusion, the SRDF branch is crucial for SRDF-based volume rendering.

**Impact of SDF Prediction:** Tab. S2 measures the importance of using SDF for 3D surface localization. Two configurations are compared: (1) *RS. Den. (w/o SDF):* This structure only predicts ray-specific SRDF, without using SDF for 3D geometry representation. During training, color loss and depth loss are used to optimize the network. Since SRDF relies on viewing direction, directly extracting

**Table S3:** Ablation study for the impact of SRDF branch in our method on ScanNet.

| | Method | Acc ↓ | Comp ↓ | Prec ↑ | Recall ↑ | F-score ↑ |
|---|---|---|---|---|---|---|
| a | Baseline | 0.035 | 0.048 | 0.799 | 0.681 | 0.733 |
| b | SDF to SRDF | 0.049 | 0.068 | 0.673 | 0.591 | 0.628 |
| c | SRDF branch | 0.040 | 0.044 | 0.772 | 0.720 | 0.745 |

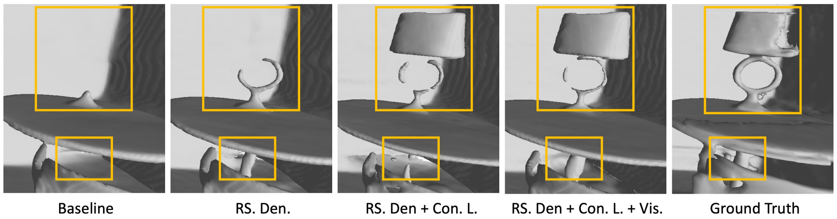

**Fig. S2: Visualization for ablation study.** Our ray-specific volume density (*RS. Den.*), SRDF-SDF consistency loss (*Con. L.*), and self-supervised visibility task (*Vis.*) all lead to better performance.

**Table S4:** Evaluation of 3D reconstruction meshes on ScanNet.

| Method | Acc ↓ | Comp ↓ | Prec ↑ | Recall ↑ | F-score ↑ |
|---|---|---|---|---|---|
| NeuRIS [7] | 0.050 | 0.049 | 0.717 | 0.669 | 0.692 |
| NeuRIS + ours | 0.044 | 0.046 | 0.768 | 0.717 | 0.742 |

the surface from SRDF is not feasible. Therefore, during inference, to generate 3D meshes, the network first renders 2D depth with volume density derived from SRDF, after which TSDF Fusion [2] is adopted to fuse the multi-view depth maps. (2) *RS. Den.:* This structure predicts both SRDF and SDF, where SRDF models the volume density and SDF locates 3D surfaces. Notably, the SRDF-SDF consistency loss and self-supervised visibility task are not employed in this ablation study. In Tab. S2, The structure *RS. Den. (w/o SDF)* achieves much worse results than *RS. Den.*. This occurs because the depth label in the depth loss is generated by a pre-trained network, which may yield inaccurate absolute depth and introduce noises. Besides, the scale inconsistency between overlapped regions in the predicted depth also results in coarse surfaces, *e.g.* walls and floors, as illustrated in Fig. S1. In contrast, our structure with SDF yields superior reconstruction meshes.

**Visualization for Ablation Study:** To further comprehend the effectiveness of our proposed method, Fig. S2 shows the visualization for ablation study. It can be seen that our ray-specific volume density, SRDF-SDF consistency loss, and self-supervised visibility task all contribute to high-quality reconstruction.

### D.2 Applying Our SRDF-based Solution to NeuRIS [7]

In 3D surface reconstruction, VolSDF [9] and NeuS [8] are two mainstream baselines that apply volume rendering based on SDF. Our baseline MonoSDF, discussed in the main text, is an extension of VolSDF for indoor scene reconstruction. To assess the generalization of our method, we also apply our method to the NeuS-based method, namely NeuRIS [7], which also operates on indoor scenes. Tab. S4 compares our method based on NeuRIS and the baseline NeuRIS. Our approach outperforms NeuRIS by a significant margin, *e.g.* achieving a 4.8% in-

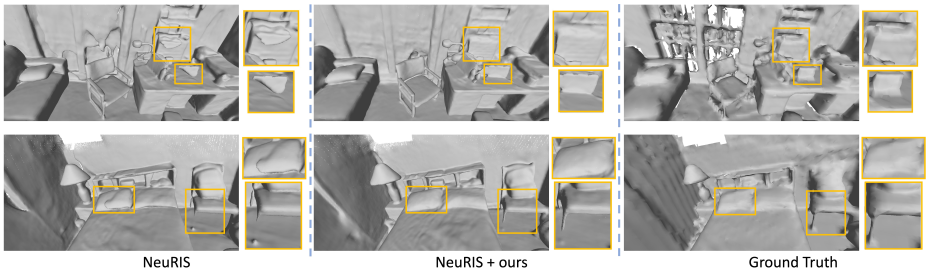

**Fig. S3: Visualization of NeuRIS and our method on ScanNet.** Our method improves the reconstruction performance.

**Table S5:** 3D reconstruction metrics for objects on ScanNet.

| | 3-class objects | | All objects | |
|---|---|---|---|---|
| | Comp↓ | Recall↑ | Comp↓ | Recall↑ |
| MonoSDF_Grid | 0.041 | 0.754 | 0.047 | 0.662 |
| Ours_Grid | 0.035 | 0.801 | 0.036 | 0.779 |
| MonoSDF_MLP | 0.034 | 0.849 | 0.038 | 0.772 |
| Ours_MLP | **0.026** | **0.886** | **0.031** | **0.827** |

crease in recall and a 5.0% increase in F-score. Qualitative comparisons are given in Fig. S3. Our approach demonstrates the ability to reconstruct more accurate surfaces compared to NeuRIS. Overall, these results highlight the efficacy of our method across both VolSDF-based and NeuS-based methods.

### D.3 Additional Reconstruction Results

To further verify that our method is beneficial for small/thin objects, we also evaluate the reconstruction performance for objects. Using the semantic segmentation mask as prior, we adopt two evaluation approaches: (1) Assessing '*3-class objects*' including chairs, tables, and lamps, which often contain thin/small regions. (2) Evaluating '*all objects*' excluding walls, ceilings, floors, windows, and doors, as they may not include thin/small regions. In Tab. S5, compared to MonoSDF, our method demonstrates significant improvements in both completeness and recall, confirming its effectiveness in object reconstruction.

Fig. S4 and Fig. S5 give additional visualizations on ScanNet and Replica, respectively, with the grid representation. In comparison to MonoSDF, our method demonstrates superior performance in capturing more surfaces and details, *e.g.* chairs, toilets, and shelves.

Fig. S6 and Fig. S8 present the reconstructed surfaces using grid and MLP representations on Tanks and Temples. It can be seen that our approach yields much better surfaces compared to MonoSDF and Occ_SDF_Hybrid.

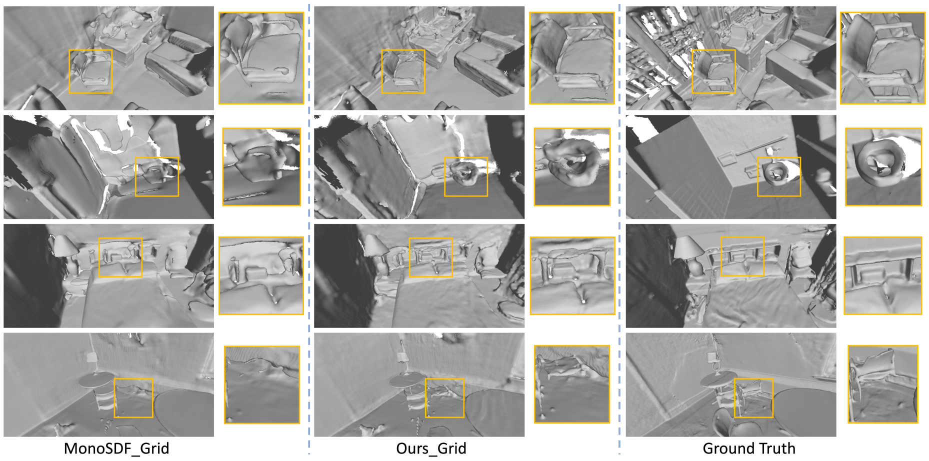

**Fig. S4: Additional results on ScanNet.** Our method can recall more regions.

### D.4 Visualization for View Synthesis

Fig. S7 presents visualization for rendered views. MonoSDF with MLP representation produces blurred regions, whereas our method captures more details and textures in the rendered views. Similarly, with the grid representation, our approach generates more precise views.

## E Additional Analysis

### E.1 Analysis for Reconstruction

The reconstruction task aims to predict accurate SDF values. Existing NeRF-based reconstruction methods [5,9] typically apply a 2D color loss as the primary loss function, in which the predicted color is rendered with volume density derived from SDF. However, the 2D color loss aims to learn the actual appearance, potentially interfering with the optimization of 3D geometry. For example, in the toy scene depicted in Fig. 1, the 2D color loss guides the network to assign a higher weight to the point **P** and a lower weight to the point **Q**. Consequently, the lower weight for the point **Q** encourages the network to learn a SDF with a larger absolute value, contradicting the actual SDF for **Q**. This discrepancy negatively impacts surface reconstruction, particularly for small objects and thin regions.

Our method, including the ray-conditioned density function with SRDF, the SRDF-SDF consistency loss, and the self-supervised visibility task, can solve the aforementioned issue, generating more accurate surface geometry. The reasons are as follows: **(1) Reduced negative impact from 2D output:** Our approach utilizes SRDF to represent density, guiding the network to predict a positive SRDF with a large absolute value at point **Q**, consistent with SRDF's

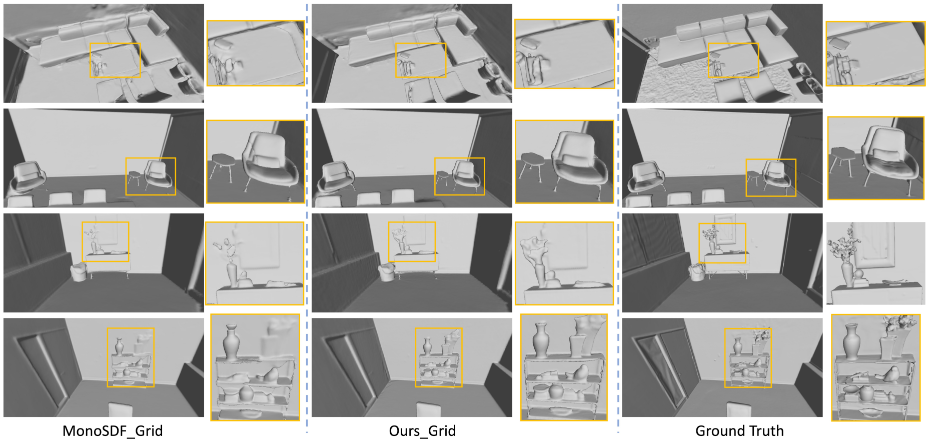

**Fig. S5: Additional results on Replica.** Our method can generate more details.

definition. This partially mitigates the negative influence of 2D losses on SDF. The comparison between Fig. S2 - *(Baseline)* and Fig. S2 -*(RS. Den.)* also proves that our method can generate better geometry, *e.g.* recalling thin objects. **(2) Positive impact from the proposed SRDF-SDF consistency loss:** As analyzed in Fig. 1, the density generated from the SRDF closely matches the actual observation. Hence, with multi-view input, the network can learn an accurate SRDF and density distribution among the entire 3D space. However, the network may struggle to produce precise SDF, resulting in missed surfaces, as shown in Fig. S2 -*(RS. Den.)*. Our SRDF-SDF consistency loss is designed to ensure SDF shares the same sign as SRDF, which can facilitate the recall of missed surfaces. As observed in Fig. S2 -*(RS. Den. + Con. L.)*, more surfaces are recalled. **(3) Positive impact from the proposed self-supervised visibility task:** Our self-supervised visibility task incorporates the physical visibility prior into the network, which can help the learning of SRDF and SDF. The comparisons in Fig. S2 demonstrate that our self-supervised visibility task can recover more surfaces. Overall, our proposed method contributes to producing accurate SDF and reconstructing superior 3D geometry.

### E.2 Analysis for View Synthesis

Existing methods [5, 9] utilize SDF to generate the volume density. However, as depicted in Fig. 1 and Fig. 5, the transformed density and weight derived from predicted SDF may introduce noises during volume rendering, resulting in inaccurate 2D color and consequently low PSNR. In contrast, our approach generates volume density from SRDF when rendering 2D color, ensuring consistent weights with actual observations and avoiding the influence of SDF. As a result, views rendered by our method achieve higher PSNR.

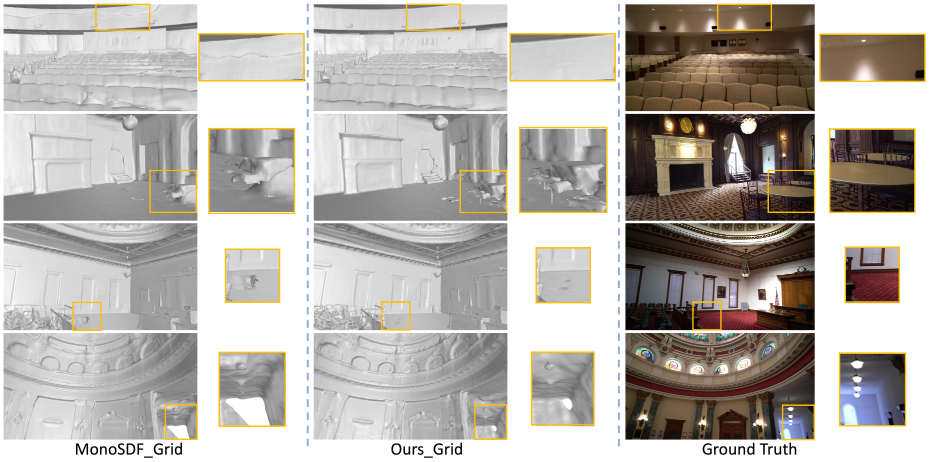

**Fig. S6: Additional results on Tanks and Temples.** Our method enhances the quality of reconstructed meshes.

## References

1. Atzmon, M., Lipman, Y.: Sal: Sign agnostic learning of shapes from raw data. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 2565–2574 (2020)
2. Curless, B., Levoy, M.: A volumetric method for building complex models from range images. In: Proceedings of the 23rd Annual Conference on Computer Graphics and Interactive Techniques. pp. 303–312 (1996)
3. Eftekhar, A., Sax, A., Malik, J., Zamir, A.: Omnidata: A scalable pipeline for making multi-task mid-level vision datasets from 3d scans. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 10786–10796 (2021)
4. Gropp, A., Yariv, L., Haim, N., Atzmon, M., Lipman, Y.: Implicit geometric regularization for learning shapes. In: Proceedings of the International Conference on Machine Learning. pp. 3789–3799 (2020)
5. Guo, H., Peng, S., Lin, H., Wang, Q., Zhang, G., Bao, H., Zhou, X.: Neural 3d scene reconstruction with the manhattan-world assumption. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 5511–5520 (2022)
6. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980 (2014)
7. Wang, J., Wang, P., Long, X., Theobalt, C., Komura, T., Liu, L., Wang, W.: Neuris: Neural reconstruction of indoor scenes using normal priors. In: Proceedings of the European Conference on Computer Vision. pp. 139–155. Springer (2022)
8. Wang, P., Liu, L., Liu, Y., Theobalt, C., Komura, T., Wang, W.: Neus: Learning neural implicit surfaces by volume rendering for multi-view reconstruction. arXiv preprint arXiv:2106.10689 (2021)
9. Yariv, L., Gu, J., Kasten, Y., Lipman, Y.: Volume rendering of neural implicit surfaces. Advances in Neural Information Processing Systems **34**, 4805–4815 (2021)

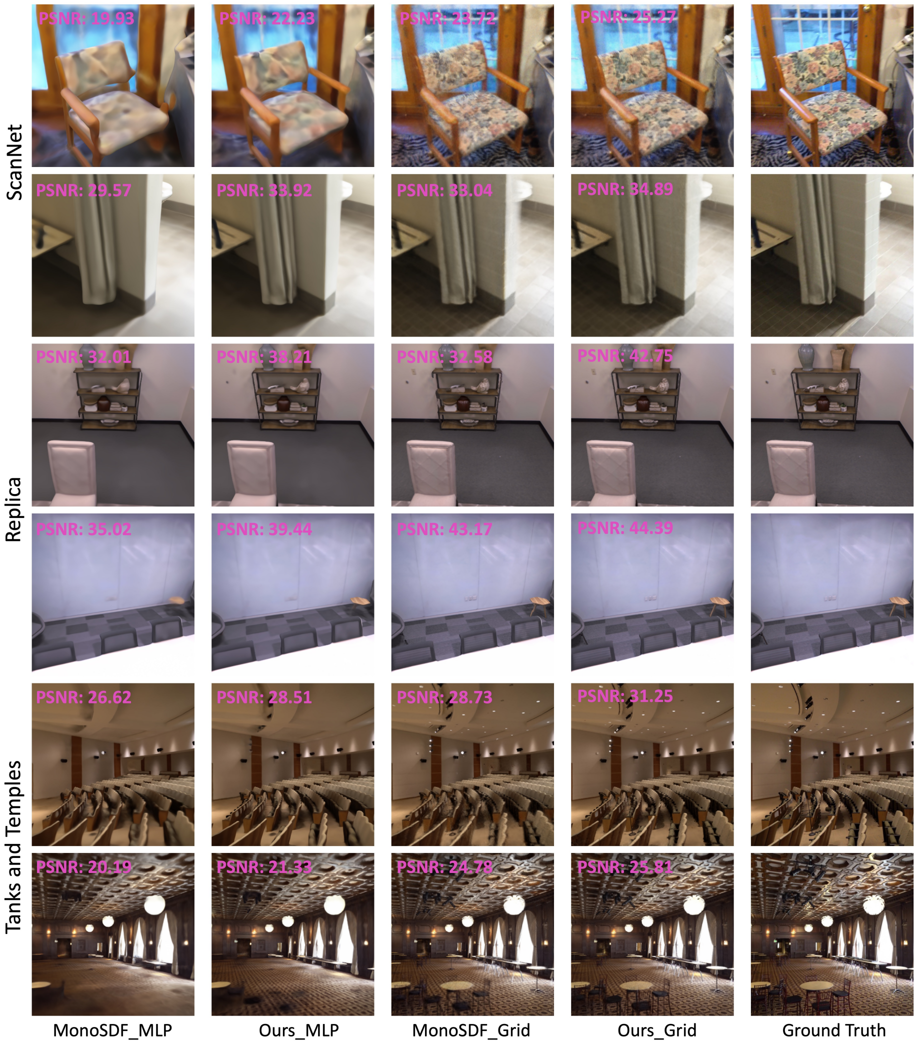

**Fig. S7: Visualization of rendered views.** Our method improves the quality of generated views.

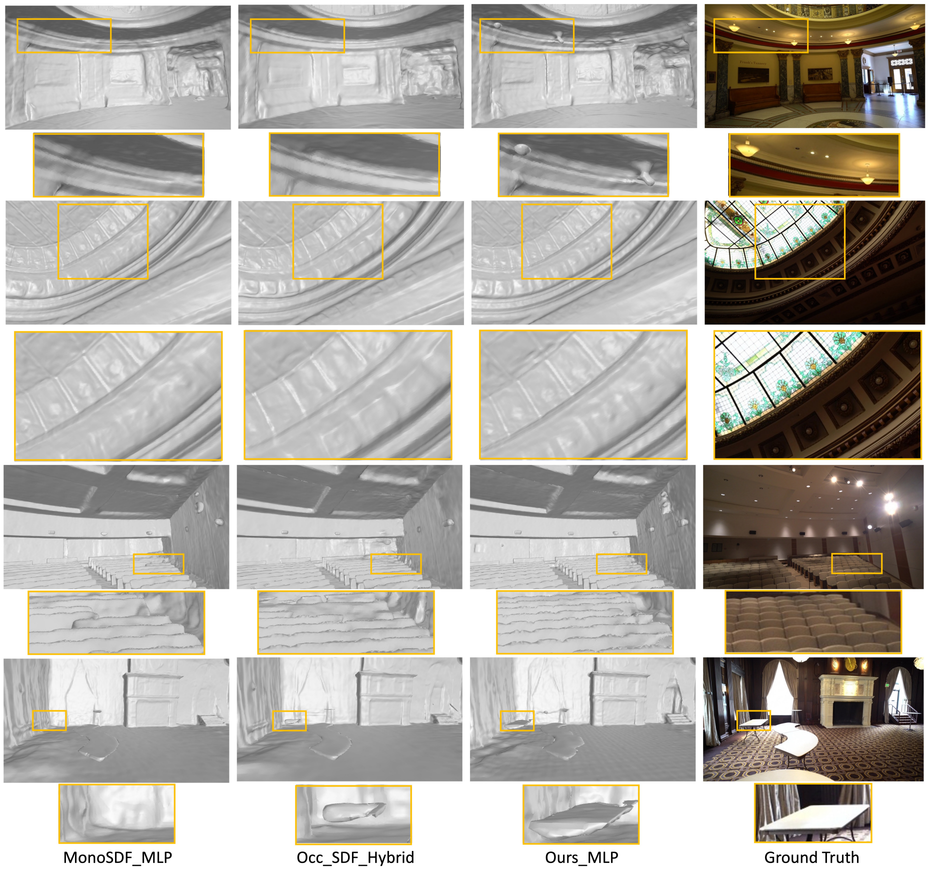

**Fig. S8: Qualitative comparisons on Tanks and Temples.** Compared to other methods, our method can reconstruct more surfaces and recover details.