Supplementary Material for Learning a Room with the Occ-SDF Hybrid: Signed Distance Function Mingled with Occupancy Aids Scene Representation

In this supplementary document, we first describe our network architecture, evaluation metrics, and implementation details in Sec. 1. Then, we present an in-depth analysis of the feature fusion scheme and the Occ-SDF hybrid representation in Sec. 2. Eventually, we report additional quantitative and qualitative results in Sec. 3.

1. Implementation Details

We elaborate on our neural network architecture and optimization.

1.1. Evaluation Metrics

Evaluation metrics used in our work are adapted from the state-of-the-art work, MonoSDF [9], and are defined below. In the following equations, P and P^* are the point clouds sampled from the predicted and ground truth mesh. $\mathbf{n_p}$ is the normal at point $\mathbf{p}$.

Accuracy

$$\textit{Accuracy} = \underset{\mathbf{p}\in P}{\texttt{mean}} \left(\min_{\mathbf{p}^*\in P*} ||\mathbf{p}-\mathbf{p}^*||_1 \right) . \tag{1}$$

Completeness

$$\textit{Completeness} = \underset{\mathbf{p}^*\in P^*}{\texttt{mean}} \left(\min_{\mathbf{p}\in P} ||\mathbf{p}-\mathbf{p}^*||_1 \right) . \tag{2}$$

Chamfer-$\mathcal{L}_1$

$$\textit{Chamfer-}\mathcal{L}_1 = \frac{\textit{Accuracy} + \textit{Completeness}}{2} . \tag{3}$$

Recall

$$\textit{Precision} = \underset{\mathbf{p}\in P}{\text{mean}} \left(\min_{\mathbf{p}^*\in P^*} ||\mathbf{p}-\mathbf{p}^*||_1 < 0.05 \right) . \tag{4}$$

Precision

$$\textit{Recall} = \underset{\mathbf{p}^*\in P^*}{\text{mean}} \left(\min_{\mathbf{p}\in P} ||\mathbf{p}-\mathbf{p}^*||_1 < 0.05 \right) . \tag{5}$$

F-score

$$\textit{F-score} = \frac{2 \times \textit{Precision} \times \textit{Recall}}{\textit{Precision} + \textit{Recall}} . \tag{6}$$

Normal Accuracy

$$\textit{Normal Acc} = \underset{\mathbf{p}\in P}{\text{mean}} \left(\mathbf{n}_\mathbf{p}^T \mathbf{n}_{\mathbf{p}^*} \right) \text{ s.t. } \mathbf{p}^* = \underset{p^*\in P^*}{\text{argmin}} \left\| \mathbf{p}-\mathbf{p}^* \right\|_1 \tag{7}$$

Normal Completeness

$$\textit{Normal Comp} = \underset{\mathbf{p}^*\in P^*}{\text{mean}} \left(\mathbf{n}_\mathbf{p}^T \mathbf{n}_{\mathbf{p}^*} \right) \text{ s.t. } \mathbf{p} = \underset{p\in P}{\text{argmin}} \left\| \mathbf{p}-\mathbf{p}^* \right\|_1 \tag{8}$$

Normal Consistency

$$\textit{Normal Cons.} = \frac{\textit{Normal Acc} + \textit{Normal Comp}}{2} \tag{9}$$

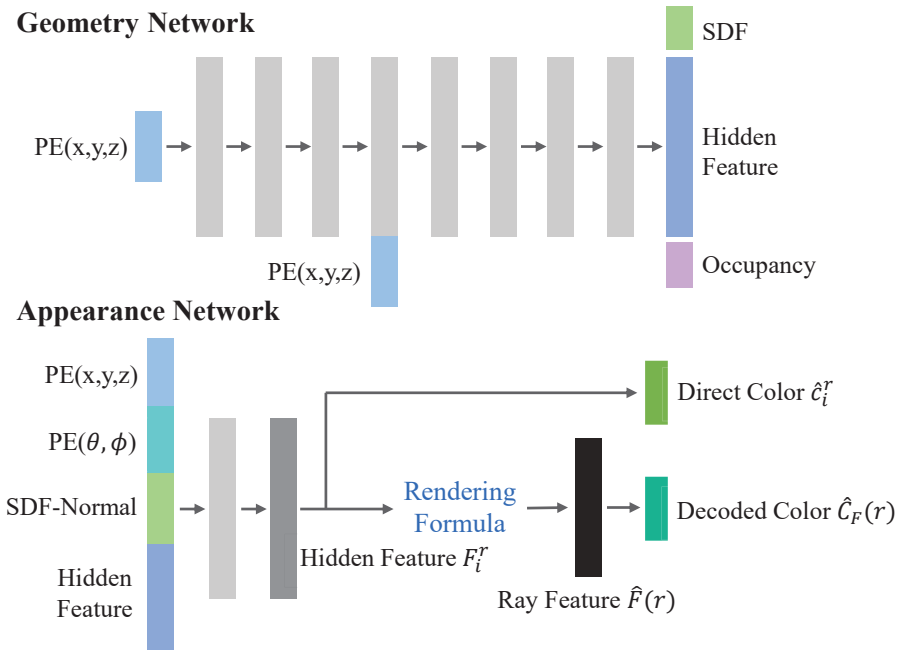


Figure 1. **Visualization of our dual-module network architecture.**

1.2. Network Architecture

As a complementary to Sec. 4 and Sec. 5 of the main paper, we observe from Fig. 1 that our model mainly consists of multi-layer perceptrons (MLPs) and eventually leads to two domain-specific sub-networks: the *Geometry* network and the *Appearance* network.

For the *Geometry* network (see Fig. 1: Geometry Network), it contains 8 linear layers, with each layer having 256 hidden nodes. Given the 3D position (x, y, z) after positional encoding (PE) as inputs, the geometry network outputs SDF value, hidden feature, and occupancy. Additionally, we also apply the skip connection [6] and sphere initialization [3] for training the geometry network.

For the *Appearance* network (see Fig. 1: Appearance Network), we follow the network architecture design in [8, 4, 9]. The inputs are composed of the positional encoded 3D position (PE(x, y, z)), positional encoded view direction (PF(θ, ϕ)), surface normal computed by SDF value (SDF-Normal), and hidden features computed by the geometry network (See Fig. 1). The *Appearance* network outputs two predictions for each point i along a ray r: one is the color vector $\hat{c}_i^r$, and the other is the hidden feature F_i^r. Given the predicted color vector $\hat{c}_i^r$, the target pixel color $\hat{C}(r)$ is rendered by integrating over all points along the ray via the following formula

$$\hat{C}_c(r) = \sum_{i=1}^{M} T_i^r \alpha_i^r \hat{c}_i^r. \tag{10}$$

And the hidden feature F_i^r is used to render the ray feature $\hat{F}(r)$ by

$$\hat{F}(r) = \sum_{i=1}^{M} T_i^r \alpha_i^r F_i^r \ . \tag{11}$$

The ray feature $\hat{F}_r$ is further decoded by a decoder $\mathcal{D}$ to yield the decoded target pixel color,

$$\hat{C}_F(r) = \mathcal{D}(\hat{F}(r)) \ . \tag{12}$$

1.3. Elaboration on Optimization

RGB Reconstruction Loss The purpose of neural rendering is to reconstruct RGB images in the training set, which yield the RGB reconstruction loss as,

$$\mathcal{L}_{\text{rgb}} = \sum_{r \in \mathcal{R}} ||\hat{C}(r) - C(r)||_1 \ . \tag{13}$$

Here $\mathcal{R}$ denotes the sampled rays/pixels in a minibatch and $C(r)$ is the color of the sampled pixel. For a single ray r, we obtain directly rendered colors from the SDF representation $\hat{C}_c^{\text{sdf}}(r)$, indirectly rendered colors with feature rendering from the SDF representation $\hat{C}_F^{\text{sdf}}(r)$. We adopt the same reconstruction loss as Eq. (13), to supervise and train the model.

Eikonal Loss. Following the previous work [3], we also add the Eikonal loss to regularize the SDF values in 3D space

$$\mathcal{L}_{\text{eik}} = \sum_{x \in \mathcal{X}} (||\nabla f_\theta(x)||_2 - 1)^2 \ , \tag{14}$$

where $\mathcal{X}$ are a set of uniformly sampled points together with near-surface points[8]. The Eikonal loss can constrain the distribution of the whole space leading to smooth and natural zero-level set surfaces.

Additional Details. In addition to the RGB reconstruction loss and Eikonal loss, we apply the depth consistency loss and normal consistency loss, as described in the main text. Thus, the overall loss is

$$\mathcal{L} = \mathcal{L}_{\text{rgb}}^{sdf_F} + \mathcal{L}_{\text{rgb}}^{sdf} + \lambda_1 \mathcal{L}_{\text{eik}} + \lambda_2 \mathcal{L}_{\text{depth}}^{\text{occ}} + \lambda_3 \mathcal{L}_{\text{depth}}^{\text{sdf}} + \lambda_4 (\mathcal{L}_{\text{normal}}^{occ} + \mathcal{L}_{\text{normal}}^{sdf}) \ . \tag{15}$$

Here, ^{occ} represents the loss computed by the occupancy-based representation, ^{sdf} represents the loss computed by SDF-based representation, and F indicates the operation of adopting the feature rendering formula to this loss.

The network is optimized using the Adam optimizer with a learning rate of $5e^{-4}$. We set the weights $\lambda_1, \lambda_2, \lambda_3, \lambda_4$ to $0.05, 1, 0.1, 0.05$, respectively, and sample 1,024 rays per iteration and leverage the error-bounded sampling strategy introduced by [8] to sample points along each ray. The entire implementation is conducted in PyTorch.

2. Complementary Analysis

2.1. Feature Rendering Formula

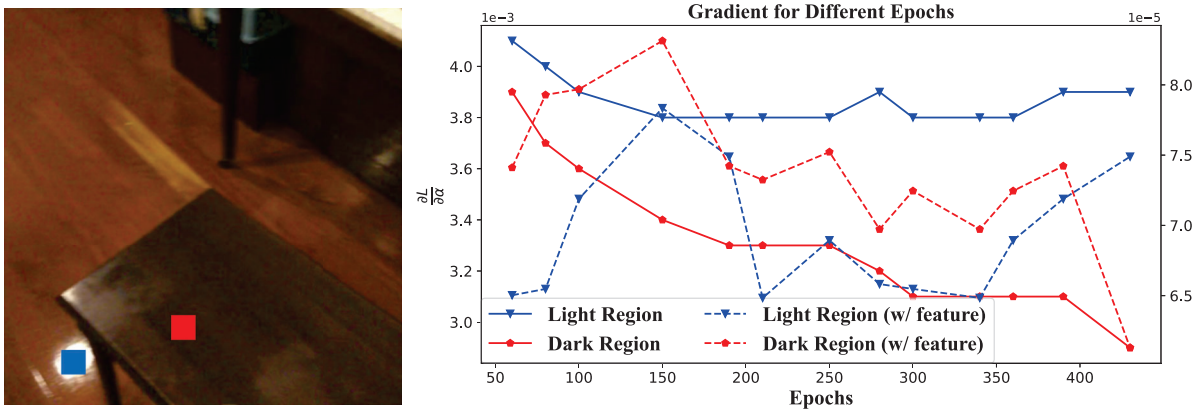
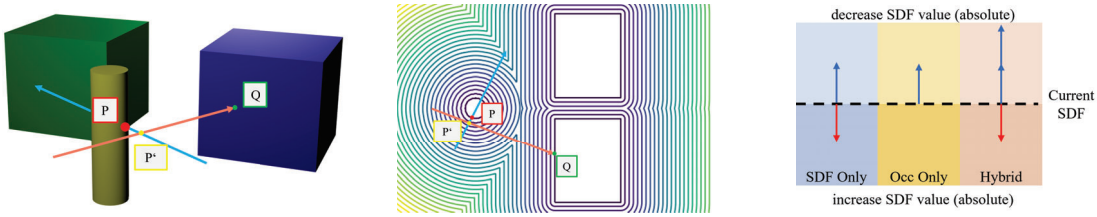


Figure 2. Justification of the influence of color space. For rays sampled in the dark region (red block) and the light region (blue block) separately, as shown on the left, we accordingly present their tendency of the gradient evolution on the right.

To demonstrate the impact of the original color loss on the optimization process, we conducted an experiment where we recorded the gradient norms incurred by rays sampled from dark and light regions separately during the optimization process, as depicted in Fig. 2. To isolate the influence of color loss and rule out the impact of loss magnitude changes during optimization, we set $\frac{\partial \mathcal{L}}{\partial \text{output}} = 1$. The results showed that while the gradient norms in both regions were similar initially, the gradient in the dark region (red solid plot) decreased as the number of epochs increased, while the gradient in the light



(a) Multi Object Scene (b) SDF Distribution (c) SDF Value at Point P

Figure 3. Justification of the effectiveness of the hybrid representation. On the left, it shows the spread-wide scene as the room-level scale. The red ray in the middle directly hits the blue cube and the blue ray hits the yellow cylinder. On the right, it shows the effect of the supervised signal in three different representations. Notably, in this sub-figure, the upper part means to encourage the generation of structure and the lower part means to encourage the disappearance of structure.

region remained stable (blue solid plot), indicating that the optimization process was less affected by the dark regions. This suggested that if the colors of points in these areas were predicted as dark colors, it could result in gradient vanishing effects as analyzed in Eq. (7) in the main paper. However, after introducing the feature rendering scheme, we observed that the gradient norms in dark regions (red dashed plot) had the same magnitude as those in light regions (blue dashed plot), demonstrating the effectiveness of our proposed feature rendering scheme in addressing the gradient dismissing effects caused by the original color loss.

2.2. The Occ-SDF Hybrid Representation

As shown in Fig. 3, we investigate a scenario where the red ray hits a surface point of the large blue cube and the blue ray hits a surface point P on the small yellow cylinder (the red point). As mentioned in Sec. 5 in the main paper, the existence of the large blue cube would enforce the network to produce SDF values ignoring the small cylinder when the geometry prior is adopted to guide the learning. Herein, we present the elaborated analysis of this observation and explain why the proposed hybrid representation can facilitate the reconstruction of thin structures and small objects. Accordingly, we analyze *three* cases: SDF representation only, Occupancy representation only, and our Hybrid representation. Note that at the beginning of optimization, the network is initialized as a hollow sphere[3], where the SDF value is positive inside and negative outside.

SDF Representation Only In the beginning, the concerned surface point P is assigned with a large SDF value, indicating an empty space. During the optimization, the depth/normal supervision that requires the point P to be on the surface will encourage the network to be optimized to reduce the absolute value of its SDF prediction (see Fig. 3(c) the blue up arrow). Assume the yellow cylinder is the single object in a scene, the optimization process will prompt the reconstruction of the object. However, the situation changes if multiple objects interfere with it (*e.g.* the large blue cube). In this case, the depth/normal supervision for points hitting the large blue cube (*e.g.* Q) will enforce the network to be optimized to output large (absolute) SDF values for points in the red ray and away from the point Q, such as point P', to reduce the depth/normal error for Q (*i.e.* minimize $\mathcal{L}_{\text{depth}}$). Besides, to make the predicted values from the network follow the distribution of SDF values (the minimization of $\mathcal{L}_{\text{eik}}$), the network will be optimized to make spatially close points have similar SDF values, requiring the SDF value of points P and P' to be similar, which in turn will enforce the network to output values to increase the SDF (absolute) value of point P (see Fig. 3(c) the red down arrow).

In sum, the direct depth/normal supervision of point P will enforce the network to reduce its SDF (absolute) value, while the indirect depth/normal supervision from other objects will promote the network to increase its SDF (absolute) value. If the indirect supervision outweighs the direct supervision (*i.e.* red down arrow), the yellow cylinder will not be well reconstructed, causing missing structures. This happens when the object of interest (*e.g.* the small thin yellow cylinder) is small, incurring a low frequency of direct supervision in optimization, while other objects are large, incurring a high frequency of indirect supervision.

Occupancy Representation Only The occupancy representation models each point separately and thus is free from interference from other objects. Therefore, the depth/normal loss will encourage the network to be optimized to make the corresponding point P on the surface, yielding a large occupancy value of P and reconstructing the blue cylinder. This can be viewed as reducing the SDF (absolute) value of point P (see Fig. 3(c): the blue up arrow in "Occ only"). However, the occupancy representation doesn't enforce any constraints on the surface and, thus will produce noisy structures that don't

exist (Fig. 1 occupancy in the main paper).

Hybrid Representation The hybrid representation joins the forces of SDF and occupancy representations, aiming to use occupancy representation to help overcome the issues of SDF representation in optimization. Although the loss for point Q still has a negative impact on the optimization of point P. The additional occupancy representation will force the network to predict a large occupancy value for a point P and thus will indirectly regularize the network to predict a small SDF value (see Fig. 3(c): the blue up arrow in "Hybrid"). We admit that this hybrid representation can only alleviate this problem, and our study is more on the empirical side. The fundamental issues arise from the insufficiency of existing neural scene representation, which requires further research efforts.

3. Additional Results

In addition to the experiments provided in the main paper, we present more ablation studies here.

3.1. Parameters Comparison

The proposed techniques, *i.e*. feature rendering scheme and hybrid representation, increase the number of parameters during training. Note that our model doesn't introduce additional parameters during the inference stage compared to the baseline model. To verify that the attained performance gains are not from the enlarged network architecture during training, we perform an ablation study to enlarge the baseline's model size to match that of our model.

	Normal C.↑	Chamfer-$\mathcal{L}_1$ ↓	F-score ↑	# Implicit Network	# Rendering Network	# All
MonoSDF-MLP	92.11	2.94	86.18	0.529012M	0.182278M	0.71129M
+ hybrid	93.22	2.77	90.24	0.529270M	0.182278M	0.711548M (+0.03%)
+ feature	93.01	2.64	91.01	0.529012M	0.248326M	0.777338M (+9.29%)
MonoSDF-MLP†	92.11	2.94	86.18	0.529270M	0.182278M	0.711548M (+0.03%)
MonoSDF-MLP*	92.2	2.90	87.12	0.529012M	0.248326M	0.777338M (+9.29%)
Ours (full model)	**93.35**	**2.58**	**91.25**	0.529270M	0.248326M	0.777596M (+9.32%)

Table 1. Comparison of parameter numbers and performance on the Replica [7] dataset. It shows the number of parameters for different components and their respective impacts. The performance of a larger network is also presented to verify the effectiveness of the proposed constraints. The baseline model is MonoSDF [9]. † indicates that the network has the same structure as the hybrid representation, while * indicates that the network has the same structure as the feature rendering scheme.

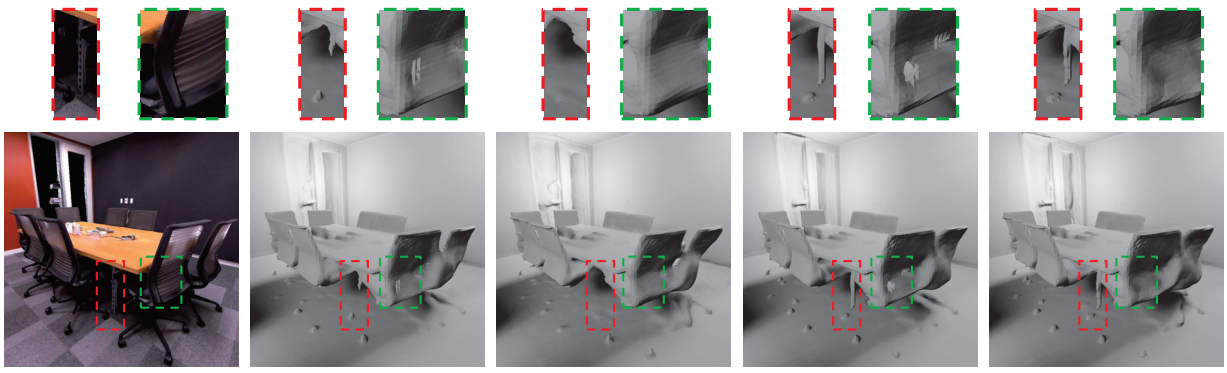


Figure 4. **Ablation of our proposed methods on Replica dataset.** The red dashed boxes show the reconstruction results on detailed structures and the green dashed boxes show the reconstruction results in low intensities regions. It is evident that the table legs within the red dashed boxes can be substantially reconstructed through hybrid representation. Moreover, the original representation yields satisfactory reconstruction results in regions with low intensity within the green dashed boxes, owing to the additional geometry constraints; however, it still results in empty holes on the chair back. Our feature rendering scheme adeptly fills these holes to attain superior results.

Table 1 illustrates a comparison of the number of parameters and the effectiveness of our proposed methods on the Replica [7] dataset. First, the hybrid representation only incurs a marginal increase of parameters (0.03%) during training yet significantly improves the performance (F-score $+4.06$). In comparison, if we only increase the number of parameters of the baseline model (MonoSDF-MLP†), the performance is still similar to the baseline model. Second, the feature rendering scheme introduces 9.29% of the parameters during training and largely boosts the performance (F-score $+4.83$). On the other

hand, our feature rendering scheme introduces an extra layer to decode the concealed feature F_i^r, as depicted in Figure 1. To illustrate that the enhancement in performance is not from a larger network, we design a new architecture, MonoSDF-MLP*, with the same number of parameters as the feature rendering scheme. As displayed in Table 1, this network exhibits only a marginal improvement in performance (F-score $+0.92$). This substantiates the effectiveness of our proposed feature rendering scheme. When combined with the hybrid representation and feature rendering scheme, our full model further enhances the performance (F-score $+5.07$), which is not attributed to a larger network structure.

We present the visualized results with different components in Figure 4. The initial MonoSDF exhibits unsatisfactory reconstruction results in areas with low intensity (i.e., the black chair with perforations in the green box) and detailed structures (i.e., absent table legs in the red box). The utilization of the feature rendering scheme enables the filling of empty perforations (i.e., green box in +feature), while the hybrid representation allows for the reconstruction of table legs (i.e., red box in +hybrid). Our full model adeptly addresses both issues and attains the best results.

3.2. Direct SDF Supervision

As illustrated in Figure 5 of the main paper, the primary problem arose from the conversion function (Eq.(2) and Eq.(3) in the main paper) and the rendering formula (Eq. (4) in the main paper). While the most straightforward approach is to supervise the SDF value for each point directly, the inconsistent scale of estimated depth can lead to suboptimal TSDF-Fusion outcomes. To explore the effects of direct SDF supervision signals, we initially train an implicit network with MonoSDF settings and calculate the scale and shift for each frame. We then use these computed values to align the estimated depth maps and subsequently use TSDF-Fusion[5] to construct the mesh.

After reconstruction, we can get the pseudo-SDF value from the reconstructed mesh. We use the original MonoSDF structure and add the SDF supervision to each point on the ray r following

$$\mathcal{L}_{\text{sdf}} = \sum_{r \in R} \sum_{i=1}^{m} ||SDF_i^r - \hat{SDF}_i^r||_1 \ , \tag{16}$$

As illustrated in Figure 5, while the implicit reconstruction method can partially align the estimated depth, the resulting reconstruction still exhibits multi-layered walls and rough surfaces, which is caused by imperfect scale consistency and floaters due to erroneous depth estimations. Consequently, even though TSDF-Fusion can capture detailed structures, it yields poor qualitative results as presented in Table 2.

Table 2 shows that the original aligned TSDF-Fusion results have poor reconstruction performance. Therefore, directly using these results to supervise the implicit network does not result in a good performance (MonoSDF + SDF). Additionally, when depth is used as a supervised signal along with SDF (MonoSDF* + SDF), the performance is still worse than the original framework, which demonstrates that directly leveraging imperfect SDF signals to supervise the geometry network has negative influences on reconstruction results. Figure 6 further illustrates that an incorrect supervised SDF signal can cause the implicit geometry to erroneously incorporate the structure from TSDF-Fusion meshes.

We posit that the inadequate performance of direct SDF supervision primarily stems from a heavy dependence on imperfect geometry estimation results and simplistic supervised strategies. With the utilization of a superior geometry prior, such as an RGB-D camera, the reconstruction results are expected to improve considerably [1].

	Normal Cons.↑	Accuracy	Completeness	Chamfer-$\mathcal{L}_1$ ↓	Precision	Recall	F-score ↑
TSDF-Fusion	82.05	6.20	4.91	5.55	53.26	77.79	63.05
MonoSDF + SDF	91.47	3.52	4.95	4.23	83.67	77.24	80.28
MonoSDF* + SDF	91.72	3.69	3.85	3.77	82.21	80.85	81.41
Ours	**93.35**	**2.56**	**2.66**	**2.58**	**92.08**	**90.45**	**91.25**

Table 2. Quantitative assessment on Replica dataset with SDF supervision.

4. Limitation Analysis

In this work, we mainly explore the impact of room-level neural surface reconstruction. Unlike the single-object scenario, the occlusion problem in the scene may make some small objects or thin structures difficult to optimize. This interesting yet challenging phenomenon drives us to investigate an empirical solution, the Occ-SDF hybrid representation, which leverages the properties of occupancy to aid SDF representation. We admit that our analysis and representation still fall on the empirical side and believe the identified problems might arise from fundamental issues of the scene representation itself. We hope our empirical study could inspire more research in this direction.

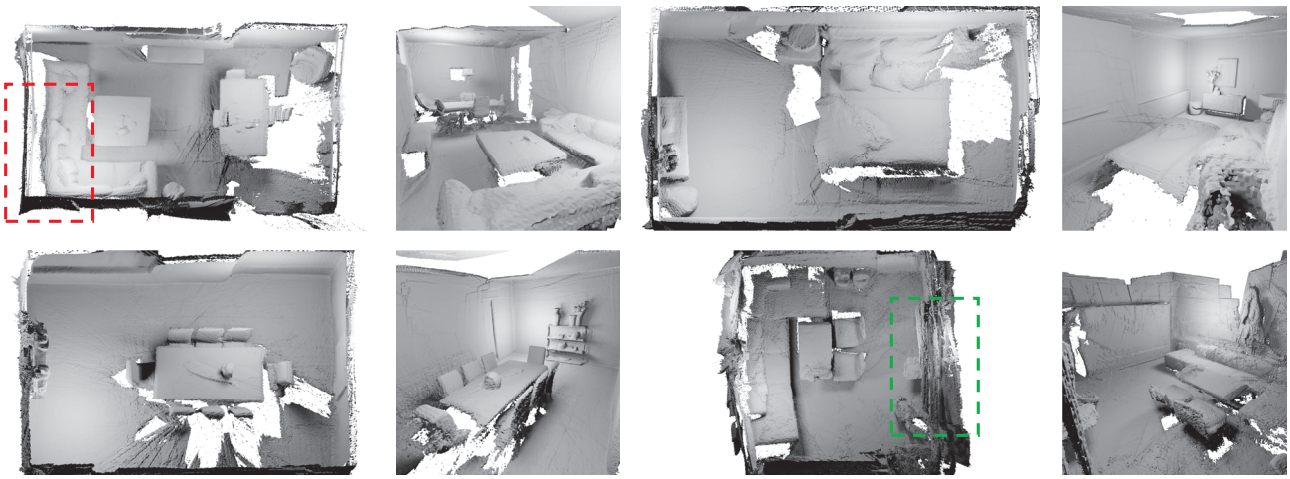


Figure 5. Visualization of TSDF Results. Despite using an implicit network to align the depth scale for each frame, the TSDF-Fusion results still exhibit multi-layer surfaces and inaccurate reconstruction (red and green dashed boxes).

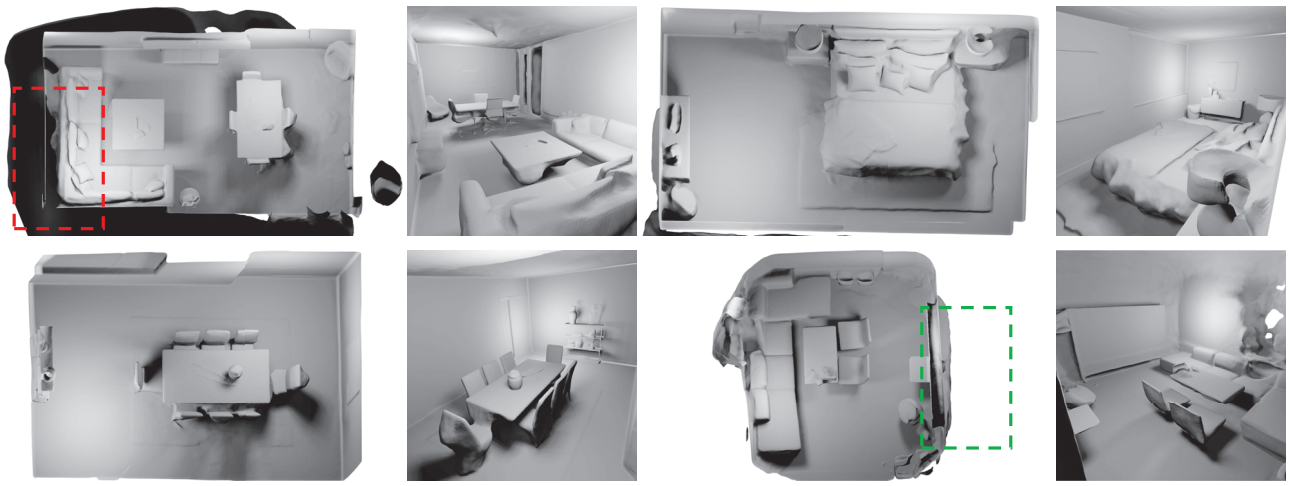


Figure 6. Visualization of Reconstruction Results with SDF Supervision. These reconstructed results also possess multiple layers of surfaces (red and green dashed boxes), demonstrating their susceptibility to inheriting erroneous estimations from an imperfect TSDF-Fusion mesh.

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 (CVPR)*, pages 6290–6301, June 2022. 6

[2] Angela Dai, Angel X Chang, Manolis Savva, Maciej Halber, Thomas Funkhouser, and Matthias Nießner. Scannet: Richly-annotated 3d reconstructions of indoor scenes. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5828–5839, 2017. 8

[3] Amos Gropp, Lior Yariv, Niv Haim, Matan Atzmon, and Yaron Lipman. Implicit geometric regularization for learning shapes. In *International Conference on Machine Learning*, pages 3789–3799. PMLR, 2020. 2, 3, 4

[4] 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, 8

[5] Paul Merrell, Amir Akbarzadeh, Liang Wang, Philippos Mordohai, Jan-Michael Frahm, Ruigang Yang, David Nistér, and Marc Pollefeys. Real-time visibility-based fusion of depth maps. In *2007 IEEE 11th International Conference on Computer Vision*, pages 1–8. Ieee, 2007. 6

[6] Michael Niemeyer, Lars Mescheder, Michael Oechsle, and Andreas Geiger. Differentiable volumetric rendering: Learning implicit 3d representations without 3d supervision. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*,

VolSDF [8] Manhattan-SDF [4] MonoSDF [9] Ours Ground Truth [2]

Figure 7. **Qualitative comparison on ScanNet.** Different views for each scene are presented, revealing that our method can lead to a higher-fidelity reconstruction quality compared with state-of-the-art neural implicit methods.

pages 3504–3515, 2020. 2

[7] Julian Straub, Thomas Whelan, Lingni Ma, Yufan Chen, Erik Wijmans, Simon Green, Jakob J. Engel, Raul Mur-Artal, Carl Ren, Shobhit Verma, Anton Clarkson, Mingfei Yan, Brian Budge, Yajie Yan, Xiaqing Pan, June Yon, Yuyang Zou, Kimberly Leon, Nigel Carter, Jesus Briales, Tyler Gillingham, Elias Mueggler, Luis Pesqueira, Manolis Savva, Dhruv Batra, Hauke M. Strasdat, Renzo De Nardi, Michael Goesele, Steven Lovegrove, and Richard Newcombe. The Replica dataset: A digital replica of indoor spaces. *arXiv preprint arXiv:1906.05797*, 2019. 5

[8] Lior Yariv, Jiatao Gu, Yoni Kasten, and Yaron Lipman. Volume rendering of neural implicit surfaces. *Advances in Neural Information Processing Systems*, 34:4805–4815, 2021. 2, 3, 8

[9] Zehao Yu, Songyou Peng, Michael Niemeyer, Torsten Sattler, and Andreas Geiger. Monosdf: Exploring monocular geometric cues for

neural implicit surface reconstruction. *arXiv preprint arXiv:2206.00665*, 2022. 1, 2, 5, 8