Supplementary Material for PMVC

In this supplementary document, we discuss the architectural and implementation details in Section 6. Next, in Section 7, we provide additional quantitative and qualitative results across the various datasets we experimented with, as well as our scene reconstruction results. Finally, we discuss the potential negative impact of this work in Section 8.

6. Implementation Details

In this section, we first describe additional details regarding our parameterizations and optimization in Section 6.1. Next, we present an overview of two different architectures for Feature Extractor in Section 6.2 and provide details of the semantic Cues in Section 6.3, and discuss evaluation metrics in Section 6.4.

6.1. Parameterizations

In this experiment, we utilized PyTorch [32] as our experimental framework and employed Adam [17] as our optimizer. Our code was initially set up using the framework from MonoSDF [56], and we adhered to their learning rate 5e-4. Empirically, we established λ_1, λ_2, λ_3, λ_4, λ_5 and γ at 0.1, 0.1, 0.05, 0.5, 0.04 and 0.01 respectively. For feature extraction, we utilized VGG-19 [39] model. All experiments were executed on a single RTX3090 GPU. To align with our task requirements, we make minor modifications to the NYU labels, reducing them to 38 classes representing indoor scene objects and eliminating ambiguous labels such as "other furniture". For additional details on semantic map generation, please refer to section 6.3.

Implement details. We implement our Neural Implicit Representation architecture using two MLPs. Each MLP corresponds to a 256-dimension feature. Our Neural Renderer network outputs both an RGB intensity and a label. As previously mentioned in the paper, we utilize MonoSDF for geometry initialization. In addition to this, we deploy our pipeline to generate a semantic map for each frame during the pre-processing phase 6.3. We optimize our model over 200,000 iterations. In terms of computational time, optimizing a single scene using the full model on a single NVIDIA RTX 3090 GPU takes around 24 hours. The feature constraint process requires approximately 18 hours. Meanwhile, adaptive sampling and semantic constraint take around 15 hours and 12 hours, respectively.

6.2. Feature extractors

In this section, we present results obtained from two distinct feature extractors, namely ResNet-18 and VGG-19. We carried out the ablation experiments on ScanNet, and the data presented in Table S5 discovers the differences between feature extractors. Additionally, it reveals a consistent improvement across the F-score.

Scene	Acc↓	Comp↓	Prec↑	Recall↑	F-score↑	Chamfer-dist↓
MonoSDF	**0.035**	0.048	**79.9**	68.1	73.3	0.042
+ ResNet-18	0.041	0.043	76.4	72.5	74.3	0.042
+ VGG-19	0.040	**0.042**	79.8	**74.9**	**77.4**	**0.041**

Table S5. Two Feature Extractors comparison

6.3. Semantic Map Pipeline

In this section, we introduce our new semantic generation pipeline, as shown in Figure S5, which outperforms other methods. We also conduct several studies to verify the effectiveness of our method. These include comparing with the Manhattan fine-tuned model [11] (as presented in Table S6) and integrating our pipeline into the NeuRIS [47] framework (Table S7). We evaluate our semantic priors on ScanNet.

Thanks to the powerful zero-shot capability of the SAM model [18], we can perform pixel-level segmentation of arbitrary scenes. However, SAM does not possess label classes, which is a problem we sought to address. To rectify this, we first employed the method outlined in [21] for each image, then used prompt text hints to generate bounding boxes (BBox). Subsequently, we used SAM to execute pixel clustering segmentation within each BBox. To better adapt to our task, we eliminated semantically ambiguous portions of the NYU label, such as 'other furniture' and 'other structures'. As a result, our final label classes consist of 38 distinct categories. For the objects that are not included in the 38 classes, we define them as 'unknown', which will not contribute to our semantic optimization.

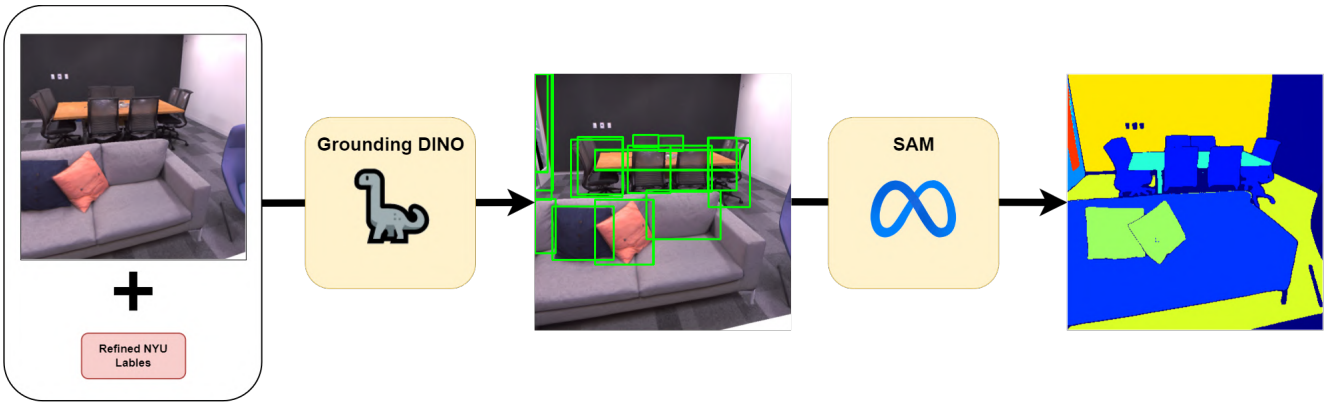


Figure S5. In this pipeline, we utilize the concept of Grounding DINO and incorporate modified NYU labels as our textual cues. Initially, we employ these cues to generate bounding boxes for each class. Subsequently, we apply the SAM (Segment Anything Model) to obtain pixel-wise semantic maps. This approach allows us to effectively map the textual information to the corresponding regions in the image, providing a detailed and accurate representation of the semantic content.

Model	Acc↓	Comp↓	Prec↑	Recall↑	F-score↑	Chamfer-dist↓
MonoSDF	**0.035**	0.048	**79.9**	68.1	73.3	0.042
+ semantic cues (DeepLabV3)	0.040	0.042	77.0	73.5	75.1	0.041
+ semantic cues (Ours)	0.040	**0.041**	79.2	**75.6**	**77.3**	**0.040**

Table S6. Two pipeline comparison

Model	Acc↓	Comp↓	Prec↑	Recall↑	F-score↑	Chamfer-dist↓
NeuRIS	0.050	0.049	71.7	66.9	69.2	0.050
+ semantic cues (DeepLabV3)	0.048	0.048	72.7	67.7	70.1	0.048
+ semantic cues (Ours)	**0.044**	**0.047**	**75.7**	**69.9**	**72.7**	**0.046**

Table S7. Semantic constraints experiments on NeuRIS

6.4. Evaluation Metrics

In line with prior research, we adopted several evaluation metrics to assess the quality of our reconstruction. For the ScanNet dataset, our report features Accuracy, Completeness, Chamfer Distance, Precision, Recall, and F-score. In contrast, for the Replica dataset, we present the Normal Consistency, Chamfer Distance, and F-score. Detailed definitions of these evaluation metrics are specifically provided in Table S8.

7. Additional Results

This section provides more qualitative and quantitative comparison results for the Replica (Figure S8) and ScanNet (Figure S7) datasets. In addition, we demonstrate the full evaluation metrics mentioned in the main paper (Table S9) and discuss the lack of a significant difference between the ACC scores of previous methods and our full model on ScanNet. Lastly, we provide some rendered images to show that our methods reduce the dependency on the performance of pre-trained models (Figure S9).

Performance discussion. Upon closer examination (Figure S6), we found that the ground truth provided by ScanNet tends to be quite noisy. For instance, we observed missing objects that should be present in the ground truth (as illustrated in Supplementary Figure S5). This noise and incompleteness in the ground truth might affect the evaluation metrics, potentially leading to the observed inconsistent improvement in the ACC performance between our result (0.038) and MonoSDF's result (0.035). Notably, this issue does not arise in the synthetic dataset, Replica.

Metric	Definition
Acc	$\underset{p \in P}{\text{mean}}(\min_{p^* \in P^*} ||p - p^*||_1)$
Comp	$\underset{p^* \in P^*}{\text{mean}}(\min_{p \in P} ||p - p^*||_1)$
Chamfer	$\frac{\text{Acc}+\text{Comp}}{2}$
Precision	$\underset{p \in P}{\text{mean}}(\min_{p^* \in P^*} ||p - p^*||_1 < 0.05)$
Recall	$\underset{p^* \in P^*}{\text{mean}}(\min_{p \in P} ||p - p^*||_1 < 0.05)$
F-score	$\frac{2 \cdot \text{Precision} \cdot \text{Recall}}{\text{Precision}+\text{Recall}}$
Normal-Acc	$\underset{p \in P}{\text{mean}}(n_p^T n_{p*})$ s.t. $p^* = \underset{p^* \in P^*}{\text{argmin}} ||p - p^*||_1$
Normal-Comp	$\underset{p^* \in P^*}{\text{mean}}(n_p^T n_{p*})$ s.t. $p = \underset{p \in P}{\text{argmin}} ||p - p^*||_1$
Normal-Consistency	$\frac{\text{Normal-Acc}+\text{Normal-Comp}}{2}$

Table S8. Evaluation Metrics. We present the evaluation metrics along with their definition, which we employ to assess the quality of reconstruction. P and P^* represent the point clouds obtained from the predicted and the actual mesh, respectively. n_p stands for the normal vector at the point p.

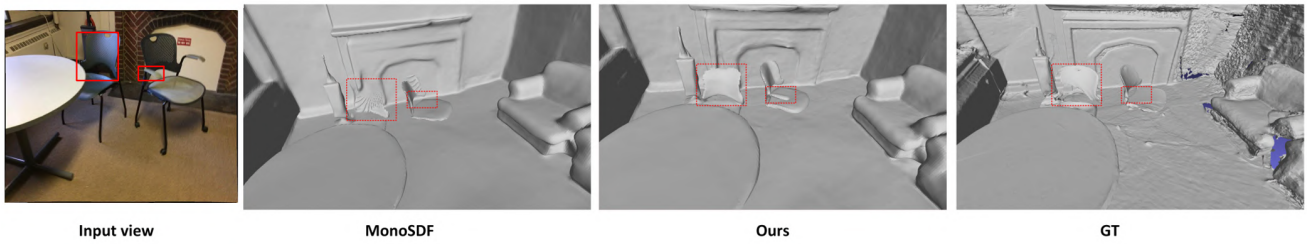


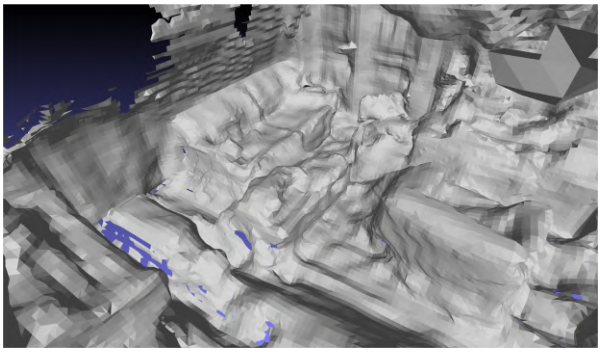
Figure S6. By examining the input views, it is evident that the chair possesses handles and a complete back part. Surprisingly, this crucial information is missing in the provided ground truth (GT). In contrast, our proposed method excels in reconstructing this missing part, highlighting the capability of our approach in capturing and reproducing fine details accurately.

Metric	Acc↓	Comp↓	Chamfer-L1 ↓	Prec↑	Recall↑	F-score↑
COLMAP [36]	0.047	0.235	0.141	71.1	44.1	53.7
UNISURF [30]	0.554	0.164	0.359	21.2	36.2	26.7
NeuS [48]	0.179	0.208	0.194	31.3	27.5	29.1
VolSDF [55]	0.414	0.120	0.267	32.1	39.4	34.6
Manhattan [11]	0.072	0.068	0.070	62.1	58.6	60.2
NeuRIS [47]	0.050	0.049	0.050	71.7	66.9	69.2
MonoSDF [56]	**0.035**	0.048	0.042	79.9	68.1	73.3
Ours	0.038	**0.039**	**0.038**	**81.5**	**77.4**	**79.4**

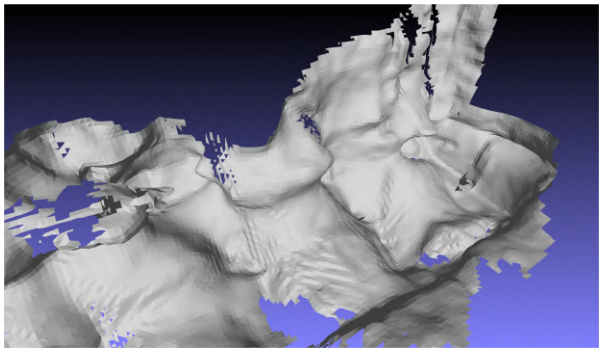
Table S9. Full results on ScanNet dataset.

8. Societal Impact

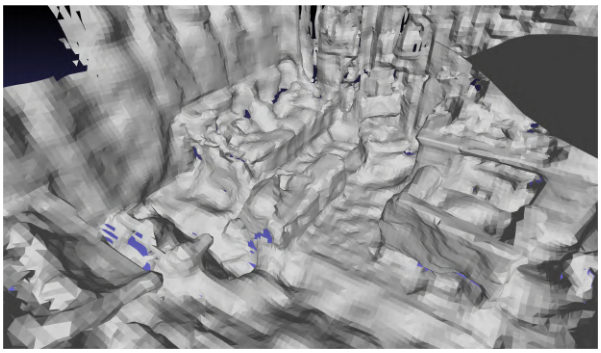
Our proposed method has the potential for significant improvements in 3D reconstruction from multiple viewpoints, which can be applied to virtual reality or greatly reduce the modeling time for designers. However, there are some drawbacks to consider. One drawback is that our approach requires relatively dense inputs; otherwise, multi-view consistency is hard to obtain. Besides, we did not impose additional constraints on the reconstruction process, which may raise privacy concerns when applied to indoor scene reconstruction. Additionally, our training time is relatively long, resulting in increased power consumption. However, further engineering improvements may address this environmental issue in the future.



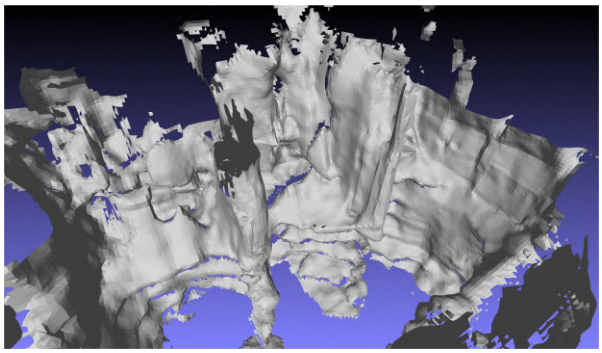
(a) NeuS, Scene 1



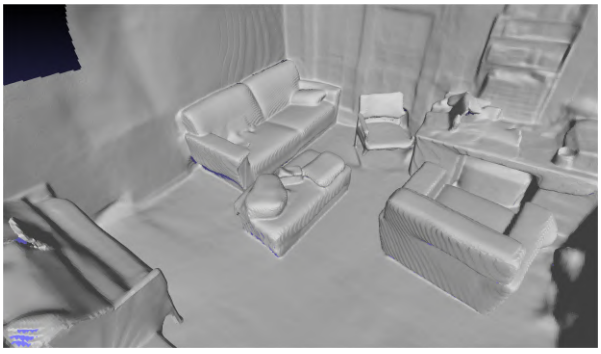
(b) NeuS, Scene 2



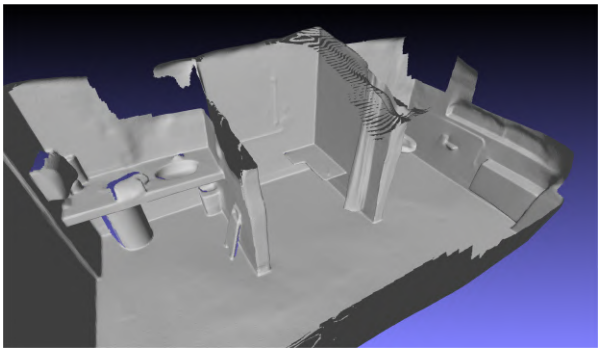
(c) Volsdf, Scene 1



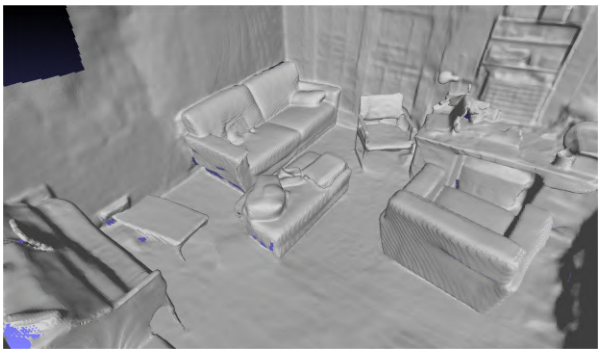
(d) Volsdf, Scene 2



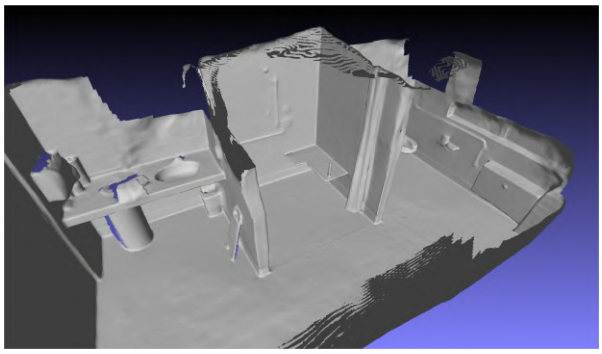
(e) Monosdf, Scene 1



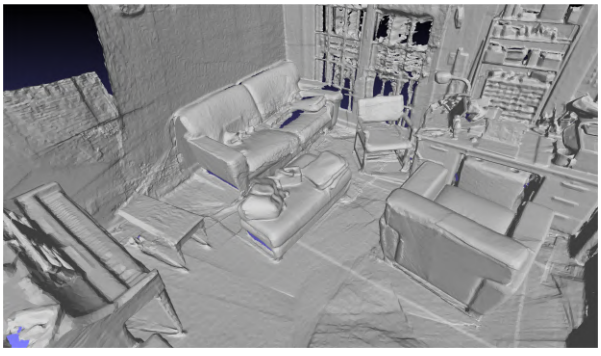
(f) Monosdf, Scene 2



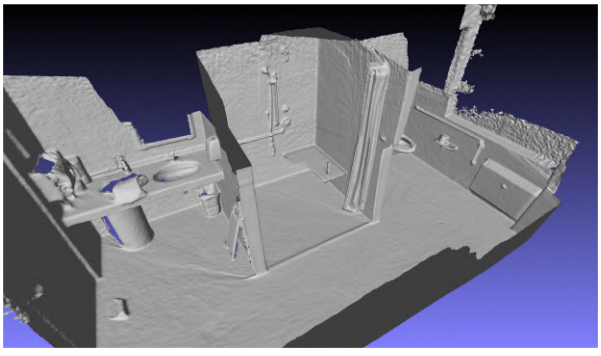
(g) Ours, Scene 1



(h) Ours, Scene 2



(i) ground truth, Scene 1



(j) ground truth, Scene 2

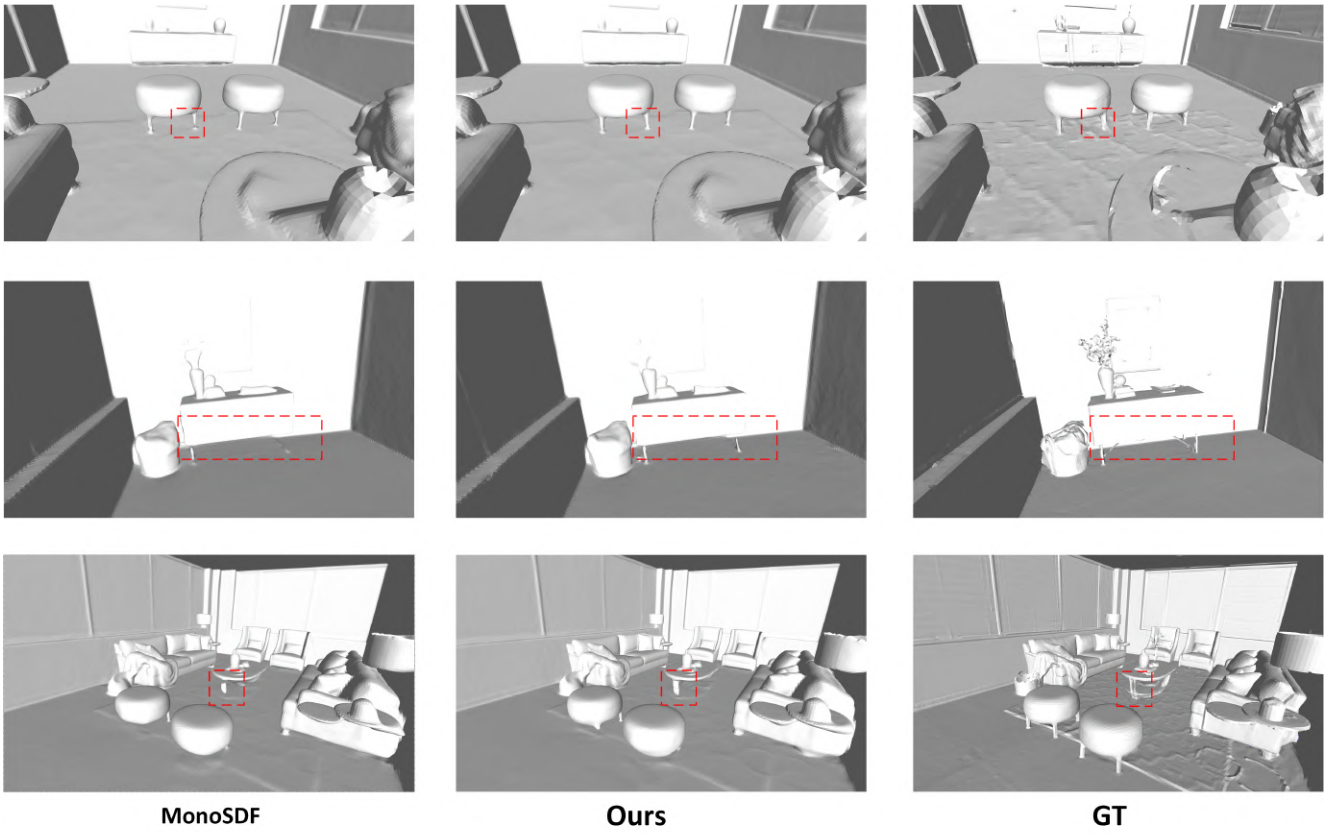
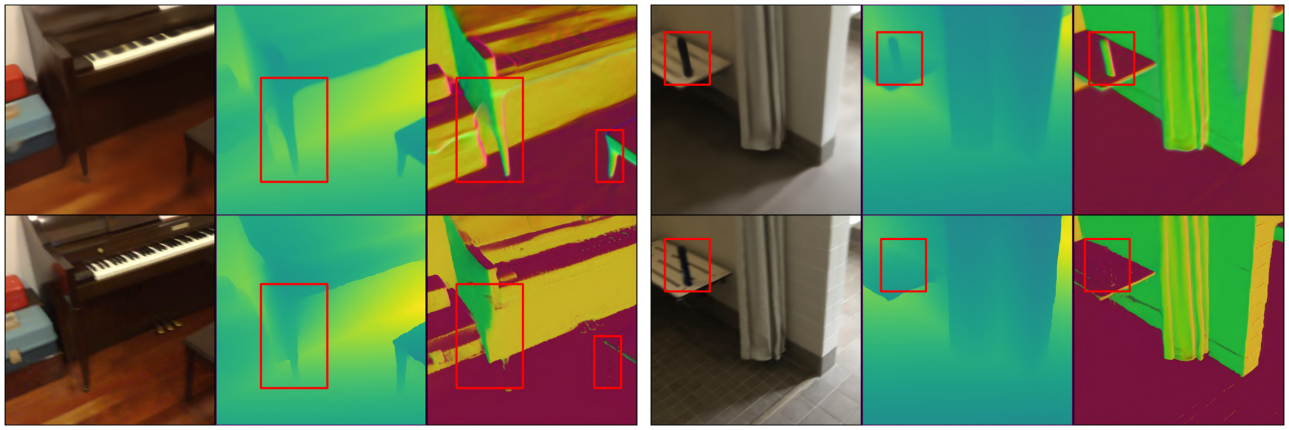


Figure S8. We compare our proposed technique with MonoSDF and the ground truth. As highlighted by the rectangles, our technique shows improvements. We can observe that, compared to previous methods that are only using pre-trained models, our technique reconstructs fine detail well.



(a) case1: pseudo ground truth shows inconsistency

(b) case2: pseudo ground truth provides negative guidance

Figure S9. Rendered results from the ScanNet dataset are presented. The second row shows the RGB image used as our input. The depth map and normal map were estimated using pre-trained models. We observed that the pseudo-ground truth does not always help the model understand scenes due to the potential limitations of the pre-trained models. In contrast, our rendering results exhibit superior details that help mitigate this issue.