Single-view 3D Scene Reconstruction with High-fidelity Shape and Texture

Supplementary Material

In Sec. S1, we present how we prepare the training data on the 3D-FRONT [19] and Pix3D [69] datasets. In Sec. S2, we report more implementation details, including model architecture, learning curriculum, and training strategies. Sec. S3 presents more experimental details, evaluation metrics, results, and failure case discussion. For a more comprehensive view of the qualitative results, we recommend referring to the supplementary video with detailed visualizations and animations.

S1. Training data preparation

S1.1. Datasets and splits

3D-FRONT [19] contains synthetic and professionally-designed indoor scenes populated by high-quality textured 3D models from 3D-FUTURE [19]. Following Liu *et al*. [41], we use about 20K scene images for training and validation and report quantitative evaluation on 2000 images in 8 different categories. Pix3D [69] provides real-world images along with corresponding 3D furniture models that are aligned with the images in 9 object categories. In the splits from Nie *et al*. [54] and [87], there is a significant overlap of objects between the training and testing split. Consequently, we follow Liu *et al*. [41] to employ a split without overlapping objects.

S1.2. Fixing CAD models

One major issue with 3D-FRONT [19] dataset is that the majority of the CAD models are not watertight, which hinders the learning of SDF as the neural implicit surface representations in our framework. To mitigate this issue, we first utilize the automatic remesh method following Liu *et al*. [41] to transform the non-watertight models into watertight ones with a more evenly-distributed topology. Additionally, we have noticed that some models, primarily beds and sofas, lack a back or underside, confusing the model when learning 3D object priors. Consequently, we have manually fixed a total of 1734 models to resolve this issue.

S1.3. SDF supervision

In our framework, we employ supervision from both explicit 3D shapes and volume rendering of color, depth, and normal images. The direct 3D supervision focuses on minimizing the differences between the predicted and actual SDF values of the sampled points. To generate ground-truth SDF values for a given CAD model, we voxelize its 3D bounding box in a normalized coordinate system with a resolution of 64. The SDF value for each voxel grid center is calculated as the distance to the mesh surface. The SDF value is positive if the point is outside the surface and negative if inside. During training, we obtain the ground-truth SDF value for the query points using trilinear interpolation.

S1.4. Monocular cues rendering

To further alleviate the ambiguities in recovering 3D shapes from single-view inputs, we follow Yu *et al*. [84] to exploit monocular depth, normal, and segmentation cues to facilitate the training process. However, since these images are not available in the 3D-FRONT [19] dataset, we render them using the 3D scans of the scene, 3D CAD models of the objects, and the camera's intrinsic and extrinsic parameters provided in the dataset. The Pix3D [69] dataset offers instance segmentation but lacks depth and normal images. Since rendering is impossible, we utilize the estimated depth and normal maps as the pseudo-ground-truth from state-of-the-art estimator [17]. Note that the depth, normal, and segmentation information is solely used during the training stage to guide the model's learning process, and none is required during the inference stage. This ensures that our model remains flexible and applicable to various scenarios.

S2. Technical details

S2.1. Model architecture

For the implicit network, we use an 8-layer MLP with hidden dimension 256. We implement the rendering network with a 2-layer MLP with hidden dimension 256. We use Softplus activation for the implicit network and Sigmoid activation for the rendering network. We use a ResNet34 backbone pre-trained on ImageNet as the image encoder following Yu *et al*. [83]. We use positional encoding γ from NeRF [51] for the spatial coordinates, with $L = 6$ exponentially increasing frequencies:

$$\begin{aligned}\gamma(x) = (&\sin(2^0\omega\mathbf{x}), \cos(2^0\omega \\ &\sin(2^1\omega\mathbf{x}), \cos(2^1\omega\mathbf{x}), \\ &\dots, \\ &\sin(2^{L-1}\omega\mathbf{x}), \cos(2^{L-1}\omega\mathbf{x}))\end{aligned} \tag{S1}$$

S2.2. Learning curriculum

As discussed in the paper, we propose a two-stage learning curriculum to effectively employ supervision from both 3D shapes and volume rendering. In Stage One, the loss weight is set to be 1 for the 3D supervision $\mathcal{L}_{3D}$ and 0 for the rest of the losses. In Stage Two, we linearly increase the loss weights for color, depth, and normal supervision while

maintaining a constant weight for the 3D supervision loss. We utilize the $\mathcal{L}_{3D}$ loss curve when training with 3D supervision only to determine the suitable value for λ_0. Specifically, once the training approaches convergence according to the SDF loss curve, we identify the epoch at this point as the suitable λ_0 value. In our paper, we choose $\lambda_0 = 150$. Slight deviations below or above this threshold have minimal impact. However, significantly reducing the value, such as $\lambda_0 = 0$ or $\lambda_0 = 70$, results in substantial degradation of performance because early injection of 2D supervision may affect the 3D shape learning due to the shape-appearance ambiguity. More discussion can be referred to the ablation experiments in Sec. 4.1.

S2.3. Training strategy

During training, the image, depth, and normal images have the same resolution of 484×648. The implicit network takes 3D points as input in canonical coordinate system to ease the learning of reconstructing indoor objects with implicit representations. The volume rendering is performed along the sampled points in the camera coordinates to calculate the color, depth, and normal values for all the pixels in a minibatch. We sample 64 rays per iteration and apply the error-bounded sampling strategy introduced by Yariv *et al*. [82]. We additionally apply 3D supervision to another 30,000 points uniformly sampled near the ground-truth surfaces (set $\mathcal{X}$). Our model is trained for 400 epochs on 4 NVIDIA-A100 GPUs with a batch size of 96. We implement our method in PyTorch [60] and use the Adam optimizer [37]. The learning rate is initialized as 1e-3 and decays by a factor of 0.2 in the 330^{th} and 370^{th} epochs.

S3. More experiment details and results

S3.1. Evaluation metrics

3D object reconstruction Following Wang *et al*. [75] and Mescheder *et al*. [50], we adopt Chamfer Distance (CD), F-Score and Normal Consistency as the metrics to evaluate 3D object reconstruction. Following prior work [41], we proportionally scale the longest edge of reconstructed objects to $2m$ to calculate CD and F-Score. After mesh alignment with Iterative Closest Point (ICP) [4], we uniformly sample points from our prediction and ground-truth. CD is calculated by summing the squared distances between the nearest neighbor correspondences of two point clouds after mesh alignment. The values of CD are reported in units of 10^{-3}. We calculate precision and recall by checking the percentage of points in prediction or ground-truth that can find the nearest neighbor from the other within a threshold of $2mm$. F-Score [38] is the harmonic mean of precision and recall on in prediction and ground-truth. Finally, to measure how well the methods can capture higher-order information, the normal consistency score is computed as the mean absolute dot product of the normals in one mesh and the normals at the corresponding nearest neighbors in the other mesh after alignment.

Depth and normal estimation We adopt the L1 error for depth estimation and utilize both L1 and Angular errors for normal estimation following Eftekhar *et al*. [17]. Since the baseline methods [17, 86] estimate relative depth values rather than absolute ones, we first align the estimated depth values with the ground-truth values to the range of $[0, 1]$ using the approach outlined in Eftekhar *et al*. [17]. After the alignment, we compute the L1 error to quantify the discrepancy. For normal estimation, we normalize both the estimated and ground-truth values to unit vectors and then compute the L1 error and the angle error for evaluation.

S3.2. Indoor object reconstruction

S3.2.1 Experiments on 3D-FRONT and Pix3D

In this subsection, we provide more details about our reproduced results and more qualitative results. Tabs. S1 and S3 list the quantitative outcomes on 3D-FRONT [19] and Pix3D [69] datasets documented by Liu *et al*. [41] in the original paper. The "NC" (normal consistency) columns in these tables are empty, as Liu *et al*. [41] did not measure normal consistency in their paper. Tabs. S2 and S4 present our reproduced results, which are comparable with the results in Tabs. S1 and S3. We additionally provide more qualitative results, Fig. S3 on 3D-FRONT and Fig. S4 on Pix3D. As can be seen from these figures, our model can learn finer and smoother surfaces with high-fidelity textures.

S3.2.2 Failure cases

In this section, we present and diagnose some representative failure examples. Fig. S1(a) illustrates that occlusion between objects can lead to distortions in both shape and appearance recovery, particularly in the occluded areas. Prior work [34, 43] necessitates unoccluded object images as in-

Table S1. **Object reconstruction on the 3D-FRONT [19] dataset.** Our model achieves the best performance on mean CD and F-Score, as well as the best NC on all object categories, outperforming MGN [54], LIEN [87], and InstPIFu [41].

Category		bed	chair	sofa	table	desk	nightstand	cabinet	bookshelf	mean
CD ↓	MGN	15.48	11.67	8.72	20.90	17.59	17.11	13.13	10.21	14.07
	LIEN	16.81	41.40	9.51	35.65	26.63	16.78	7.44	11.70	28.52
	InstPIFu	18.17	14.06	7.66	23.25	33.33	**11.73**	**6.04**	8.03	14.46
	Ours	**4.96**	**10.52**	**4.53**	**16.12**	**25.86**	17.90	6.79	**3.89**	**10.45**
F-Score ↑	MGN	46.81	57.49	64.61	49.80	46.82	47.91	54.18	54.55	55.64
	LIEN	44.28	31.61	61.40	43.22	37.04	50.76	69.21	55.33	45.63
	InstPIFu	47.85	59.08	67.60	56.43	**48.49**	57.14	**73.32**	66.13	61.32
	Ours	**76.34**	**69.17**	**80.06**	**67.29**	47.12	**58.48**	70.45	**85.93**	**71.36**
NC ↑	MGN	-	-	-	-	-	-	-	-	-
	LIEN	-	-	-	-	-	-	-	-	-
	InstPIFu	-	-	-	-	-	-	-	-	-
	Ours	**0.896**	**0.833**	**0.894**	**0.838**	**0.764**	**0.897**	**0.856**	**0.862**	**0.854**

put to avoid this problem. One of the future directions is to involve the accurate reconstruction of object shapes with textures under heavy occlusions. The bookshelf depicted in Fig. S1(b) exemplifies the challenges our framework encounters when attempting to reconstruct intricate geometry. We hypothesize that this can be attributed to the limited representation power of SDF as the implicit surface representations for thin surfaces. Specifically, it requires the model to recognize and reconstruct abrupt changes in the signed distance field within centimeters, transitioning from positive (outside) to negative (inside) and then back to positive again. To address such issues, we suggest future endeavors in integrating unsigned distance field [22, 44] with volume rendering to capture such intricate geometry and non-watertight meshes.

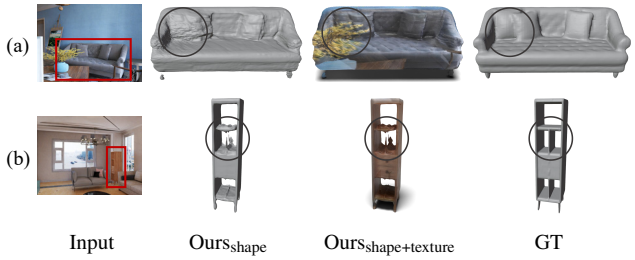

Figure S1. **Qualitative examples for failure cases.** Two representatives with (a) occlusions and (b) intricate geometry.

S3.2.3 Comparison with prior-guided models

We randomly select 100 samples from the test split in 3D-FRONT to perform a comparative assessment of the reconstruction performance between our framework and prior-guided models, *i.e.*, Zero-1-to-3 [43] and Shap·E [34]. In the original paper, Zero-1-to-3 [43] utilized SJC [73] for the 3D reconstruction task, whereas we follow Tang *et al.* [70] to employ DreamFusion [62] to achieve enhanced results. In Shap·E [34], the text prompts are required for reconstruction using the conditional generative model. We utilize the ground-truth object category as the designated text prompt to maintain fairness. It is noteworthy that both Zero-1-to-3 [43] and Shap·E [34] necessitate background-free input images for the target objects. Consequently, we utilize the ground truth mask to segment the object as input. For our model, we use the original image as input.

Table S2. **Reproduced results on the 3D-FRONT [19] dataset.** †: Results reproduced from the official repository.

Category		bed	chair	sofa	table	desk	nightstand	cabinet	bookshelf	mean
CD ↓	MGN†	5.95	14.31	6.01	24.21	38.07	20.27	12.49	11.42	14.97
	LIEN†	4.58	11.85	7.21	30.40	42.52	20.84	14.81	18.13	16.33
	InstPIFu†	7.68	13.79	6.78	21.56	31.32	**13.14**	**5.94**	13.79	13.63
	Ours	**4.96**	**10.52**	**4.53**	**16.12**	**25.86**	17.90	6.79	**3.89**	**10.45**
F-Score ↑	MGN†	65.47	52.83	68.98	54.04	42.9	45.01	52.01	55.26	57.19
	LIEN†	72.16	62.23	68.25	48.18	32.07	42.87	49.62	43.12	58.37
	InstPIFu†	62.28	66.31	69.65	58.73	40.49	57.52	**76.59**	71.48	65.34
	Ours	**76.34**	**69.17**	**80.06**	**67.29**	**47.12**	**58.48**	70.45	**85.93**	**71.36**
NC ↑	MGN†	0.829	0.758	0.819	0.785	0.711	0.833	0.802	0.719	0.787
	LIEN†	0.822	0.793	0.803	0.755	0.701	0.814	0.801	0.747	0.786
	InstPIFu†	0.799	0.782	0.846	0.804	0.708	0.844	0.841	0.790	0.810
	Ours	**0.896**	**0.833**	**0.894**	**0.838**	**0.764**	**0.897**	**0.856**	**0.862**	**0.854**

Table S3. **Object Reconstruction on the Pix3D [69] dataset.** On the non-overlapped split [41], our model outperforms the state-of-the-art methods by significant margins.

Category		bed	bookcase	chair	desk	sofa	table	tool	wardrobe	misc	mean
CD ↓	MGN	22.91	33.61	56.47	33.95	9.27	81.19	94.70	10.43	137.50	44.32
	LIEN	11.18	29.61	40.01	65.36	10.54	146.13	29.63	4.88	144.06	51.31
	InstPIFu	10.90	7.55	32.44	**22.09**	8.13	45.82	10.29	**1.29**	47.31	24.65
	Ours	**6.31**	**7.21**	**26.23**	28.63	**5.68**	**43.87**	**8.29**	2.07	**35.03**	**21.79**
F-Score ↑	MGN	34.69	28.42	35.67	65.36	51.15	17.05	57.16	52.04	10.41	36.20
	LIEN	37.13	15.51	25.70	26.01	49.71	21.16	5.85	59.46	11.04	31.45
	InstPIFu	54.99	62.26	35.30	**47.30**	56.54	37.51	64.24	**94.62**	27.03	45.62
	Ours	**68.78**	**66.69**	**55.18**	42.49	**71.22**	**51.93**	**65.38**	91.84	**46.92**	**59.71**
NC ↑	MGN	-	-	-	-	-	-	-	-	-	-
	LIEN	-	-	-	-	-	-	-	-	-	-
	InstPIFu	-	-	-	-	-	-	-	-	-	-
	Ours	**0.825**	**0.689**	**0.693**	**0.776**	**0.866**	**0.835**	**0.645**	**0.960**	**0.599**	**0.778**

Table S4. **Reproduced results on the Pix3D [69] dataset.** †: Results reproduced from the official repository.

Category		bed	bookcase	chair	desk	sofa	table	tool	wardrobe	misc	mean
CD ↓	MGN†	11.73	17.50	36.74	29.63	7.02	47.85	25.63	6.61	62.32	27.9
	LIEN†	16.31	26.05	33.06	39.84	7.34	76.97	27.84	4.66	126.88	33.66
	InstPIFu†	9.82	7.73	31.55	**23.18**	7.28	45.89	9.64	**1.79**	43.4	24.35
	Ours	**6.31**	**7.21**	**26.23**	28.63	**5.68**	**43.87**	**8.29**	2.07	**35.03**	**21.79**
F-Score ↑	MGN†	51.51	40.85	49.42	44.06	59.48	45.02	45.52	64.76	26.23	50.55
	LIEN†	50.20	36.26	48.41	32.55	64.98	29.97	38.72	76.03	18.29	47.87
	InstPIFu†	57.02	61.45	38.06	**45.98**	62.76	37.26	63.50	**93.94**	40.14	48.09
	Ours	**68.78**	**66.69**	**55.18**	42.49	**71.22**	**51.93**	**65.38**	91.84	**46.92**	**59.71**
NC ↑	MGN†	0.737	0.592	0.525	0.633	0.756	0.794	0.531	0.809	0.563	0.659
	LIEN†	0.706	0.514	0.591	0.581	0.775	0.619	0.506	0.844	0.481	0.646
	InstPIFu†	0.782	0.646	0.547	0.758	0.753	0.796	0.639	0.951	0.580	0.683
	Ours	**0.825**	**0.689**	**0.693**	**0.776**	**0.866**	**0.835**	**0.645**	**0.960**	**0.599**	**0.778**

S3.3. Rendering capability

S3.3.1 Novel view synthesis

To measure the capability for novel view synthesis, we compare our model with PixelNeRF [83] on the category-agnostic model, which is first trained on the ShapeNet [7] and then fine-tuned using the 3D-FRONT [19]. Results in Fig. 7 show that PixelNeRF struggles to render images outside the vicinity of the original viewpoints where our model is capable of generating meaningful renderings from novel views. The main reason behind this is that our method employs 3D supervision, which helps the model learn better 3D object priors. PixelNeRF fails to acquire a meaningful 3D prior from the training data, especially when each image exists independently in 3D-FRONT [19], which is in stark contrast to the ShapeNet [7] where images are presented in a sequence of related perspectives. Fig. S5 shows that our model not only generates high-quality new perspective images but also produces reasonable depth and normal maps by volume rendering.

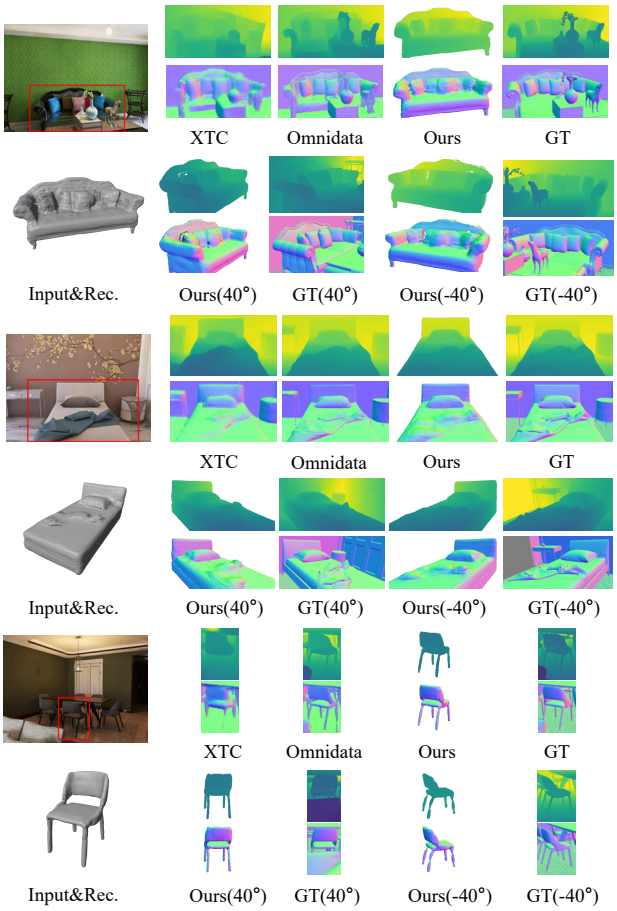

Figure S2. **Qualitative results for depth and normal estimation.** Our method produces results comparable with [17, 86] on the input view, and can estimate depth and normal for novel views.

Table S5. **Novel views depth and normal estimation.** We evaluate depth using L1 ↓ and normal using L1↓ / Angular°↓ error.

Angle	Method	Depth(L1↓)	Normal(L1↓ / Angular°↓)
0°	XTC [86]	1.188	12.712 / 14.309
	Omnidata [17]	0.734	10.015 / 11.257
	Ours	0.992	10.962 / 12.392
5°	Ours	1.0313	11.2132 / 12.631
10°		1.0911	11.5866 / 13.1078
15°		1.1796	12.0943 / 13.7095
20°		1.2624	12.6315 / 14.2108
30°		1.3934	13.4501 / 15.1012
40°		1.4694	15.0034 / 16.6108

S3.3.2 Depth and normal estimation

Our framework can serve as a proficient single-view depth and normal estimator, and to compare with zero-shot state-of-the-art methods [17, 86], we randomly select 200 samples from the test split in 3D-FRONT. To assess our model's capability to generate depth and normal maps from novel views, we rotated the camera vertically left and right by 5°, 10°, 15°, 20°, 30°, and 40°. It is worth noting that only our method possesses the capability to estimate depth and normal from novel views. Tab. S5 and Fig. S2 show that our model can consistently produce satisfactory outcomes, even when the viewing angle changes significantly.

Figure S3. **More qualitative results from 3D-FRONT [41].**

Figure S4. **More qualitative results from Pix3D [69].**

Figure S5. **More qualitative results for novel views synthesis.**