Supplementary Material: FIRe: Fast Inverse Rendering using Directional and Signed Distance Functions

Tarun Yenamandra[1] Ayush Tewari[2] Nan Yang[1] Florian Bernard[3]
Christian Theobalt[4] Daniel Cremers[1]
[1]Technical University of Munich [2]Massachusetts Institute of Technology [3]University of Bonn [4]Max Planck Institute for Informatics, SIC

In this supplementary material, we provide the implementation details, such as our MLP architecture, and hyperparameters. We show additional qualitative and quantitative results. Finally, we evaluate our model for 3D shape reconstruction from single-view silhouettes and compare the results with DIST's [7].

A. Implementation Details

In this section, we provide details about our training, inference, network architecture, and hyperparameters.

Training: We train a model for each of the 6 classes of the ShapeNet dataset with the splits from DeepSDF [9]. We use a batch size of 64 and 4096 samples per scene. We train for 3000 iterations for classes with less than 3000 shapes, and 2000 iterations otherwise. Each batch takes about 2.5s on an Nvidia A100 GPU. Depending on the number of shapes in a class it takes $1-4$ days to train a model per class. We do not train the SDF model with the Track-SDF regularizer in Eq. 10 (main). We truncate the SDF values for better shape representations [9].

Inference: We run all the inference tasks on an Nvidia A40 GPU both for ours and DIST [7] to ensure that the optimization and rendering performance we report are consistent.

A.1. Hyperparameters

We optimize the losses during both training and inference with ADAM [5] algorithm in Pytorch [10]. During **training**, we set the weights $w_s = 1.0$, $w_d = 1.0$, $w_\sigma = 1.0$, $w_{tv} = 100.0$, $w_{ts} = 0.1$, and $w_l = 0.0001$ respectively for SDF loss, DDF loss, Ray hit loss, TV regularizer, Track-SDF regularizer, and latent code regularizer. We train our models with a learning rate of 0.0005 and our latent codes with a learning rate of 0.001. After every 750 or 500 iterations (for 3000 or 2000 total iterations, respectively), we divide the learning rate by 2. During **inference**, for 3D reconstruction from single-view depth maps, we set $w_S = 1.0$, $w_D = 1.0$, and $w_l = 0.0001$ for the silhouette loss, the depth loss, and the latent regularizer respectively. For 3D reconstruction from single-view silhouettes, we set $w_S = 1.0$, $w_D = 0.0$, and $w_l = 0.005$. We run our algorithm for 1000 iterations with a starting step size of 0.001 and a step size of 0.0005 after 500 iterations. We set a threshold of 0.8 for DDF ray hit prediction for rendering. We extract meshes at the level-set of SDF, $s = 0.001$. Similar to DeepSDF [9], we set a truncation distance $\tau = 0.1$. In other words, SDF values that are more than 0.1 are set to 0.1 and those less than -0.1 are set to -0.1. We use the released code and parameters of DIST [7] and IF-NET [1] for comparisons.

A.2. Network Architecture

DDF Model (f_d):
2D Feature Grids: For each of the 15 feature grids, we set a resolution of 512×512 and a feature dimension of 32. **MLP:** The MLP in the DDF model consists of 3 blocks of fully-connected layers. Each fully-connected block has 1 hidden layer of 512 dimension. There are skip connections to each block from the input. We use ReLU activations for the outputs of all the layers except the last layer where we use no activation function. We positionally encode [8] each dimension of the input points and direction tuple with 3 frequencies.
SDF Model (f_s):
2D Feature Grids: For each of the 3 feature grids, we set a resolution of 512×512 and a feature dimension of 32. **MLP:** The MLP in the SDF model consists of a fully-connected block with 2 hidden layers of 256 dimension. We use ReLU activations for the outputs of all the layers except the last layer where we use no activation function. We positionally encode [8] each dimension of the input points with 3 frequencies.

A.3. Quantitative Evaluation Metrics

We extract meshes at a resolution of 256^3 using marching cubes to evaluate our SDF reconstructions quantitatively. To evaluate our DDF reconstructions, we first randomly sample points on the unit sphere and directions. We

use rejection sampling by means of the ray hit predictions from our DDF model to obtain point-direction pairs that hit the object's surface. We obtain a point for each point-direction pair using $o+dr$ where (o, r) is the point-direction pair that points to the surface and d is the predicted directional distance. We use the symmetric L2 chamfers distance as the metric for our quantitative experiments, i.e.

$$\text{CD} = \frac{1}{N}\Sigma_{i=1}^{N} \min_{y \in Y} \|x_i - y\|_2^2 + \frac{1}{N}\Sigma_{i=1}^{N} \min_{x \in X} \|x - y_i\|_2^2 , \tag{1}$$

where $X = \{x_i \in \mathbb{R}^3 | \, i = 1, \ldots, N\}$ and $Y = \{y_i \in \mathbb{R}^3 | \, i = 1, \ldots, N\}$ are points on the surfaces of two shapes, and N is the number of points sampled on the two shapes. We compute chamfer's distance between 30000 points from the two sets in the ShapeNet dataset's scale similar to DeepSDF [9].

B. Experiments

In this section, we show additional qualitative results on 3D shape reconstruction from depth maps in Fig. 5. We show qualitative results on our DDF model evaluations of the state-of-the-art directional distance models in Fig. 3. We show results on reconstructing shapes from synthetic and real-world videos using our trained model in Sec. B.1, followed by shape reconstruction from silhouette experiments in Sec. B.2. Finally, we provide details about our visualizations in Sec. C.

	Ours			DIST	
Class	**SDF**	**DDF**	**Time (s)**	**DeepSDF**	**Time (s)**
Cars	0.62	0.48	2.59	0.69	29.54
Planes	1.60	1.58	2.59	1.44	30.18
Sofas	1.80	1.87	2.50	1.82	29.89

Table 1. Quantitative results ($1000\times$ CD) of 3D reconstruction from video sequences in PMO [6]. Our method is on average $12\times$ faster than DIST [7] while being as accurate.

B.1. Reconstruction from Videos

While our representation performs well with the algorithm that we propose, we evaluate its capacity to accelerate existing algorithms. Towards that, we replace the sphere tracer in DIST's reconstruction from videos of PMO [6] dataset. We are given camera poses for the frames without object masks. We optimize for the latent code as

$$\underset{z}{\text{argmin}} \sum_{i=0}^{N-1} \sum_{j \in \mathcal{N}_i} \|I_i - I_{j \to i}(f(z))\| + \mathcal{L}_{s_+}(\sigma) + \mathcal{L}_{s_-}(\sigma) ,$$

where $N = 72$ is the number of frames of the video, I_i is the i^{th} image and $I_{j \to i}$ is an image formed using pixel colors warped from image j to i using the rendered depth, $f(z)$, obtained from DDF model as $(f(z), \sigma) = f_d((p, r), z, f_{pd}; \Theta_d)$ (see Eq. (5), main), $\sigma \in \{0, 1\}$ is the predicted mask, and $\mathcal{N}_i$ is the neighborhood of frame i. We enforce the constraints $\mathcal{L}_{S_{s_+}}(\sigma)$ and $\mathcal{L}_{S_{s_-}}(\sigma)$, from Eqs. (16) and (17) (main) using predicted masks (σ) to ensure that the predicted SDF surface follows DDF. We show the qualitative results in Fig. 1. We show quantitative results in Tab. 1. As corroborated in the ablations (wo $\mathcal{L}_S$), our algorithm performs similarly compared to DIST when ground truth masks are unavailable. However, as we do not need to perform the expensive sphere tracing in each iteration, our method is $12\times$ faster. This shows the potential of our trained DDF model to replace sphere tracing in existing algorithms for an order of magnitude acceleration.

Reconstruction from real-world videos: We show qualitative results on reconstruction from 40 real-world videos of chairs from the Redwood dataset [2] in Fig. 2. We optimize for the photometric consistency across frames using the depth predicted by our model. As the dataset consists of reconstructed meshes of entire scenes, we can evaluate the distance from the predicted shape using our model to the ground mesh. The mean distance is 4.0cm.

B.2. Shape from Silhouettes

We evaluate our method on the challenging task of reconstructing 3D shapes from single-view silhouettes with a given camera pose. The task is much more challenging compared with reconstructing from single-view depth maps as we have no information about the shape of an object apart from a silhouette in an image.

We optimize for shape with the 3D reconstruction from the single-view depth maps algorithm, presented in Sec. 3.4 of the main paper, without the depth loss, i.e., $w_D = 0.0$. We compare with DIST [7] for this task while setting the weight of the depth loss in DIST's reconstruction algorithm to 0.0. We test on the same test split as in 3D reconstruction from depth maps, i.e., the first 200 shapes from test splits of each category and the first image from the test dataset of 3D-R2N2 [3].

Results: We show qualitative results of reconstructing 3D shape from a silhouette in Fig. 4. We show the quantitative results in Tab. 2. We report $1000\times$ the L2 chamfer distance (see Sec. 4.3, main paper) between the reconstructed and ground truth meshes. Our method outperforms DIST [7] in all classes. As can be seen, our algorithm reconstructs more plausible shapes as our DDF model's ray hit output can be used to fit our models to the given silhouettes, as corroborated by the reconstruction accuracy of DDF. Further, as our DDF model is trained using the Track-SDF loss and as our models share a latent space, our SDF reconstructions are more accurate compared with DIST's reconstructions.

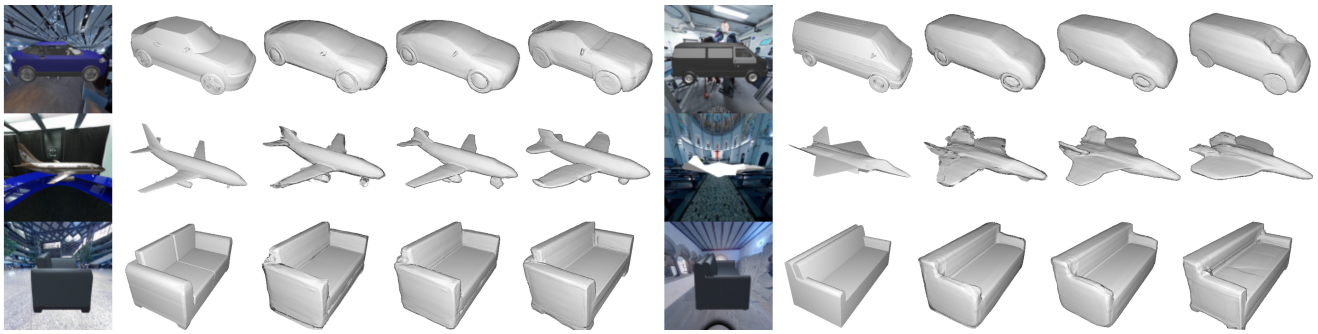

Figure 1. Qualitative results of 3D reconstruction from videos of PMO [6]. In both columns, Left to right: a frame from the input video, reference ground truth geometry, 1 forward pass renders from our DDF, our SDF reconstructions, and reconstructions with vanilla DIST's algorithm. When we replace the raymarching algorithm in the vanilla implementation (DIST [7]) with our DDF model, the algorithm is accelerated by about $12\times$ as our model only needs 1 forward pass through the network per ray to render.

Figure 2. Qualitative results of 3D reconstruction from 40 real-world videos of chairs from Redwood dataset [2]. We reconstruct chairs by optimizing for photometric consistency between different frames of videos using the depth predicted by our model. Left: A frame from the input video. Right: DDF renders and reconstructed SDF in two views. The average distance from the predicted shape to the ground truth mesh is 4.0cm.

	Ours	**Ours DDF**	**DIST**	**Ours**	**Ours DDF**	**DIST**
	$1000\times$ CD $\downarrow$ (Mean)			$1000\times$ CD $\downarrow$ (Median)		
Car	0.99	**0.75**	1.53	0.90	**0.51**	1.16
Chair	2.70	**2.06**	7.84	1.93	**1.51**	4.58
Lamp	**6.91**	8.02	11.92	**3.81**	4.33	5.19
Plane	**0.74**	0.78	5.74	0.45	**0.40**	4.22
Sofa	**2.08**	2.33	3.82	1.69	**1.62**	2.18
Table	3.25	**1.97**	5.95	2.37	**1.28**	3.35

Table 2. Quantitative results on 3D shape reconstruction given a silhouette. Our method outperforms DIST [7] in all the classes by a large margin as our DDF model predicts silhouette along with directional distance. Further, as the DDF model is trained to track the surface predicted by SDF model, both our DDF and SDF models can reconstruct from silhouettes better than the state-of-the-art.

C. Visualizations

For the results named "DDF Renders" or "DDF Reconstructions", we visualize the results of our method with one evaluation of DDF per ray. We obtain the silhouette predicted by the DDF model and use the SDF model to compute the surface normals for shading. The rendering time we reported in the main paper includes the time to compute normals using finite differences. We use sphere tracing to show our 3D reconstructions, and the results of DIST [7]. We show reconstructed meshes for IF-NET [1]. We render meshes of IF-NETs for visualizations, and 3D-R2N2's test data set for obtaining depth maps and masks with the help of Trimesh [4]. For shading, we use an accelerated version of Blinn-Phong shader from i3DMM's code [11].

References

[1] Julian Chibane, Thiemo Alldieck, and Gerard Pons-Moll. Implicit functions in feature space for 3d shape reconstruction and completion. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, jun 2020. 1, 3, 7

[2] Sungjoon Choi, Qian-Yi Zhou, and Vladlen Koltun. Robust reconstruction of indoor scenes. In *IEEE Conference*

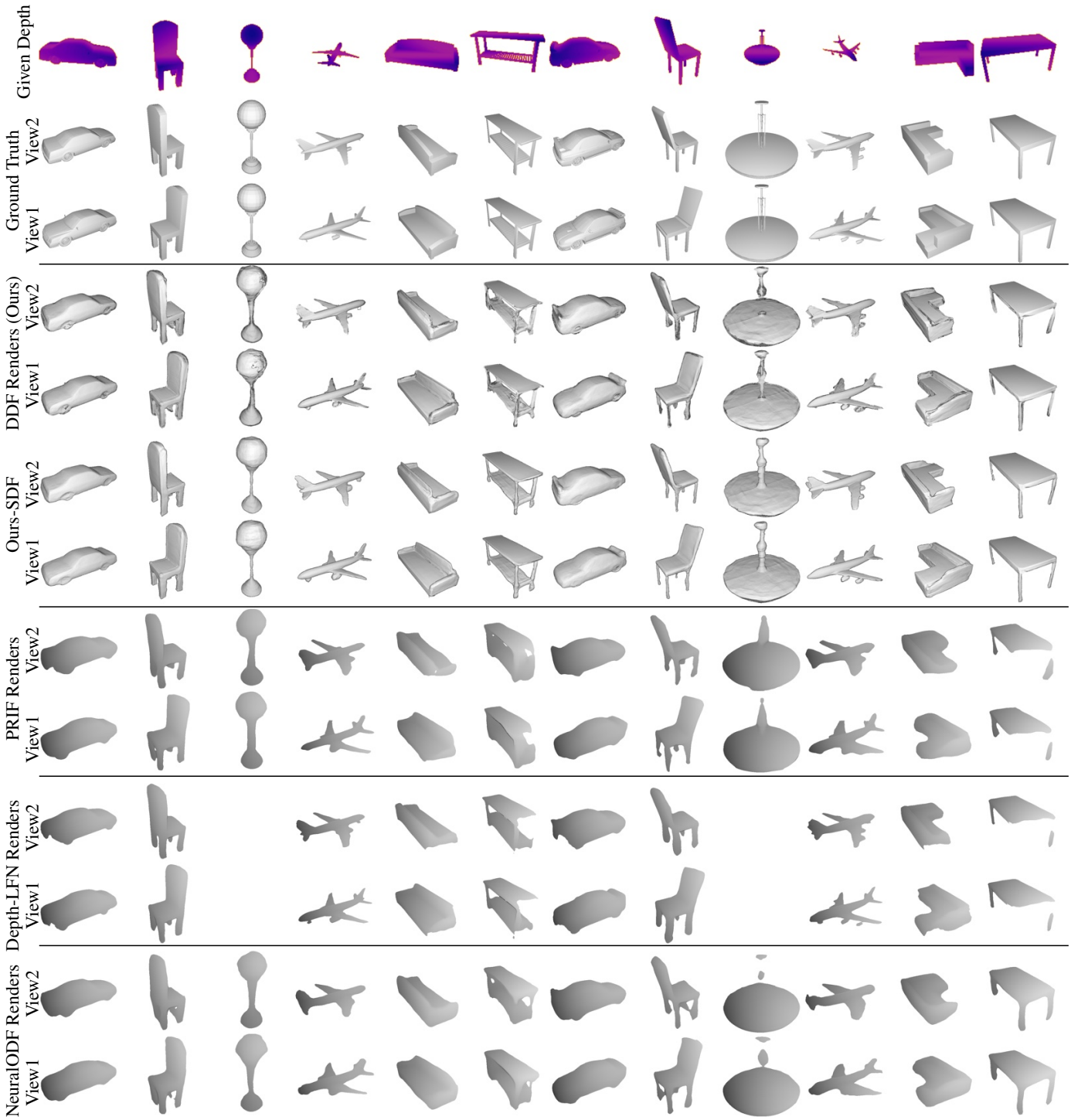

Figure 3. Qualitative results of model evaluation against PRIF, Depth-LFN, and NeuralODF on shape reconstruction from a given depth map. Each column shows reconstruction results for different shapes. Top rows: given depth map (top), ground truth geometry for reference (bottom). Middle-top rows: views rendered with 1 forward pass from our DDF model (top) and views of 3D shapes reconstructed from our SDF model (bottom). Middle rows: views rendered with 1 forward pass from PRIF model. Middle-bottom rows: views rendered with 1 forward pass from Depth-LFN model. Bottom rows: views rendered with 1 forward pass from NeuralODF model. We shade competitive models by estimating normals from rendered depth. As can be seen, our model results in more view-consistent shapes with better geometric details, given that we couple our DDF model with the SDF model. (Depth-LFN did not converge for lamps class.)

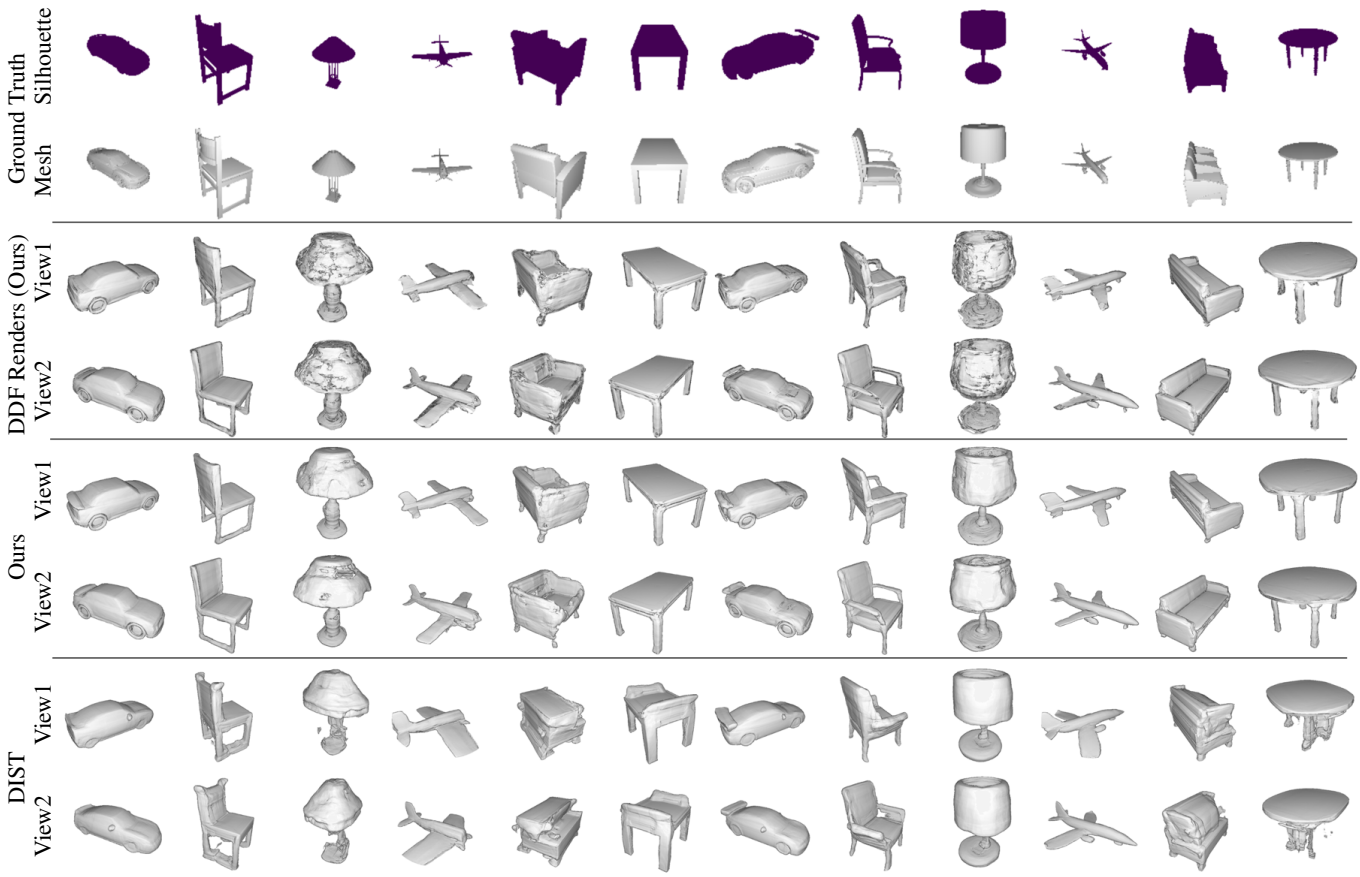

Figure 4. Qualitative results of 3D shapes reconstructed from a given silhouette. Each column shows reconstruction results for different shapes. Top rows: given silhouettes and meshes (for reference). Upper-middle rows: reconstructed views rendered with 1 forward pass of our DDF model per camera ray. Middle rows: views of 3D shapes reconstructed by our SDF model. Last rows: views of 3D shape reconstructed using DIST [7].

on Computer Vision and Pattern Recognition (CVPR), 2015. 2, 3

[3] Christopher B Choy, Danfei Xu, JunYoung Gwak, Kevin Chen, and Silvio Savarese. 3d-r2n2: A unified approach for single and multi-view 3d object reconstruction. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2016. 2

[4] Dawson-Haggerty et al. trimesh. 3

[5] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization, 2014. 1

[6] Chen-Hsuan Lin, Oliver Wang, Bryan C Russell, Eli Shechtman, Vladimir G Kim, Matthew Fisher, and Simon Lucey. Photometric mesh optimization for video-aligned 3d object reconstruction. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 2, 3

[7] Shaohui Liu, Yinda Zhang, Songyou Peng, Boxin Shi, Marc Pollefeys, and Zhaopeng Cui. Dist: Rendering deep implicit signed distance function with differentiable sphere tracing. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. 1, 2, 3, 5, 7

[8] 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 *ECCV*, 2020. 1

[9] Jeong Joon Park, Peter Florence, Julian Straub, Richard Newcombe, and Steven Lovegrove. Deepsdf: Learning continuous signed distance functions for shape representation. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019. 1, 2

[10] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Köpf, Edward Yang, Zach DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library, 2019. 1

[11] Tarun Yenamandra, Ayush Tewari, Florian Bernard, Hans-Peter Seidel, Mohamed Elgharib, Daniel Cremers, and Christian Theobalt. i3dmm: Deep implicit 3d morphable

model of human heads. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12803–12813, 2021. 3

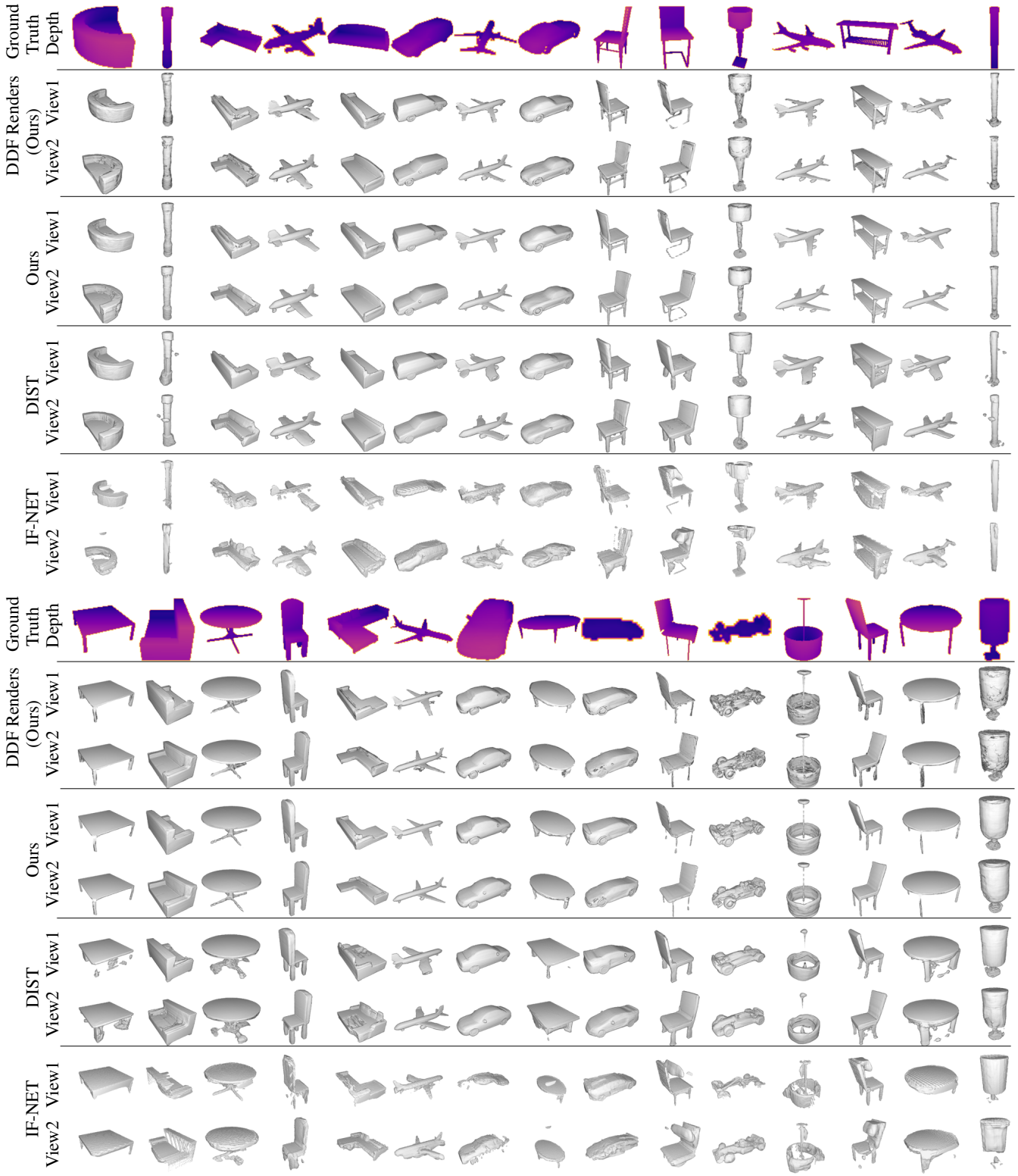

Figure 5. Additional qualitative results of 3D reconstruction from single-view depth maps discussed in Sec. 4.2 of the main paper. Each column shows reconstruction results for different shapes. Top row: given depth map. Upper-middle rows: views rendered with 1 forward pass from our DDF model. Middle rows: views of 3D shapes reconstructed by our SDF model. Lower-middle rows: views of 3D shape reconstructed using DIST [7]. Last rows: views of 3D shapes reconstructed using IF-NET [1].