Supplementary Materials for Planes vs. Chairs: Category-guided 3D shape learning without any 3D cues

Zixuan Huang[1], Stefan Stojanov[1], Anh Thai[1],
Varun Jampani[2], and James M. Rehg[1]

[1] Georgia Institute of Technology
[2] Google Research

1 Overview

This supplementary document is structured as follows: In Section 2 we provide a formal description of our viewpoint regularization; In Section 3 we provide more experimental results on Pix3D [12]; In Section 4 we empirically quantify the value of using category labels as opposed to multi-view supervision; In Section 5 we present additional SOTA comparison on ShapeNet-13; In Section 6 we analyze the benefits of multi-category learning over category-specific learning; In Section 7 we give implementation and training details for our model; In Section 8 we discuss the limitations of our approach and in Section 9 we present additional qualitative results on our large-scale ShapeNet-55 renderings.

2 Viewpoint regularization via cycle-consistency

In this section, we provide a more detailed description of our viewpoint regularization. To regularize the viewpoint predictor, we require the viewpoint predictor to accurately predict the viewpoint of randomly rendered images. Different from real images, we can render an arbitrary number of images by sampling viewpoints with a given shape and texture. This can be thought of as creating pseudo data-label pairs for the viewpoint predictor.

Formally, given shape $\boldsymbol{f_S}$, texture field $\boldsymbol{f_T}$ and a random viewpoint $\boldsymbol{v_r}$, we render an image $\hat{I} = \boldsymbol{R}(\boldsymbol{f_S}, \boldsymbol{f_T}, \boldsymbol{v_r})$. The goal is to minimize the distance between $\boldsymbol{v_r}$ and $\hat{\boldsymbol{v_r}} = \boldsymbol{V}(\hat{I})$. With the trigonometric function representation, we maximize the cosine similarity between $\boldsymbol{v_r}$ and $\hat{\boldsymbol{v_r}}$ by minimizing

$$\mathcal{L}_{cam} = 1 - \langle \boldsymbol{v_r}, \hat{\boldsymbol{v_r}} \rangle = 1 - \langle \boldsymbol{v_r}, \boldsymbol{V}(\boldsymbol{R}(\boldsymbol{f_S}, \boldsymbol{f_T}, \boldsymbol{v_r})) \rangle, \tag{1}$$

where $\langle \cdot, \cdot \rangle$ denotes dot product. In practice, we apply this regularization on the reconstructed image I_{recon} and the randomly rendered image I_{rnd}, together with the viewpoints that render them. This is computationally efficient as both images are also used in adversarial regularization. We use $\mathcal{L}_{cam}$ only to regularize the viewpoint prediction module, as we stop the gradients from $\mathcal{L}_{cam}$ to shape, texture and rendering modules.

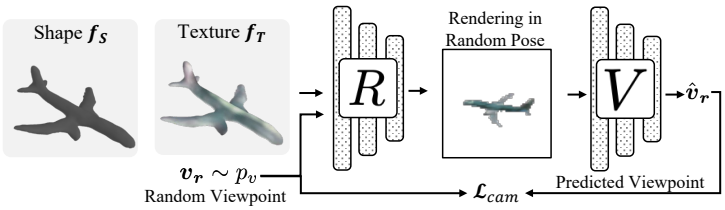

Fig. 1: Viewpoint regularization of our method. With a given shape and texture field, we sample a random viewpoint and render an image as the input to our viewpoint predictor. The viewpoint predictor is then supervised with the randomly sampled viewpoint, which forms a consistency cycle.

3 Additional Experiments on Pix3D

We additionally evaluate our methods on Pix3D [12]. For Pix3D, we use 4 categories including bookcase, chair, table and wardrobe. We split the data with a 70/10/20 percentage into training, validation and testing similar to our experiments on ShapeNet. We compare our method to SDF-SRN [5] and Ye et al. [15] on Pix3D, as shown in Table 1 and Fig. 2. Again, it is clear that our method outperforms the SOTA methods quantitatively and qualitatively. These results further verify the effectiveness of our proposed method.

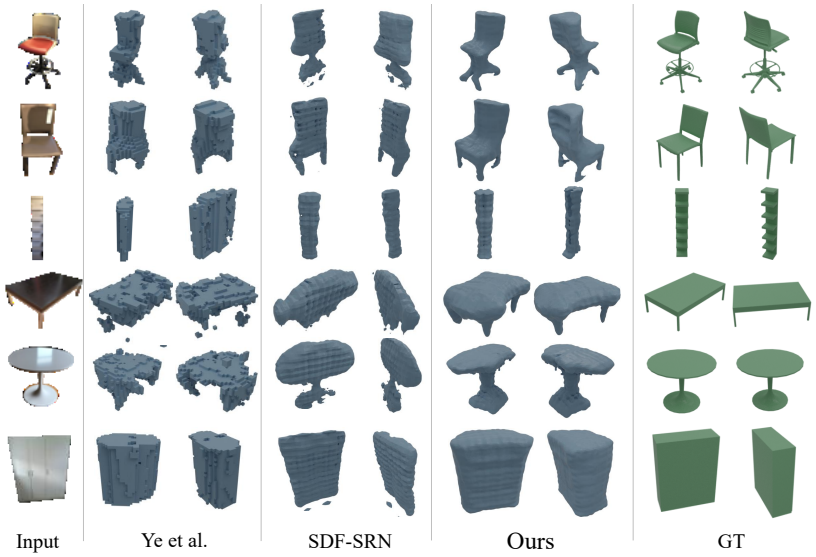

Fig. 2: Qualitative comparison on Pix3D. Our method learns both better global 3D structure and shape details on various categories.

Table 1: Quantitative result measured by CD and F-score on Pix3D. Our method performs favorably to other SOTA methods.

Methods	F-Score@1.0↑	F-Score@5.0↑	F-Score@10.0↑	CD↓
SDF-SRN [5]	0.1370	0.5622	0.7996	0.625
Ye et al. [15]	0.1325	0.5308	0.7994	0.585
Ours	**0.1745**	**0.6604**	**0.8988**	**0.421**

4 Quantifying the value of category labels

We further quantify the value of category label for the Multi-Category Single-View (MCSV) reconstruction task in this section. Our goal is to compare our method with category-guided shape metric learning with a model trained with multi-view supervision (more than a single view available per object instance). In this section, we train all models without adversarial regularization or the viewpoint predictor. We assume the viewpoint is known in this set of experiments. The baseline models are trained using an additional view as supervision. We follow a similar training process as [9] by randomly sampling a view per object for each epoch. To empirically determine the value of category labels, we vary the portion of the data that has multi-view annotations and compare that to our single-view category guided model. The quantitative results are shown in Figure 3. As shown in the figure, having access to category labels can roughly lead to the reconstruction accuracy using 15% to 20% two-view annotation, measured by Chamfer Distance. Given the availability of category label compared to multi-view data, we believe that this is a promising finding.

5 Additional Comparison on ShapeNet-13

In this section, we provide additional comparison to SDF-SRN, Ye et al. [15] and MCMR [11] on ShapeNet-13.

We first compare our method to Ye et al. and MCMR using our original setting, where viewpoints are unknown during training. We train Ye et al. in their category-specific way (13 different models). We train MCMR with category labels as supervision, where we also attach a viewpoint predictor to their model (the predictor use the same trigonometric representation as ours). The results in table 2 show that our method outperforms Ye et al. and MCMR significantly. In our experiments with unknown viewpoints, MCMR fails to disentangle shape from viewpoint, where most shapes only explain input views.

We provide further comparison to SDF-SRN and MCMR under known viewpoints, where we assume the viewpoints are given during training (similar to the setup in original SDF-SRN and MCMR). With this setup, we do not use viewpoint predictor or adversarial regularization in our model because the shape-viewpoint entanglement is not present anymore. As shown in table 3, our method still outperforms SDF-SRN and MCMR significantly.

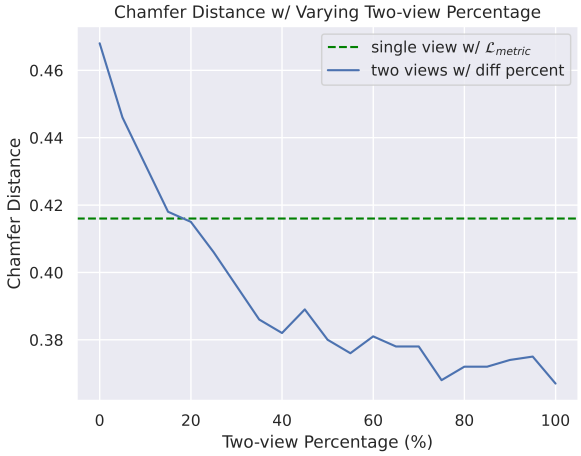

Fig. 3: We evaluate the value of category label in MCSV reconstruction with camera pose. Under this quantitative evaluation, we show the category label will lead to similar performance of having access to around 15% to 20% two-view annotation.

Table 2: Additional comparison measured by CD and F-score on ShapeNet-13.

Methods	F-Score@1.0↑	F-Score@5.0↑	F-Score@10.0↑	CD↓
MCMR [11]	0.0977	0.4054	0.6354	0.941
Ye et al. [15]	0.1349	0.5419	0.7777	0.669
Ours	**0.2005**	**0.7168**	**0.8949**	**0.430**

6 Benefit of multi-category shape learning

In this section we analyze the benefits of multi-category (MC) learning over category-specific (CS) learning. There are 3 reasons we advocate for MC learning: 1) MC learning is the key to avoiding the limitation of linear scaling in the number of models w.r.t. number of categories; 2) Learning category-agnostic features benefits shape generalization to unseen classes; 3) Although learning MC shapes in a single model is more challenging, effective category guidance can enable MC models to outperform CS models on seen classes as well.

We perform additional experiments on ShapeNet-13 to support these benefits. We train both MC and CS models on 3 major categories (car, chair and plane). When tested on the same categories, MC outperforms CS (table 4), even though CS requires extra category labels during inference and three times the memory. We further evaluate generalization performance of both models on 2 unseen categories, bench and vessel. We use oracle for CS models, where we select the best-performing model separately for each unseen category. In table 5,

Table 3: Additional comparison measured by CD and F-score on ShapeNet-13 with known viewpoints.

Methods	F-Score@1.0↑	F-Score@5.0↑	F-Score@10.0↑	CD↓
MCMR w/ V [11]	0.2529	0.7338	0.9025	0.418
SDF-SRN w/ V [5]	0.2695	0.7742	0.9217	0.364
Ours w/ V	**0.3162**	**0.8162**	**0.9410**	**0.324**

Table 4: MC vs. CS models on seen categories.

Methods	F-Score@1.0↑	F-Score@5.0↑	F-Score@10.0↑	CD↓
CS	0.2339	0.7634	0.9366	0.362
MC	**0.2777**	**0.8120**	**0.9399**	**0.331**

the superior performance of MC on unseen categories demonstrates that MC learning can benefit generalization.

Table 5: MC vs. CS models on unseen categories.

Methods	F-Score@1.0↑	F-Score@5.0↑	F-Score@10.0↑	CD↓
CS	0.1536	0.6022	0.8478	0.540
MC	**0.1901**	**0.6676**	**0.8718**	**0.479**

7 Implementation details

In this section, we provide more details about our architecture, loss function, training and evaluation.

7.1 Network architecture

Our architecture of image encoder, shape/texture module as well as the differentiable renderer share a similar design to SDF-SRN [5], while shape and texture modules are made more lightweight for computational efficiency. The image encoder and the viewpoint predictor are based on ResNet18 [2]. The hypernetwork that generates the weights of shape and texture field from latent code is a set of 6-layer MLPs of hidden dimension 512. Each MLP in this set generates the weights of a single layer of either shape or texture. The MLP representing $\boldsymbol{f_S}$ also has a 6-layer structure, with first 4 hidden layer 64 neurons and last hidden layer 32 neurons. The texture MLP has 2 layers, with a hidden dimension of 128. It is conditioned on the shape embedding following [14]. Note that both shape and texture MLPs use Positional Encoding [8] to encode input coordinates for better details. The differentiable renderer we use is from SDF-SRN [5], where a

LSTM [3] learns to perform the ray marching steps. The LSTM predicts a step length for each rendering step based on local implicit feature as input and previous steps encoded as hidden state. We follow IDR [14] (equation 7) to render the extra alpha channel, where we use the negative minimum SDF value on each ray with Sigmoid to represent the alpha value. In practice, we use the minimum SDF value from the ray-marcher steps instead of sampling numerous depths for each ray, and the SDF value is scaled by 30 before Sigmoid to increase the sharpness of the mask.

7.2 Loss function

Our overall loss function is a weighted summation of the reconstruction loss, regularization losses and the losses that facilitate the learning of the renderer and the shape field as in SDF-SRN [5]:

$$\mathcal{L}_{total} = \mathcal{L}_{recon} + \lambda_1 \mathcal{L}_{metric} + \lambda_2 \mathcal{L}_{gan} + \lambda_3 \mathcal{L}_{cam} + \lambda_4 \mathcal{L}_{SDF-SRN} \tag{2}$$

In our experiments, we set $\lambda_2 = 0.2$, $\lambda_3 = 0.03$, $\lambda_4 = 1$ for all datasets. We use a λ_1 of 0.1 for ShapeNet-55 and Pix3D, 0.05 for ShapeNet-13 and 0.03 for Pascal3D+. $\mathcal{L}_{render}$ itself is a weighted summation of several losses as well, we refer to [5] for more details. Since we render an extra alpha channel, we also use the commonly used soft IOU loss [6] to supervise the predicted soft masks with GT masks.

7.3 Training and inference

To train our model, we iterate between the reconstruction step and the adversarial step. In the reconstruction step, the whole model except the discriminator is updated to minimize $\mathcal{L}_{total}$. All the regularizations are activated during this step. In the adversarial step, only the discriminator is updated by maximizing $\lambda_1 \mathcal{L}_{gan}$, with all other losses disabled. For viewpoint sampling, we follow uniform distributions for azimuth, with a range of $[0°, 360°]$ across all datasets. The elevation is also sampled uniformly within $[20°, 40°]$ on ShapeNet-55. We sample elevation and tilt following Gaussian distributions on Pascal3D+ and treat mean and standard deviation as hyperparameters, similar to [15]. Please see Fig. 4 for the comparison between our prior distribution and the GT camera distribution.

We use a learning rate of 0.0001 for both steps, and the optimizer is Adam [4] with $\beta_1 = 0$, $\beta_2 = 0.9$ and a batch size of 12. We did not use weight decay, learning rate scheduling or data augmentations. Our model is trained on a single NVIDIA GTX TITAN V for 40 to 80 epochs, depending on the dataset size. The shape field is pretrained with the SDF values of a sphere for a better initialization as in SDF-SRN [5]. During inference, we only keep the image encoder and the shape prediction module. We implement our method in PyTorch [10].

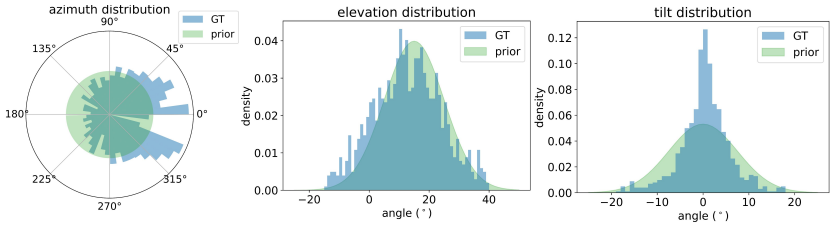

Fig. 4: Comparison between our prior camera distribution (green) for training and the GT (blue) distributions on Pascal3D+.

7.4 Evaluation details

We use the Marching Cubes algorithm [7] to convert the implicit representation to meshes prior to computing the metrics. Specifically, we sample SDF values with a 128^3 spatial grid and extract the 0-isosurface for marching cubes. We further sample 100000 points from each mesh for calculating the metrics. To align the predicted and GT shapes under the same canonical space, we transform both shapes to view-centered frames.

Specifically, when training our models on Pascal3D+ and Pix3D, we assume weak-perspective cameras, and perform center crop and scaling over the input images. Since we do not know where the center of each GT shape is located w.r.t. the cropped and scaled image under a weak perspective camera, we align the meshes by registering shape predictions to the ground truth using the Iterative Closest Point (ICP) algorithm. This is in line with prior works such as SDF-SRN [5].

7.5 License

We develop our code based on the code of SDF-SRN [3] under the MIT license. We use ShapeNetV2 [1], of which the license is specified at https://shapenet.org/terms. We use Pascal3D+ [13] under a MIT license and Pix3D [12] under a Creative Commons Attribution 4.0 International License.

8 Limitations

We discuss more limitations of our method in this section with qualitative examples. When our model fails to reconstruct accurate shapes for some samples we observe it is primarily due to 3 reasons: 1) concavity, 2) class imbalance and 3) complex topology.

Concavity. As discussed in the experiments, our method cannot model highly concave shapes such as bowls or hats. The masks for such shapes do not provide any information that reveals concavity. On the other hand, our method does not

[3] https://github.com/chenhsuanlin/signed-distance-SRN

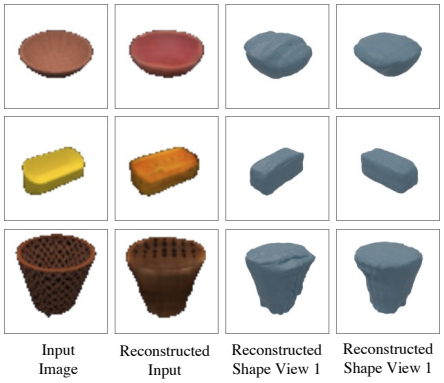

Fig. 5: Illustration of limitation on concavity modeling. Our model fails to reconstruct concave regions for these samples.

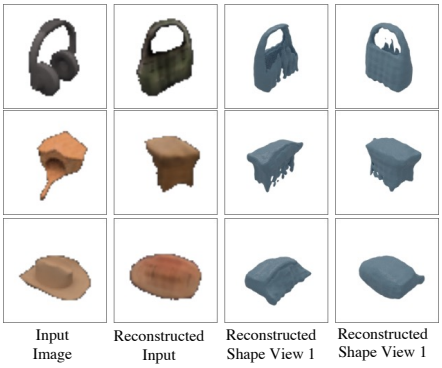

Fig. 6: Illustration of limitation with class imbalance. Our model fails to reconstruct the shape of these rare categories accurately.

have access to explicit lighting or shading information. These factors make the learning of concavity hard. We demonstrate this issue in Fig. 5 with examples from ShapeNet-55, including bowls, bath tubs and trash bins. We think it will be interesting to explore the explicit modeling of lighting/shading for future works.

Class imbalance. We see a strong class imbalance on ShapeNet-55, where several classes have 500 samples while some only have 40 samples. Such an imbalance makes the learning challenging for some categories, as the gradient update within a minibatch can be dominated by major categories. We illustrate this in Fig. 6 by showing the reconstruction on 3 rare categories. We think it will be interesting to systematically explore the imbalance issue for 3D reconstruction.

Complex topologies. Due to the lack of 3D or multi-view supervision, it is still quite challenging to learn accurate shapes when the topologies are complex. We illustrate this issue in Fig. 7 by showing three shelves from ShapeNet-55. We believe it is still an open problem to learn accurate shapes for such examples under the challenging *multi-category, single-view (MCSV)* setting without viewpoint supervision.

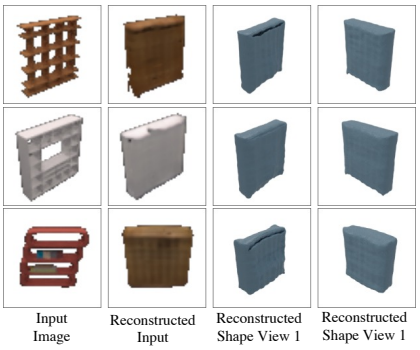

Fig. 7: Illustration of complex topologies. For samples that have complex topologies, our model can only reconstruct a rough global structure.

We hope these limitations are beneficial observations to inform and guide future research under similar challenging settings. On the other hand, despite these limitations, our method can reconstruct accurate shapes

for the majority of images or categories. We believe this is a significant step toward fully unsupervised shape learning.

9 Additional Qualitative Results.

In this section, we show more qualitative results of our model on ShapeNet-55 across various categories, as in Fig. 8, Fig. 9, Fig. 10 and Fig. 11.

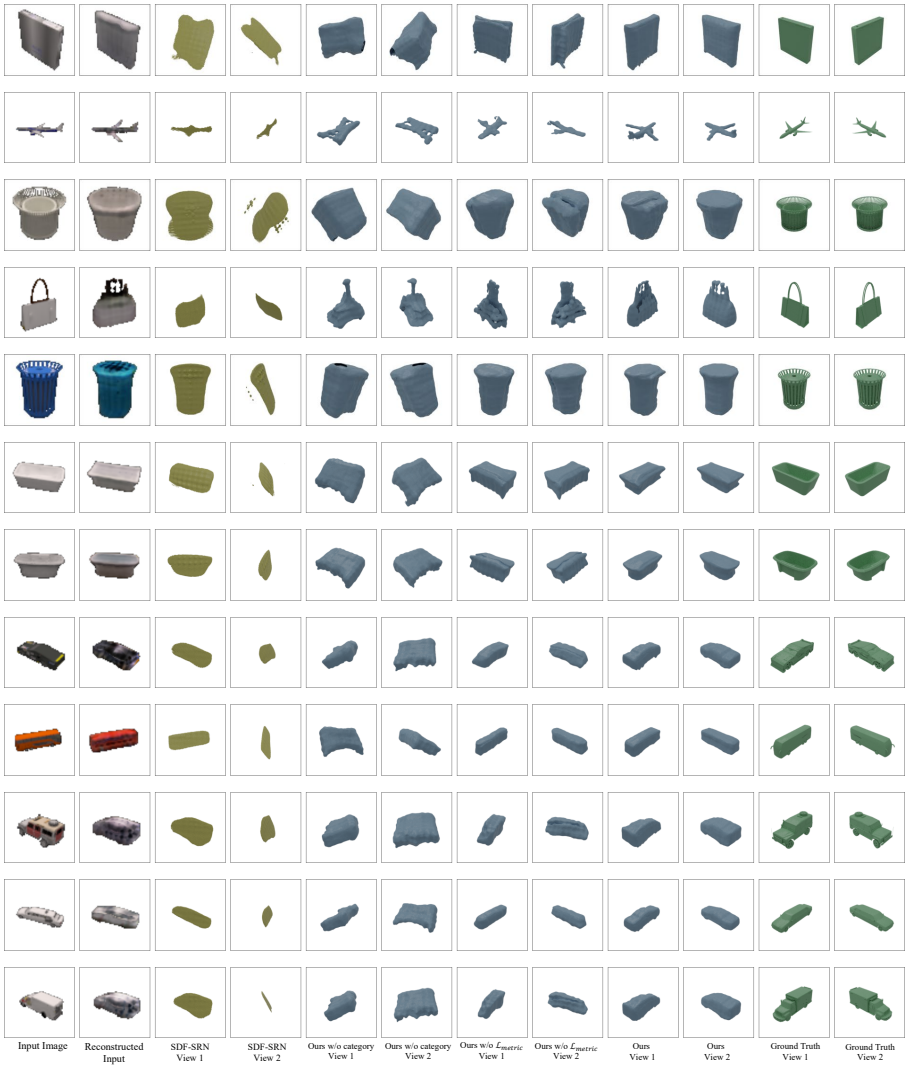

Fig. 8: Additional qualitative results on ShapeNet-55.

Fig. 9: Additional qualitative results on ShapeNet-55.

Fig. 10: Additional qualitative results on ShapeNet-55.

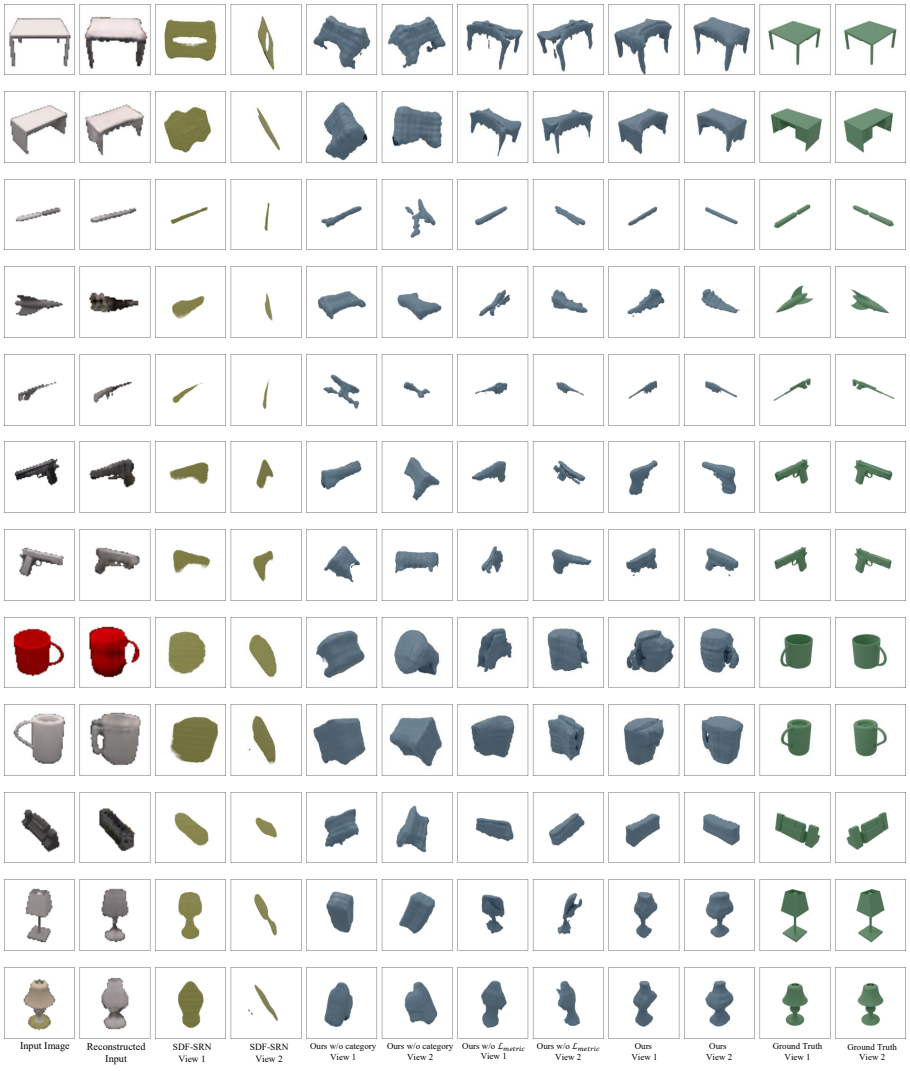

Fig. 11: Additional qualitative results on ShapeNet-55.

References

1. Chang, A.X., Funkhouser, T., Guibas, L., Hanrahan, P., Huang, Q., Li, Z., Savarese, S., Savva, M., Song, S., Su, H., et al.: Shapenet: An information-rich 3d model repository. arXiv preprint arXiv:1512.03012 (2015)
2. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 770–778 (2016)
3. Hochreiter, S., Schmidhuber, J.: Long short-term memory. Neural computation **9**(8), 1735–1780 (1997)
4. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980 (2014)
5. Lin, C.H., Wang, C., Lucey, S.: Sdf-srn: Learning signed distance 3d object reconstruction from static images. arXiv preprint arXiv:2010.10505 (2020)
6. Liu, S., Li, T., Chen, W., Li, H.: Soft rasterizer: A differentiable renderer for image-based 3d reasoning. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 7708–7717 (2019)
7. Lorensen, W.E., Cline, H.E.: Marching cubes: A high resolution 3d surface construction algorithm. ACM siggraph computer graphics **21**(4), 163–169 (1987)
8. Mildenhall, B., Srinivasan, P.P., Tancik, M., Barron, J.T., Ramamoorthi, R., Ng, R.: Nerf: Representing scenes as neural radiance fields for view synthesis. In: European Conference on Computer Vision. pp. 405–421. Springer (2020)
9. Niemeyer, M., Mescheder, L., Oechsle, M., Geiger, A.: Differentiable volumetric rendering: Learning implicit 3d representations without 3d supervision. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 3504–3515 (2020)
10. Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., et al.: Pytorch: An imperative style, high-performance deep learning library. Advances in neural information processing systems **32**, 8026–8037 (2019)
11. Simoni, A., Pini, S., Vezzani, R., Cucchiara, R.: Multi-category mesh reconstruction from image collections. In: 2021 International Conference on 3D Vision (3DV). pp. 1321–1330. IEEE (2021)
12. Sun, X., Wu, J., Zhang, X., Zhang, Z., Zhang, C., Xue, T., Tenenbaum, J.B., Freeman, W.T.: Pix3d: Dataset and methods for single-image 3d shape modeling. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2018)
13. Xiang, Y., Mottaghi, R., Savarese, S.: Beyond pascal: A benchmark for 3d object detection in the wild. In: IEEE winter conference on applications of computer vision. pp. 75–82. IEEE (2014)
14. Yariv, L., Kasten, Y., Moran, D., Galun, M., Atzmon, M., Basri, R., Lipman, Y.: Multiview neural surface reconstruction by disentangling geometry and appearance. arXiv preprint arXiv:2003.09852 (2020)
15. Ye, Y., Tulsiani, S., Gupta, A.: Shelf-supervised mesh prediction in the wild. arXiv preprint arXiv:2102.06195 (2021)