3PSDF: Three-Pole Signed Distance Function for Learning Surfaces with Arbitrary Topologies – *Supplemental Materials* –

Weikai Chen Cheng Lin Weiyang Li Bo Yang
Digital Content Technology Center, Tencent Games
{weikaichen,arnolin,kimonoli,brandonyang}@tencent.com

In this supplemental material, we discuss an alternative framework of learning 3PSDF (Section 1), use 3PSDF to model functions or manifolds (Section 2), provide additional implementation details (Section 3), network structure for each experiment (Section 4), comparison between our proposed 3PSDF and TSDF (Section 5), and more results (Section 6).

1. Alternative Learning Framework

In addition to 3-way classification, 3PSDF can be learned using an alternative framework that combines binary classification and regression. Specifically, the binary classification branch learns to classify the space into nan and non-nan regions, where the non-nan region forms a valid narrow band for extracting surface as demonstrated in Figure 2(b) as shown in the main paper. The regression branch strives to regress a continuous SDF in the narrow-band region as generated by the classification branch. Formally, we formulate this alternative framework as follows:

$$\Phi_C(\mathbf{p}, \mathbf{x}) : \mathbb{R}^3 \times \mathcal{X} \mapsto [0, 1], \tag{1}$$

$$\Psi_R(\mathbf{p}, \mathbf{x}) = SDF(\mathbf{p}). \tag{2}$$

In particular, the classification branch Φ_C consumes a 3D query point $\mathbf{p}$ and its corresponding observation $\mathbf{x}$ and predicts the probability of the query point locating in the non-nan region; the regression branch Φ_R directly infers the signed distance of $\mathbf{p}$ as defined in Equation (3) in the main paper.

Surface extraction. The framework based on binary classification and regression requires training of two branches, which can be implemented either using two heads of a backbone network or two independent networks. Once the networks are trained, the sampling points that are classified as nan points by the classification branch are assigned with nan value. The rest points are assigned with continuous SDF distance using the predictions of the regression branch. The resulting 3PSDF field can be directly converted into mesh using the Marching Cubes (MC) algorithm with the isovalue set to 0. Same as 3-way classification, after MC computation, we only need to remove all the nan vertices and faces generated by the null cubes. The remaining vertices and faces serve as the meshing result.

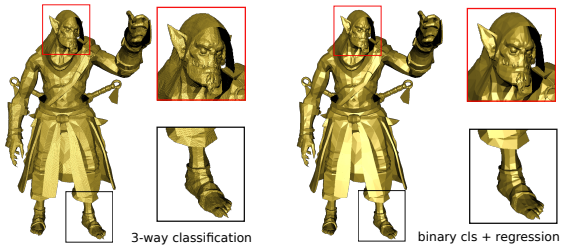

Figure 1. Comparisons of two ways of learning 3PSDF. Quantitative comparisons of shape reconstruction, Mixamo: 0.32:0.31(CD); 0.944:0.950(F-score); MGN: 0.07:0.07(CD); 0.991:0.993(F-score). Note all numbers are reported in format of (3-way cls. : bin. cls.+reg.).

1.1. Comparisons with 3-way Classification

We provide in-depth comparisons between the two candidate learning frameworks: binary classification + regression (BR) v.s. 3-way classification (3C) in this section. Specifically, we evaluate both methods in the task of shape reconstruction and point cloud completion.

Shape reconstruction. We use the same experiment settings with that of the main paper for evaluating the two candidate frameworks. Both methods are validated using two datasets that contain non-watertight open surfaces: MGN [3] and Mixamo [1].

We show both the qualitative and quantitative comparisons in Figure 1. While the two methods are trained using the same data, the BR framework can generate smoother reconstruction compared to that of 3C method, thanks to its

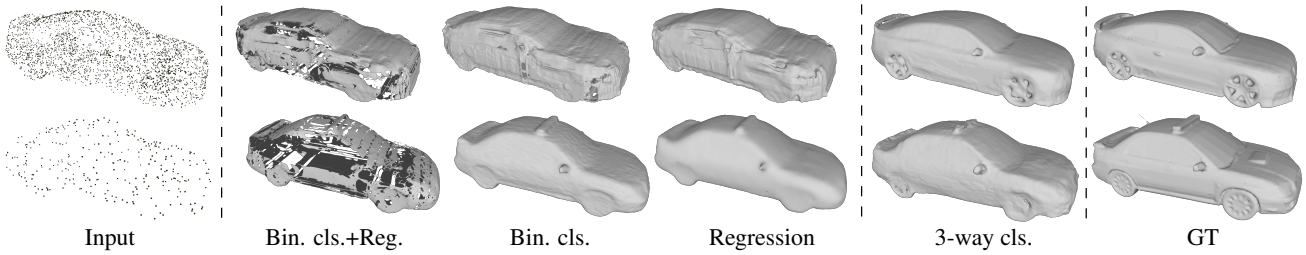

Figure 2. Comparisons of point cloud completion trained on watertight shapes by using two candidate learning frameworks of 3PSDF: binary classification (bin. cls.)+regression (reg.) and 3-way classification (cls.). For the results of BR, we also show the results generated from the two branches.

continuous SDF output. This is also reflected in the quantitative measurements, where BR can achieve comparable or even better results.

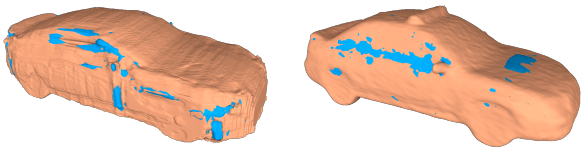

Figure 3. We overlay the reconstruction results of the classification and regression branches under the BR framework as shown in Figure 2. The classification results are highlighted in orange while the regression results are marked with blue. The misalignment of the two branches' results leads to the incomplete reconstruction in Figure 2.

	Chamfer-L_2	
	3K	300
BR	0.312	1.025
3C	**0.112**	**0.595**

	Chamfer-L_2	
	10K	3K
BR	0.095	0.314
3C	**0.071**	**0.258**

Table 1. Left: results of point cloud completion for closed watertight cars from 3000 and 300 points. Right: results of point cloud completion for unprocessed cars from 10000 and 3000 points. Chamfer distance is reported in $\times 10^{-4}$.

Reconstruction from point cloud. We also validate the performance of both candidate frameworks in the task of surface reconstruction from sparse point cloud. Specifically, we evaluate their performance on reconstructing both closed and open surface with the same setting as that of the main paper. We show in Figure 2 that in the BR framework, though both the classification and regression branches can generate reasonable reconstructions, the final merged results still exhibit incompleteness. We further demonstrate the cause of incomplete reconstructions in the overlaid visualization of the two branches (Figure 3). Since the results of the two branches are not perfectly aligned due to the different natures of their tasks, the classification branch would mistakenly remove part of the regressed surfaces generated by the regression branch. This could render holes and discontinuity in the results of the BR method. In comparisons, the 3C method does not suffer from such a problem as it only requires a single branch to generate the final reconstruction. This is also reflected in the quantitative measurements in Table 1.

Discussion. We have evaluated the performance of both candidate frameworks in two different tasks. In the applications where the binary classification and regression branches are well aligned, e.g. the shape reconstruction task, the BR method can lead to higher-quality results with smoother surface compared to the 3C approach. However, for more challenging scenarios, e.g. point cloud completion, where the two branches of BR framework may produce slightly deviated reconstructions, the final reconstruction may be incomplete despite that the two branches have obtained faithful reconstructions. In contrast, the 3C framework is robust over all kinds of task without the need of worrying about the misalignment issue. It would be an interesting future avenue to investigate how to resolve the misalignment problem of the BR method while enjoying its smooth nature.

2. Modeling Functions and Manifolds using 3PSDF

Following NDF, we train 3PSDF on 1 million points sampled from 1000 functions, which are either linear, parabola or sinusoids. Figure 4 shows the fitting results of 3PSDF to a variety of functions and manifolds. In Figure 4, red dots are points labeled as "inside" while cyan ones as "outside". "Nan" points are omitted for clear demonstration. As shown in the results, 3PSDF can faithfully model various functions and manifolds, which further validate that it is a versatile representation.

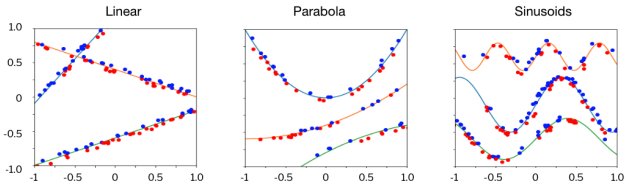

Figure 4. Function and manifold fitting using 3PSDF.

3. More Implementation Details

3.1. Reconstruction from Sparse Point Cloud

We use octree-based sampling to generate the ground-truth data for our approach. The sampling points are the corner points of the leaf cells generated by octree decomposition. In particular, we use depth of 6 for generating training data on pre-processed ShapeNet car category. For raw, unprocessed ShapeNet car, MGN, and 3D-Front, we use depth of 8, 7, and 9 respectively for training data generation. We train separate models for different numbers of input points. All models are trained using the same set of hyperparameters. For all experiments, we use the Adam optimizer with parameters $lr = 1e^{-4}$, $betas = (0.9, 0.999)$, $eps = 1e^{-8}$, $weight\_decay = 0$.

For MGN dataset, we split the data into train and test set with 9:1 ratio. For 3D-Front dataset, we extract 100 living rooms, 10 of which is used for testing and the rest is used for training. For NDF, we generate 1 million points for all experiments except the scene reconstruction task where we generate a more dense point containing 3 million points. The meshing results of NDF are obtained by running the script (including Ball Pivoting algorithm (BPA) and post-processing operations) provided by the authors in MeshLab. All the results are reported using the test data. For the ShapeNet car dataset, we use the common train and test split by [10].

3.2. Single-view Reconstruction on MGN

We evaluate and compare the representation capability of 3PSDF, DISN [10] and OccNet [8] on MGN dataset [3] for single-view 3D reconstruction. Each garment model in MGN dataset is rendered into an 256×256 RGB image from a front-view textured mesh. All the meshes and images are aligned with the same camera settings and normalized.

For 3PSDF, open surface models in MGN dataset are directly sampled with Octree-based subdivision at a resolution of 128^3, resulting in a mean sampling points of 300k across all models. The training batch size is set to 8 and the number of sampling points is 10k per sample. We use Adam optimizer with initial learning rate of 3e-4 and exponentially decayed to 0.99 at every 10k steps. For DISN and OccNet, models in MGN dataset are first converted to watertight form and then sampled with the default strategies used in the original papers. Each watertight model is sampled with 300k points, equivalent to that in 3PSDF. All the other training hyperparameters are set to default values.

MGN dataset is split into training and testing datasets with 9:1 ratio, and all 3 networks are evaluated at 20k epoches.

3.3. Single-view Reconstruction on ShapeNet

We use 17803 shapes from 5 categories of ShapeNet [4] for evaluation, including Airplane, Car, Lamp, Chair and Boat. We use the same image renderings (24 views per shape) and train/test split as Choy et al. [7].

We directly use the raw mesh of ShapeNet to generate the ground truth to train 3PSDF, while the competitive methods are trained using pre-processed watertight meshes. The ground truth 3PSDF values are sampled with resolution 128^3 and the results are evaluated using resolution 256^3. The images are all scaled to the resolution of 224×224. We first train the network for 30 epochs with learning rate 1e-4, and then finetune the network for 80 epochs using learning rate 5e-5. The batch size is set to 8 and the number of sampled points is 20k for one shape in each iteration during training. The reconstruction results are post-processed with simple hole filling and smoothing.

4. Network Structure

4.1. Network Architecture for Shape Reconstruction

Figure 5 shows the detailed network structure for the experiment of shape reconstruction. In particular, the network follows the design of the auto-decoder [9] which does not requires an encoder for learning the shape priors of training data. The input to the decoder contains: 1) a 512-dimensional per-object latent code, that is learned during training, and 2) a point feature obtained after applying point feature extractor to the 3D coordinate of the query point.

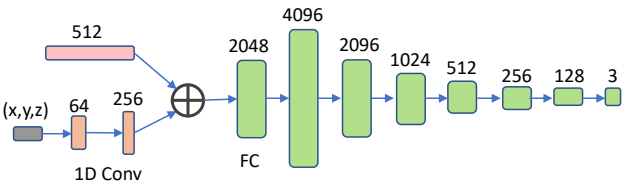

Figure 5. Network structure for shape reconstruction.

The point feature extractor is implemented using 1D convolutional operator. The concatenation of the latent code and the point feature is then fed into the decoder which consists of multiple fully connected layers. The output layer of the decoder predicts the per-class probability for the 3 categories defined by 3PSDF.

4.2. Network Architecture for Reconstruction from Point Cloud

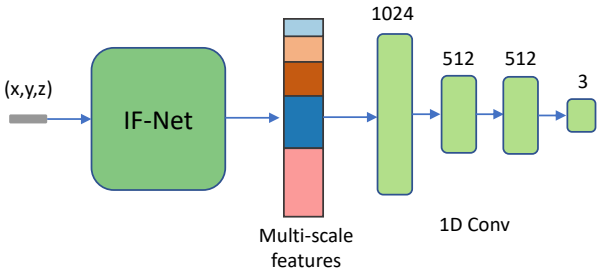

Figure 6. Network structure for reconstruction from point cloud.

We show the detailed network structure for reconstruction from point cloud in Figure 6. To ensure fair comparison, we use the identical network with NDF [6], which is based on IF-Net [5], for extracting the features from the input point cloud. The extracted multi-scale point features are then fed into the decoder. The decoder is implemented using four 1D convolution layers, where the last layer predicts the per-class probability for 3PSDF.

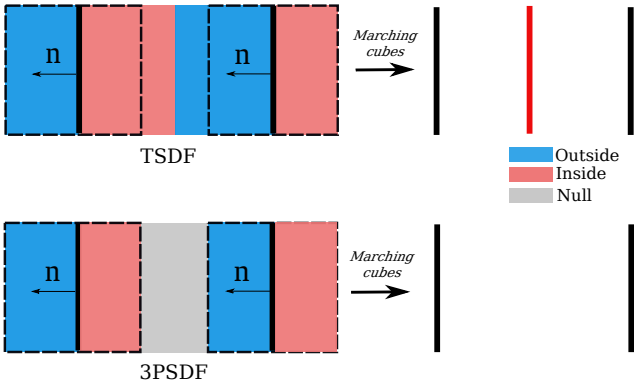

Figure 7. Comparison between TSDF and the proposed 3PSDF. For reconstructing two adjacent single layers of mesh, TSDF would introduce artifacts (the red layer show on the right of first row) to the reconstruction result.

4.3. Network Architecture for SVR

Figure 8 shows the detailed network architecture for 3D reconstruction based on single-view images. The network takes a set of sampled 3D points and a single view image as input. We use several 1D convolution layers to obtain the point features and a VGG-16 (with batch normalization) architecture to encode the input image. We adopt a two-stream network architecture, where the point features are concatenated with global and local image features respectively, and then fed into two branches to predict the 3PSDF.

The global image features are obtained from an average pooling and a fully connected layer at the end of the image encoder. For the local features, we project the input 3D points to the image plane and retrieve the features on each feature map using the projected coordinates. The retrieved features on each feature map are concatenated together to obtain a local image feature vector.

The decoder has two streams with the same structure, each of which consists of a set of fully connected layers to predict the 3PSDF separately. The outputs from the two branches are summed up and passed through a Softmax layer to obtain the final prediction.

5. Comparison with TSDF

Truncated Signed Distance Field (TSDF) is widely used in obtaining reconstruction results from the volumetric range data, e.g. the RGBD stream from depth sensors. One mainstream application of TSDF is large-scale tracking and mapping in reconstructing 3D scenes. As one may have seen open surfaces, e.g. the walls in the reconstructed 3D environment, can be reconstructed using TSDF, we provide detailed comparisons here stating the difference between TSDF and 3PSDF regarding the ability of modeling surfaces with arbitrary topologies.

The motivation of introducing TSDF is to set a lower bound of reconstruction error during the fusion of different SDFs converted from the depth maps. In particular, in real-world scanning, the raw data obtained from the depth sensor is highly likely to be contaminated by the noises. In practice, the depth maps are converted into SDFs in order to fuse the per-frame observation into a more complete reconstruction in the canonical space. However, the most widely adopted way of fusing the SDFs is based on weighted summation, where the errors brought by each SDF would be accumulated and affecting the previously fused results. TSDF alleviates this issue by clipping the minimum and maximum signed distance value and hence prevents the summed TSDFs from deviating too much from the ground-truth value.

After analyzing the motivation of TSDF, we can better understand the difference between TSDF and our proposed 3PSDF. (1) Unlike 3PSDF, TSDF remains a binary-sided signed distance function which only has positive and negative signs. This could render TSDF failed to represent open surfaces without introducing artifacts in many cases. As shown in Figure 7 upper row, for two adjacent surfaces with consistent normals, the positive and negative signs would intersect with each other in the middle region where the SDFs are truncated to maximum and minimum respectively. This leads to an additional surface/artifact (the red boundary on the right) if meshing such a field using the Marching Cubes algorithm. In contrast, 3PSDF can achieve artifact-free reconstruction by inserting a NULL layer in between to prevent the formation of the additional decision boundary.

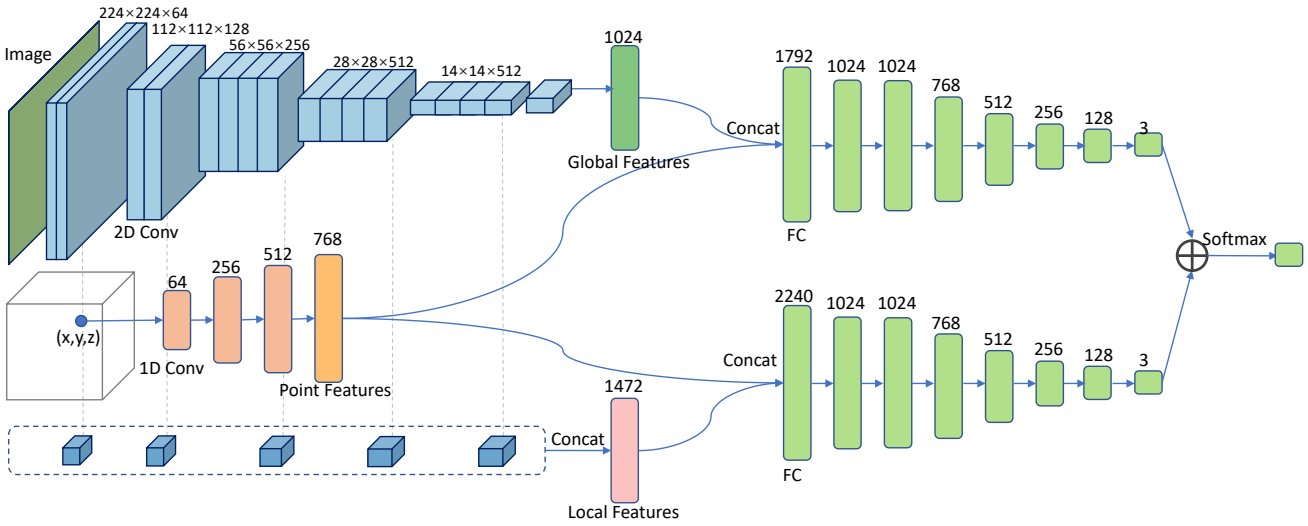

Figure 8. Detailed network architecture for single view reconstruction.

(2) The way that TSDF models open surfaces is completely different from that of 3PSDF. In particular, TSDF generates open surfaces by space clipping, where only the field within a bounded volume is converted into mesh. In comparison, 3PSDF is able to model open surfaces by directly meshing the entire 3D space without requiring a clipping bounding volume.

6. More Results

Reconstruction of closed surfaces from sparse point cloud. We provide more qualitative comparison results with the state-of-the-art approaches on the task of shape reconstruction from sparse point cloud. In Figure 9, we show the reconstruction result using the models trained on preprocessed ShapeNet car data (watertight mesh with inner structure removed) provided by [10].

Reconstruction of complex surfaces from sparse point cloud. In Figure 10 we provide more qualitative comparisons of shape reconstruction results of complex surfaces that contain both closed and open surfaces. All the candidate approaches, including ours, are trained on on raw, unprocessed ShapeNet car data, which contain inner structures and open surfaces. As seen in the highlighted regions within the red rectangles, our approach is able to generate shapes with consistent normals even when the ground truth data may contain flipped face patches.

Reconstruction of 3D scenes from sparse point cloud. In Figure 11, we show more visual comparisons of scene reconstruction results. The input point cloud (for both main paper and supplementary material) contains 50K points. Note that we are not able to generate plausible meshing result for NDF even after experimenting with various parameters of BPA algorithm. Hence we show the raw output point cloud of NDF in the closeup figure.

Single-view reconstruction. We include more qualitative comparison results on the test set of ShapeNet in Figure 12.

References

[1] Adobe. https://www.mixamo.com/, 2017. 1

[2] Matan Atzmon and Yaron Lipman. Sal: Sign agnostic learning of shapes from raw data. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020. 6

[3] Bharat Lal Bhatnagar, Garvita Tiwari, Christian Theobalt, and Gerard Pons-Moll. Multi-garment net: Learning to dress 3d people from images. In *IEEE International Conference on Computer Vision (ICCV)*. IEEE, oct 2019. 1, 3

[4] Angel X Chang, Thomas Funkhouser, Leonidas Guibas, Pat Hanrahan, Qixing Huang, Zimo Li, Silvio Savarese, Manolis Savva, Shuran Song, Hao Su, et al. Shapenet: An information-rich 3d model repository. *arXiv preprint arXiv:1512.03012*, 2015. 3

[5] 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. 4, 6

[6] Julian Chibane, Aymen Mir, and Gerard Pons-Moll. Neural unsigned distance fields for implicit function learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, December 2020. 4, 6, 7

[7] 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 *European conference on computer vision*, pages 628–644. Springer, 2016. 3

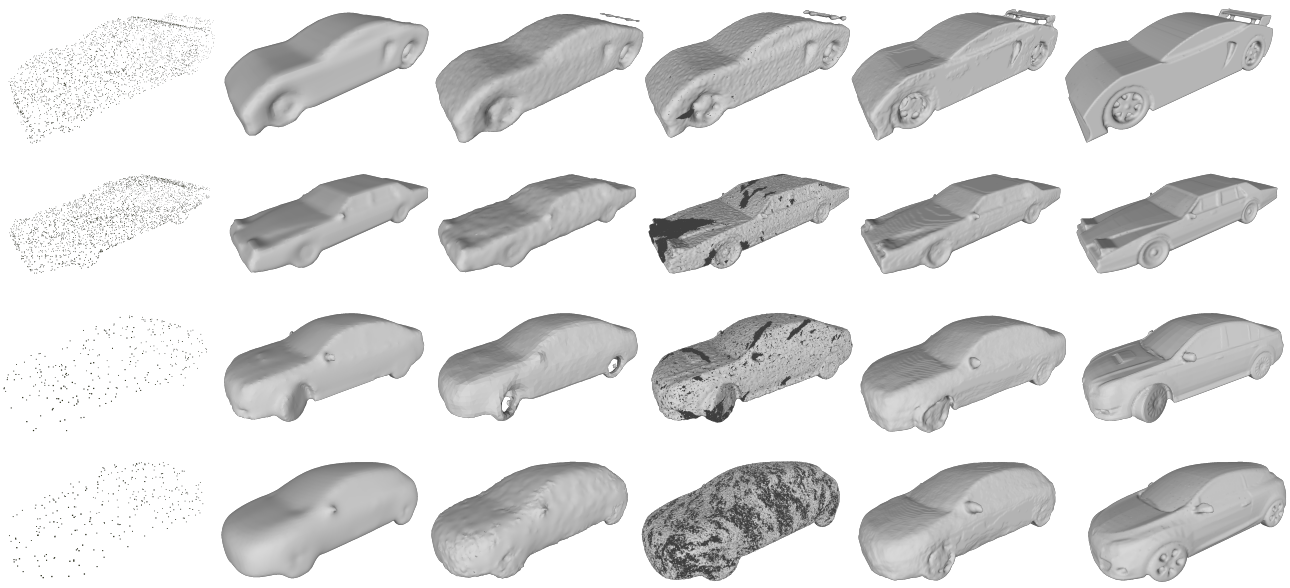

Figure 9. More shape reconstruction results trained on watertight data. We show four groups of results: the first two rows are reconstructed from 3000 points while the last two rows are generated given 300 points.

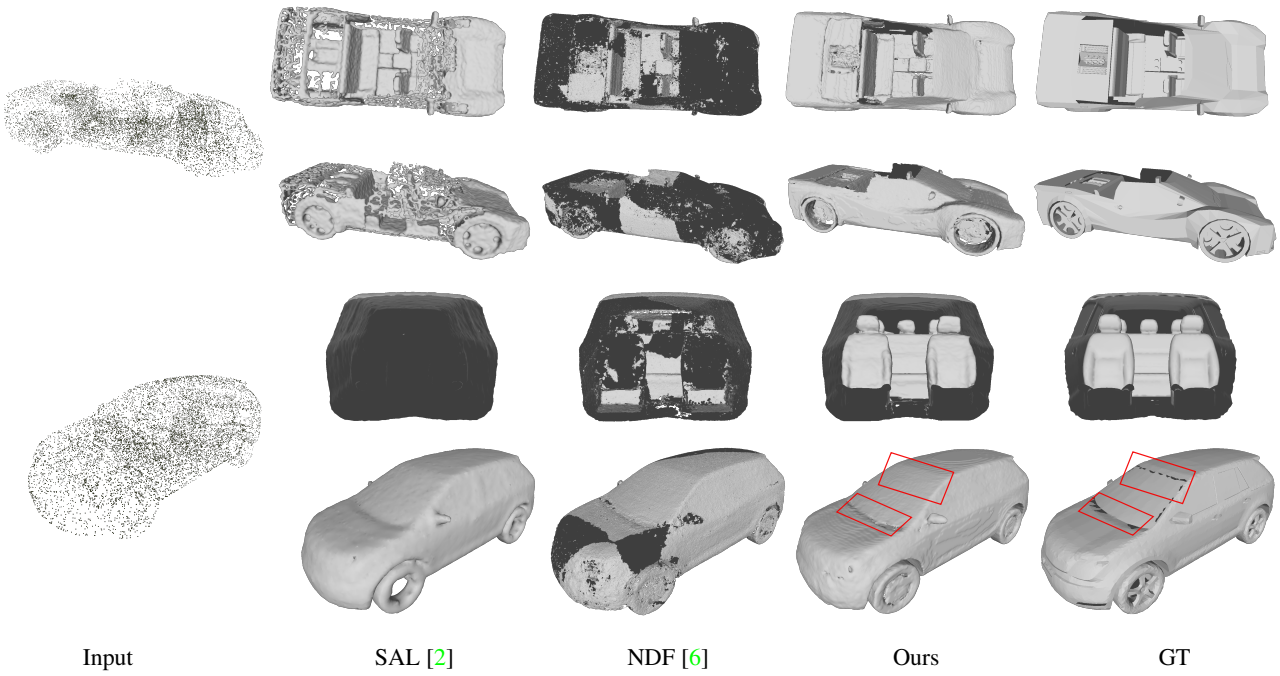

Figure 10. More shape reconstruction results trained on unprocessed, raw data. For each group of results, we show the input (10K points) on the left and two rows of corresponding results on the right. For the second group of result, we show the inner structure of reconstruction on top of an external view. The highlighted regions within the red rectangles show that our method can generat reconstruction results with consistent normals even when the ground-truth data contain flipped triangles.

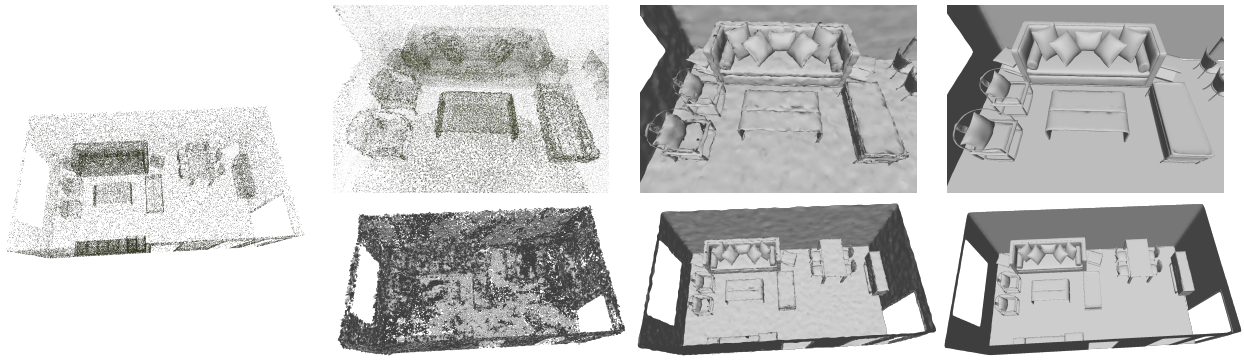

Figure 11. Scene reconstruction results from sparse point cloud. For each method, we show both the closeups (first row) and the global view (second row). NDF results contain 3 million points. Note that since we are not able to generate plausible meshing results for NDF even after experimenting with various BPA parameters, we show the output raw point cloud in the closeup of NDF. The other results are displayed in mesh form.

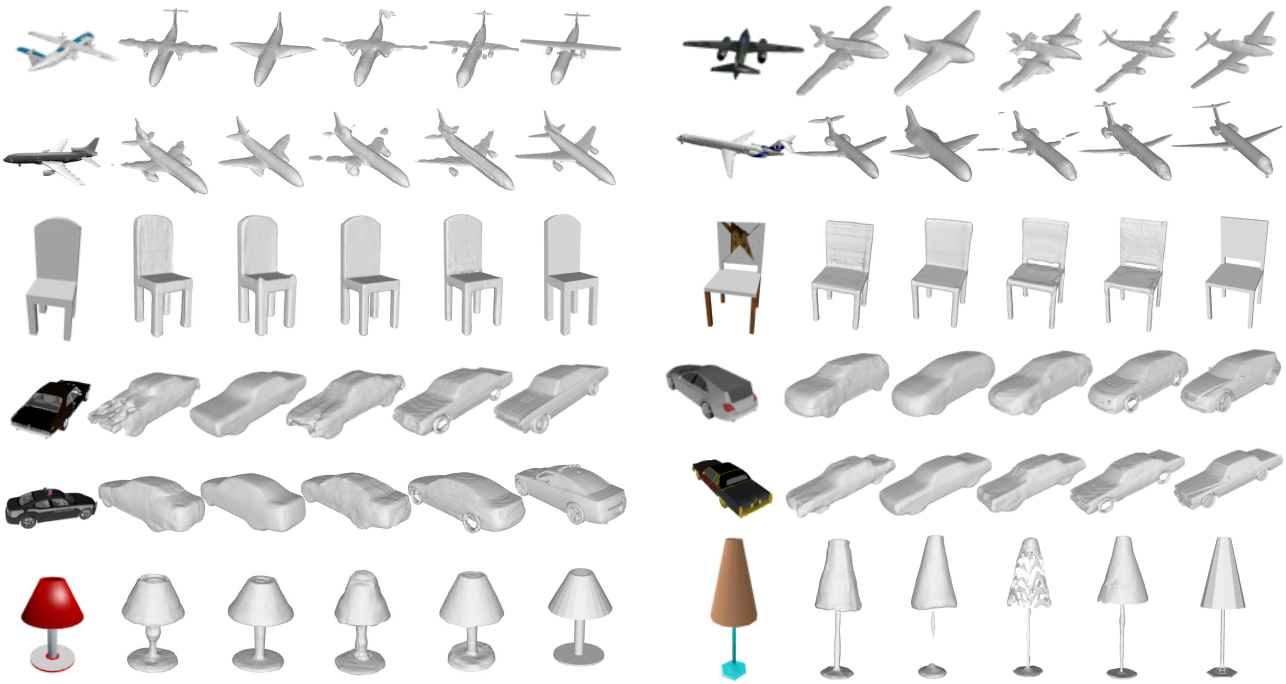

Figure 12. More qualitative comparison results with SOTA single-view reconstruction methods based on implicit functions.

[8] Lars Mescheder, Michael Oechsle, Michael Niemeyer, Sebastian Nowozin, and Andreas Geiger. Occupancy networks: Learning 3d reconstruction in function space. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4460–4470, 2019. 3, 6

[9] Jeong Joon Park, Peter Florence, Julian Straub, Richard Newcombe, and Steven Lovegrove. DeepSDF: Learning continuous signed distance functions for shape representation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 165–174, 2019. 3

[10] Qiangeng Xu, Weiyue Wang, Duygu Ceylan, Radomir Mech, and Ulrich Neumann. Disn: Deep implicit surface network for high-quality single-view 3d reconstruction. *Advances in Neural Information Processing Systems (NeurIPS)*, 2019. 3, 5