

3D Reconstruction of Novel Object Shapes from Single Images

Anh Thai*, Stefan Stojanov*, Vijay Upadhyaya, James M. Rehg
 Georgia Institute of Technology
 {athai6, sstojanov, vupadhyaya6, rehg}@gatech.edu

Abstract

The key challenge in single image 3D shape reconstruction is to ensure that deep models can generalize to shapes which were not part of the training set. This is difficult because the algorithm must infer the occluded portion of the surface by leveraging the shape characteristics of the training data, and can therefore be vulnerable to overfitting. Such generalization to unseen categories of objects is a function of architecture design and training approaches. This paper introduces SDFNet, a novel shape prediction architecture and training approach which supports effective generalization. We provide an extensive investigation of the factors which influence generalization accuracy and its measurement, ranging from the consistent use of 3D shape metrics to the choice of rendering approach and the large-scale evaluation on unseen shapes using ShapeNetCore.v2 and ABC. We show that SDFNet provides state-of-the-art performance on seen and unseen shapes relative to existing baseline methods GenRe[49] and OccNet[27]. We provide the first large-scale experimental evaluation of generalization performance. The codebase released with this article will allow for the consistent evaluation and comparison of methods for single image shape reconstruction.¹

1. Introduction

Recently, there has been substantial progress in the development of deep models that reconstruct the 3D shape of an object from a single RGB image [42, 7, 27, 46, 6, 28, 39, 12, 16], enabled by the availability of 3D model datasets such as ShapeNet [4] and ModelNet [43]. In spite of these advances, however, many basic questions remain unresolved, such as the most effective choice of object coordinate system and 3D shape representation, or the impact of the object rendering approach on reconstruction performance. Moreover, recent works [38, 33] have identified sev-

Method	2.5D Estimator	Continuous Representation	3-DOF Viewer-Centered
GenRe [49]	✓	✗	✗
Multi-View [33]	✓	✗	✗
OccNet [27]	✗	✓	✗
DISN [46]	✗	✓	✗
SDFNet	✓	✓	✓

Table 1: Characteristics of different single-view 3D shape reconstruction methods.

eral challenges in ensuring that learned models can generalize to novel shapes, and in accurately assessing generalization performance. In this paper, we present a novel reconstruction architecture, *SDFNet* and identify four key issues influencing generalization performance. *SDFNet* leverages a continuous, implicit representation of 3D object shape based on signed distance fields (SDF), which achieves substantial improvements in reconstruction accuracy relative to previous state-of-the-art methods employing both discrete [49] and continuous [27] shape representations. There are three key issues that affect generalization performance: the choice of coordinate representation, the impact of object rendering and the use of 2.5D sketch representations (absolute depth map and surface normals) as network input. The fourth key issue is the effective evaluation of generalization.

The first issue that we investigate is the choice of coordinate representation for reconstruction. Prior work has shown that using viewer-centered (VC) coordinates results in improved generalization in comparison to object-centered (OC) representations [38, 33], by discouraging the model from learning to simply retrieve memorized 3D shapes in a canonical pose. Our experiments highlight an additional key issue, by showing that VC generalization depends critically on sampling the full range of possible object views including camera tilt variation, which has not been used in prior works.

The second issue that we address is large scale performance evaluation. Recent work by Zhang et al. [49] proposed testing on *unseen* ShapeNet categories as an effective test of generalization. They present results using 3 training and 10 testing classes. We extend this approach sig-

*Equal contribution.

¹The source code and datasets which are needed to reproduce the experimental results in this paper are available at <https://github.com/devlearning-gt/3DShapeGen>

nificantly by presenting the first results to use all of the meshes in ShapeNetCore.v2 for testing generalization to unseen categories. Our generalization task involves training on 13 classes and testing on 42 unseen classes, which contain two orders of magnitude more meshes than prior works (Sec. 4.1). We also present the first analysis of cross-dataset generalization in 3D shape reconstruction, through models trained on ShapeNet and ABC [19] (Sec. 4.6).

The third issue that we investigate is the impact of *object rendering* on generalization performance, which has surprisingly not been addressed in prior work. We demonstrate that rendering choices have a substantial impact on generalization performance, resulting in an F-Score drop of 0.31 from the most basic to the most complex rendering scenarios (Sec. 4.5).

The fourth issue concerns the use of an explicit reconstruction of depth information (i.e. a 2.5D sketch representation) as the network input. This approach was explored in the Multi-View [33] and GenRe [49] architectures as a means to facilitate good generalization performance. Our work provides a more extensive assessment of the value of a 2.5D sketch input (Sec. 4.3). In summary, this paper makes the following contributions: (1) Introduction of the SDFNet architecture, which combines 2.5D sketch representation with SDF object representation and achieves state-of-the-art reconstruction performance relative to the OccNet and GenRe baselines; (2) First comprehensive evaluation of the four key issues affecting generalization performance in single-image 3D shape reconstruction; (3) Introduction of a new large-scale generalization task involving all meshes in ShapeNetCore.v2 and a subset of the ABC [19] dataset, with detailed consideration of rendering issues.

The paper is organized as follows: Sec. 2 presents our work in the context of prior methods for shape reconstruction. Sec. 3 provides a description of our approach and introduces the SDFNet architecture. It also addresses the consistent evaluation of metrics for surface-based object representations. Sec. 4 presents our experimental findings. It includes a discussion of consistent practices for rendering images and a training/testing split for ShapeNetCore.v2 which supports the large-scale evaluation of generalization to unseen object categories.

2. Related Work

There are two sets of related works on single image 3D reconstruction which are the closest to our approach. The first set of methods were the first to employ continuous implicit surface representations [27, 46]. Mescheder et al. [27] utilized continuous occupancy functions, while Xu et al. [46] shares our use of signed distance functions (SDFs). We differ from these works in our use of depth and normal (2.5D sketch) intermediate representations, along with other differences discussed below. The second set

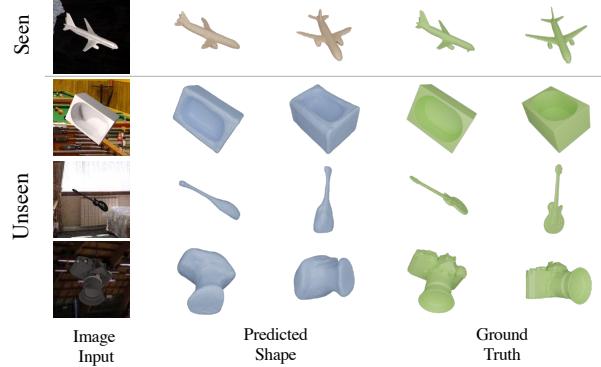


Figure 1: Reconstruction from classes seen during training and novel classes not seen during training of SDFNet.

of related works pioneered the use of unseen shape categories as a means to investigate the generalization properties of shape reconstruction [49, 33]. In contrast to SDFNet, these method train and test on a small subset of ShapeNetCore.v2 classes, and they utilize discrete object shape representations. We share with these methods the use of depth and normal intermediate representations. Additionally, we differ from all prior works in our choice of object coordinate representation (VC with 3-DOF, see Sec. 3) and in the scale of our experimental evaluation (using all 55 classes of ShapeNetCore.v2 to test generalization on seen and unseen classes, Sec. 4.1), and our investigation into the effects of rendering in Sec. 4.5. We provide direct experimental comparison to GenRe [49] and OccNet [27] in Sec. 4.3, and demonstrate the improved performance of SDFNet. GenRe and DISN [46] require known intrinsic camera parameters in order to perform projection. In contrast, SDFNet does not require projection, and only regresses camera translation for estimating absolute depth maps. See Fig. 1 for a summary of the relationship between these prior works and SDFNet.

Other works also perform 3D shape reconstruction [42, 7, 39, 12, 16, 34, 13, 24, 25] but differ from this work since they perform evaluation on categories seen during training, on small scale datasets, and use different shape representations such as voxels [42, 7], meshes [39, 12, 16, 24], and continuous implicit representations with 2D supervision [25]. Prior works on seen categories used the 13 largest ShapeNet categories, while [38] uses all of them. We use all 55 ShapeNet categories, but train on 13 and test on 42.

2.5D Sketch Estimation. We utilize the 2.5D sketch estimator module from MarrNet [41] to generate depth and surface normal maps as intermediate representations for 3D shape prediction. We note that there is a substantial body of work on recovering intrinsic images [1, 15, 9, 5, 20, 22, 37] which infers the properties of the visible surfaces of objects, while our focus is additionally on hallucinating the

self-occluded surfaces and recovering shape for unseen categories. More recent works use deep learning in estimating depth [9, 5, 20, 22], surface normals [14, 48], joint estimation [29, 30], and other intrinsic images [23, 21, 2].

Generative Shape Modeling. This class of works [28, 6, 18] is similar to SDFNet in the choice of shape representation but primarily focuses on learning latent spaces that enable shape generation. IM-NET [6] contains limited experiments for single view reconstruction, done for one object category at a time, whereas DeepSDF [28] also investigates shape completion from partial point clouds.

3D Shape Completion. This class of works [35, 10, 31, 47, 36, 11, 33] is not directly related to our primary task because we focus on 3D shape reconstruction from single images. Note that in Secs. 4.4 and 4.6 we utilize ground truth single-view 2.5D images as inputs, similar to these prior works.

3. Method

In this section we introduce our *SDFNet* architecture, illustrated in Fig. 2, for single-view 3D object reconstruction based on signed distance fields (SDF). We describe our architectural choices along with the design of a 3-DOF approach to viewer-centered object representation that improves generalization performance. We are the first to use the 3-DOF representation and our architecture is novel for 3D shape reconstruction (see Tbl. 1). Our approach achieves state-of-the-art performance in single-view 3D shape reconstruction from novel categories.

SDFNet Architecture Our deep learning architecture, illustrated in Fig. 2, incorporates two main components: 1) A 2.5D sketch module that produces depth and normal map estimates from a single RGB input image, followed by 2) a continuous shape estimator that learns to regress an implicit 3D surface representation based on signed distance fields (SDF) from a learned feature encoding of the 2.5D sketch. The use of depth and normals as intermediate representations is motivated by the observation that such intermediate intrinsic images explicitly capture surface orientation and displacement, key object shape information. As a result, the shape reconstruction module is not required to jointly learn both low-level surface properties and the global attributes of 3D shape within a single architecture. Prior works that study generalization and domain adaptation in 3D reconstruction [49, 41] have also adopted these intermediate representations and have demonstrated their utility in performing depth estimation for novel classes. Note that, following the approach in [49], we use ground truth silhouette when converting from a normalized depth map to an absolute depth map. Our findings are that models trained with a 2.5D sketch have a slight performance improvement in generalization to unseen classes, in comparison to models trained directly from images, but are less

robust to novel variations in lighting and object surface reflection (see Secs. 4.3 and 6).

Our SDF network is adapted from [27], with the distinction that rather than producing binary occupancy values, our model regresses SDF values from the mesh surface. Mescheder et al. [27] point out that randomly sampling 3D points from a unit cube for training gives the best performance. However, since the SDF representation also indicates the distance of the points to the surface of the mesh, it captures the surface shape more precisely. Therefore, it is beneficial to sample input points more densely closer to the surface during training. We modify the ground truth SDF generation procedure defined by [46] to better accommodate our training process. Specifically, we first rescale the mesh to fit inside a unit cube and set the mesh origin to the center of this cube, then we sample 50% of the training points within a distance of 0.03 to the surface, 80% within a distance of 0.1 and 20% randomly in a cube volume of size 1.2. Note that since we are training with viewer-centered coordinate system where the pose of the testing data is unknown, it is important that training signals come from 3D points sampled from a volume of sufficient size. This ensures that during mesh generation at inference, the algorithm does not need to extrapolate to points outside of the training range. During training, we scale the loss by 4 for points within $d = 0.01$ of the surface to improve estimation accuracy of points near the mesh surface. To perform viewer-centered training, we generate ground truth SDF values for the canonical view and apply rotations during each data loading step. Note that this approach is not feasible with voxels, which have to be resampled offline at a significant storage cost. Using an SDF representation is more memory efficient by an order of magnitude. During testing, we generate the mesh from predicted SDF values using Marching Cubes [26]. In this step, points are sampled uniformly in a cube of size $\sqrt{3}$ to accommodate for different unknown object poses.

Training Procedure. During training, in the first stage we optimize using MSE loss on the predicted 2.5D representations. The second stage is trained by optimizing the L_1 loss between the ground truth SDF and the predicted values of the input 3D points. Training is done using the Adam optimizer [17] with default parameters.

2-DOF vs. 3-DOF Viewer-centered Coordinate Representation A basic question underlying all approaches to 3D shape reconstruction is the choice of coordinate representation. Early works adopted an object-centered (OC) representation, but recent works [49, 33, 38] have argued that adopting a viewer-centered coordinate representation is helpful in preventing shape regressors from performing reconstruction in a recognition regime (i.e. memorizing training shapes in canonical pose) and encouraging more effective generalization. Our approach extends this observation

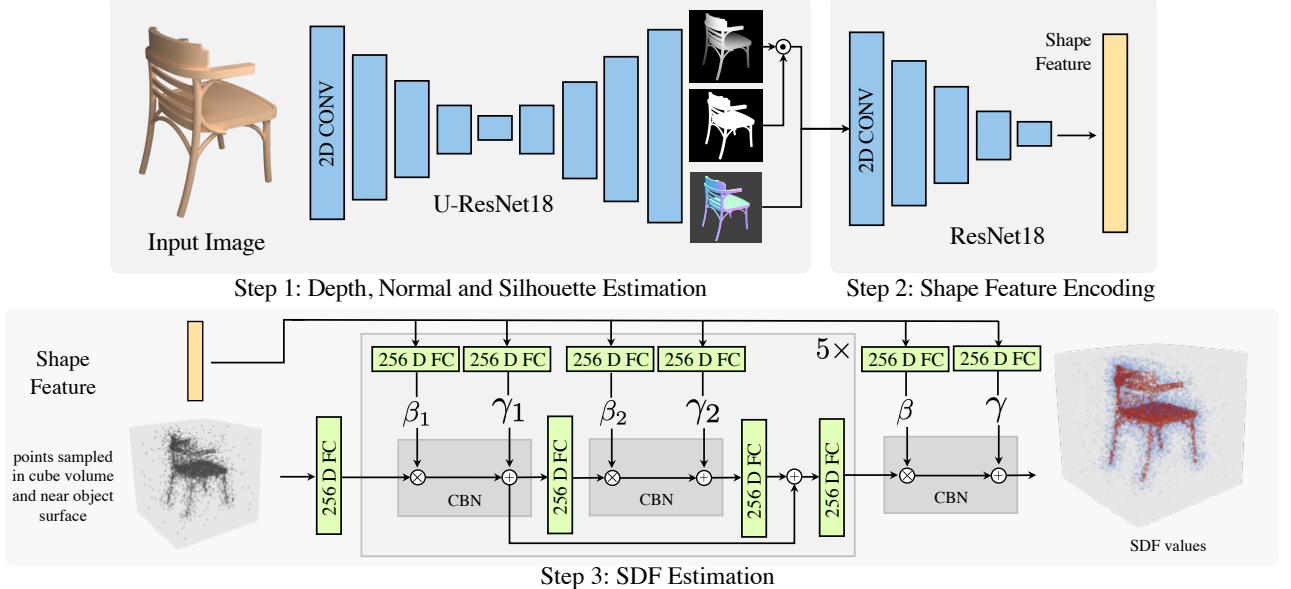


Figure 2: Two-stage SDFNet architecture: 2.5D estimation and 3D shape completion. The 2.5D sketch estimator is a U-ResNet18 [32] architecture as in [49]. Given the depth and surface normal output, the 3D completion module produces a feature encoding used to produce conditional batch norm (CBN [8]) values for an MLP as in OccNet[27] and assigns SDF values to a sampled set of points in 3D.

in an important way. While prior works adopted a viewer-centered (VC) representation [49, 38, 33], object models were nonetheless oriented in a fixed, *canonical* pose, and views were generated by varying camera azimuth and elevation (i.e. 2-DOF of viewpoint variation), which we refer to as 2-DOF VC. Specifically, the vertical axis of object models in the same categories is aligned with the same gravity reference. As a consequence, the set of generated views remains biased to the canonical pose for each category. Our proposed solution is to add camera tilt along with rotation and elevation, which we refer to as 3-DOF VC (see Fig. 6). We demonstrate in Sec. 4.4 that the 3-DOF VC approach results in better generalization performance in comparison to both OC and 2-DOF VC object representations.

4. Experiments

This section presents the results of our large-scale experimental evaluation of SDFNet and related methods. It is organized as follows: Our choice of datasets and train/test splits is outlined in Sec. 4.1, followed by a discussion of metrics for evaluation in Sec. 4.2. Sec. 4.3 reports on the generalization performance of SDFNet, GenRe [49], and OccNet [27], and investigates the utility using depth and normals (i.e. 2.5D sketch) as an intermediate representation. In Sec. 4.4, we investigate the impact of the choice of object coordinate representation on generalization ability, while Sec. 4.5 discusses the impact of the image rendering process. Last, in Sec. 4.6 we analyze cross-dataset generalization for 3D shape reconstruction. Unless otherwise

specified, the images used in all experiments were rendered with light and reflectance variability and superimposed on random backgrounds from SUN Scenes [45], as described in Sec. 4.5 (the LRBg condition).

For all of the experiments, we train on one random view per object for each epoch. Testing is done on one random view per object in the test set. We find that the standard deviation for all metrics is approximately 10^{-4} based on three evaluation runs.

4.1. Datasets, Evaluation Split, and Large Scale Generalization

Datasets Our experiments use all 55 categories of ShapeNetCore.v2 [4]. Additionally, in Sec. 4.6 we use a subset of 30,000 meshes from ABC [19]. To generate images, depth maps, and surface normals, we implemented a custom data generation pipeline in Blender [3] using the Cycles ray tracing engine. Our pipeline is GPU-based, supports light variability—point and area sources of varying temperature and intensity, and includes functionality that allows for specular object shading of datasets such as ShapeNet, which are by default diffuse.

Data Generation In our experiments, we use 25 images per object in various settings as illustrated in Fig. 6 resulting in over 1.3M images in total. In contrast with prior approaches to rendering 3D meshes for learning object shape [38, 33, 7], we use a ray-tracing renderer and study the impact of lighting and specular effects on performance. The ground truth SDF generating procedure is described in

Sec. 3. We convert SDF values to mesh occupancy values by masking: $\mathbb{1}\{\text{sdf} \leq i\}$ where i is the isosurface value.

Data Split We use the 13 largest object categories of ShapeNetCore.v2 for training as *seen* categories, and the remaining 42 categories as *unseen*, testing data which is not used at all during training and validation. For the 30K ABC meshes, we use 20K for training, 7.5K for testing, and the rest for validation.

Scaling up Generalization Our generalization experiments are the largest scale to date, using all of the available objects from ShapeNetCore.v2. Our testing set of unseen classes consists of 12K meshes from 42 classes, in comparison to the 330 meshes from 10 classes used in [49].

4.2. Metrics

Following [38], we use F-Score with percentage distance threshold d (FS@ d) as the primary shape metric in our experiments, due to its superior sensitivity. We also report the standard metrics IoU, NC (normal consistency) and CD (chamfer distance), which are broadly-utilized despite their known weaknesses (see Fig. 3.)

A significant practical issue with metric evaluation which has not been discussed in prior work is what we refer to as the *sampling floor* issue. It arises with metrics, such as CD, NC and FS, that require surface correspondences between ground truth and predicted meshes. Correspondences are established by uniformly sampling points from both mesh surfaces and applying nearest neighbor (NN) point matching. Note that the accuracy of NN matching is a function of the number of sampled points. For a given dataset and fixed number of samples, there is a corresponding sampling floor, which is a bound on the possible error when comparing shapes. We note that sampling 10K points (as suggested in [38]) and comparing identical shapes using FS@1 in ShapeNetCore.v2 results in an average sampling floor of 0.8, which admits the possibility of significant error in shape comparisons (since comparing two identical meshes should result in an FS of 1). In our experiments, we use FS@1 and sample 100K points, and have verified that the sampling floor is insignificant in this case. See Supplement C for additional analysis and discussion.

4.3. Performance Evaluation of SDFNet

We evaluate the generalization performance of SDFNet relative to two prior works: GenRe [49] and OccNet [27]. GenRe defines the state-of-the-art in single-image object reconstruction for unseen object categories, while OccNet is representative of recent works that use continuous implicit shape representations for shape reconstruction. We use 2-DOF VC data (see Sec. 3) to compare with these baseline methods. While GenRe was designed for 2-DOF VC data, OccNet was designed for OC data and was adapted to facilitate a direct comparison. We refer to the adapted model

Method	Seen Classes				Unseen Classes			
	CD	IoU	NC	FS@1	CD	IoU	NC	FS@1
OccNet VC	0.078	0.72	0.78	0.27	0.11	0.67	0.76	0.22
GenRe	0.153	N/A	0.60	0.12	0.172	N/A	0.61	0.11
SDFNet	0.05	0.72	0.79	0.41	0.08	0.66	0.76	0.31

Table 2: Performance comparison on seen and unseen classes of 2-DOF ShapeNetCore.v2 testing data with LRBg renderings, using four commonly used evaluation metrics with a focus on FS@1. IoU is omitted for GenRe as per the authors’ recommendation since the meshes are not guaranteed to be watertight.

Method	Seen Classes				Unseen Classes			
	CD	IoU	NC	FS@1	CD	IoU	NC	FS@1
SDFNet Img	0.088	0.65	0.73	0.25	0.10	0.61	0.74	0.23
SDFNet Est	0.082	0.68	0.75	0.29	0.099	0.64	0.74	0.26
SDFNet Orcl	0.041	0.77	0.81	0.51	0.044	0.75	0.83	0.51

Table 3: Performance comparison for 3 versions of SDFNet that differ only in their inputs: Image only (SDFNet Img), Estimated 2.5D (SDFNet Est), and ground truth 2.5D input (SDFNet Orcl), on seen and unseen classes using 3-DOF VC and LRBg renderings.

as OccNet VC. We note that performing these experiments with GenRe required significant reimplementation effort in order to generate ground truth spherical maps and voxel grids for a large number of additional ShapeNetCore.v2 models (see Supplement B for the details). All three parts of GenRe: depth estimation, spherical inpainting, and voxel refinement, are trained using the code base provided by the authors until the loss converges (i.e. 100 epochs without improvement in validation loss.)

Our findings in Tbl. 2 demonstrate the superior generalization performance of SDFNet relative to OccNet VC and GenRe. Compared to OccNet VC, SDFNet performs better for CD and FS@1, which can be interpreted as an improved ability to capture the thin details of shapes as a result of the better-defined isosurface of the SDF representation. The performance difference relative to GenRe shows the advantage of using a continuous implicit representation. These findings further suggest that good generalization performance can be achieved without explicit data imputation procedures, such as the spherical inpainting used in GenRe. Qualitative results are shown in Fig. 4. SDFNet demonstrates the ability to capture concavity better than OccNet VC and GenRe in the sink (third row) and the hole of the watch (second row).

We performed an experiment to evaluate the effect of

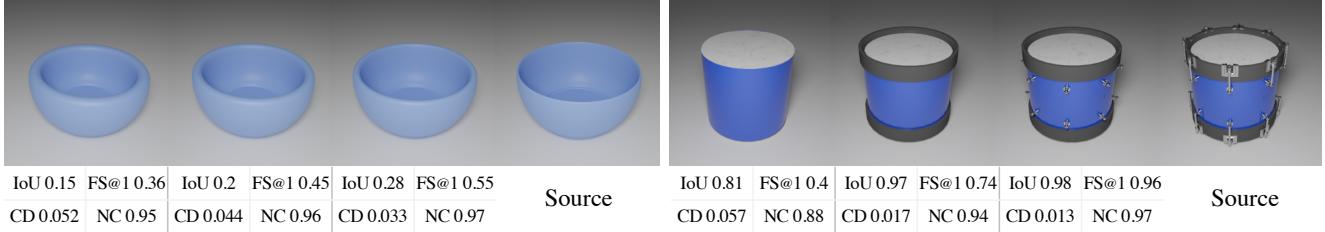


Figure 3: Issues with commonly used metrics. Left: significant change in IoU despite small change in shape for thin object. Right: small change in NC, CD and IoU despite significant loss of surface detail.

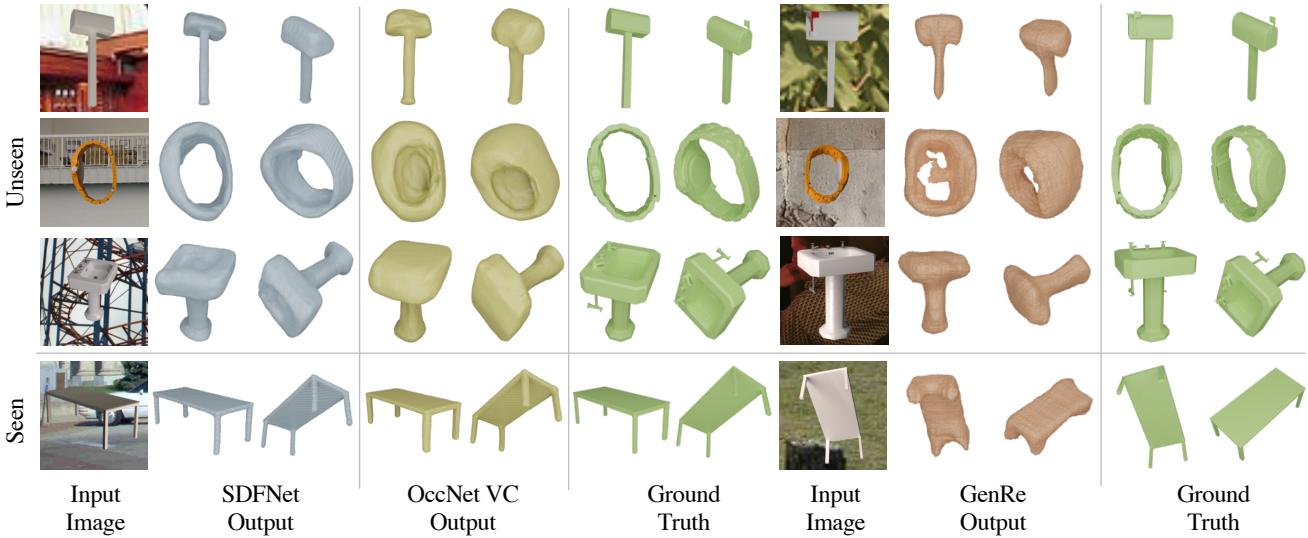


Figure 4: Qualitative results of SDFNet, OccNet VC and GenRe on seen and unseen classes of 2-DOF viewpoint ShapeNet-Core.v2 testing data with LRBg renderings. Each column shows the results from a method in two different views: input view (left), other view (right). Note that we show GenRe’s performance on different renders of the same set of objects with the same rendering settings. See Supplement D for further discussion.

SDFNet’s intermediate representation (estimated surface depth and normals) on generalization. The results are reported in Tbl. 3. Three SDFNet models were trained: with image inputs only (no depth and normals, the SDFNet Img case), with estimated depth and normals (the standard case, SDFNet Est), and with ground truth depth and normals (the oracle case, SDFNet Orcl). Our findings show a slight improvement when using an intermediate representation consisting of surface depth and normals rather than regressing SDF from images directly. The result for SDFNet Oracle demonstrates that there is room for significant gains in performance by improving the accuracy of the depth and normal estimator. Note that all remaining subsections (below) are focused on additional evaluations of SDFNet.

4.4. Viewer-Centered Representations Improve Generalization Ability

In this section, we study the effect of the object coordinate representation on generalization performance, using

SDFNet with ground truth depth and surface normal images as inputs. Three different SDFNet models are trained using object centered (OC), 2-DOF viewer centered (2-DOF VC) and 3-DOF viewer centered (3-DOF VC) representations, as described in Sec. 3. OC and 2-DOF VC are trained on 2-DOF viewpoint data and 3-DOF VC is trained on 3-DOF viewpoint data. We present our findings in Tbl. 4. We present testing results under two conditions corresponding to 2-DOF (top) and 3-DOF (bottom) *testing* data. The 3-DOF VC model performs the best in the 3-DOF testing case (bottom table, bold scores are higher across all metrics), with a significant margin on both seen and unseen classes. This is perhaps not surprising, since the OC and 2-DOF VC models are not trained on 3-DOF data. However, this finding demonstrates the significant benefit arising from our 3-DOF training approach. When tested under the 2-DOF condition (top table), the performance drop in OC and 2-DOF VC from seen to unseen categories provides evidence that these models perform reconstruction in

2-DOF Testing Data										
Method	Seen Classes			Unseen Classes			FS@1	CD	IoU	NC
	CD	IoU	NC	CD	IoU	NC				
OC	0.040	0.71	0.82	0.51	0.093	0.58	0.74	0.33		
2-DOF VC	0.033	0.77	0.82	0.57	0.040	0.74	0.83	0.51		
3-DOF VC	0.038	0.77	0.82	0.53	0.041	0.75	0.84	0.52		

3-DOF Testing Data										
Method	Seen classes			Unseen classes			FS@1	CD	IoU	NC
	CD	IoU	NC	CD	IoU	NC				
OC	0.134	0.51	0.66	0.21	0.189	0.40	0.60	0.14		
2-DOF VC	0.064	0.70	0.76	0.33	0.054	0.70	0.81	0.38		
3-DOF VC	0.041	0.77	0.81	0.51	0.044	0.75	0.83	0.51		

Table 4: Comparison between different coordinate system representations of SDFNet on seen/unseen categories with 2-DOF and 3-DOF viewpoint testing data of ShapeNet. The methods are trained with ground truth depth and normals as inputs, where OC and 2-DOF VC are trained on 2-DOF viewpoint data and 3-DOF VC is trained on 3-DOF viewpoint data.

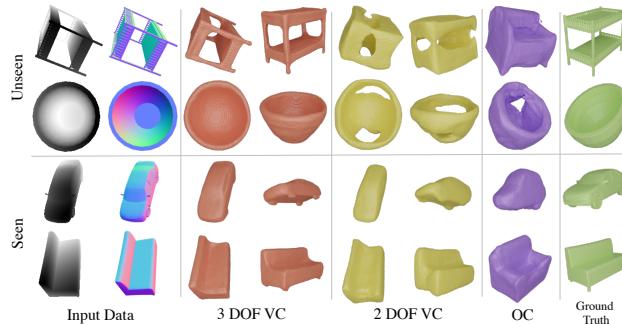


Figure 5: Visualization of output meshes from seen and unseen categories with 3-DOF viewpoint testing data of ShapeNetCore.v2.

a recognition regime and fail to generalize. In contrast, 3-DOF VC outperforms OC on unseen classes and is on-par with both methods on the seen classes. This suggests that the 3-DOF VC model learns an effective shape representation which generalizes to both the 2- and 3-DOF conditions. Comparing the seen classes across the two tables (2- and 3-DOF Testing), we note that OC and 2-DOF VC exhibit a significant drop in performance, due to their inability to generalize when camera tilt is introduced in testing. This suggests that the 2-DOF VC representations may still retain some bias towards the learned shape categories in their canonical pose. Fig. 5 illustrates qualitative results. It can be seen that 3-DOF VC outperforms 2-DOF VC and OC on capturing surface details and concavity of the shapes.

Input Data	Basic (B)				Lighting (L)			
	CD	IoU	NC	FS@1	CD	IoU	NC	FS@1
SDFNet Img	0.069	0.70	0.76	0.31	0.092	0.65	0.73	0.25
SDFNet Est	0.09	0.69	0.76	0.30	0.184	0.58	0.66	0.13
SDFNet Orcl	0.041	0.78	0.82	0.53	0.041	0.78	0.82	0.53

Input Data	L + Reflectance (LR)				LR +Background (LRBg)			
	CD	IoU	NC	FS@1	CD	IoU	NC	FS@1
SDFNet Img	0.092	0.65	0.72	0.25	0.485	0.21	0.60	0.01
SDFNet Est	0.194	0.57	0.65	0.12	0.190	0.59	0.66	0.12
SDFNet Orcl	0.041	0.78	0.82	0.53	0.041	0.78	0.82	0.53

Table 5: Generalization performance of SDFNet using images (SDFNet Img) and 2.5D predictions (SDFNet Est) to images with lighting, background and reflectance variability. All methods are trained on 3-DOF viewpoint data of ShapeNet with uniform lighting and Lambertian shading and tested on a subset of validation set of seen categories for each rendering variability. SDFNet Orcl, trained on GT 2.5D, included as reference.

4.5. Image Rendering Variability Affects Generalization

It is essential to understand the generalization ability of reconstruction algorithms with respect to changes in lighting, object surface shading,² and scene background, since models that use low-level image cues to infer object shape should ideally be robust to such changes. We perform a seen category reconstruction experiment on the 13 largest ShapeNet categories using SDFNet trained under three input regimes: image inputs only (SDFNet Img), estimated 2.5D sketch (SDFNet Est), and ground truth surface depth and normals (SDFNet Orcl). We generate images under four rendering conditions: (1) Basic with Lambertian shading, uniform, area light sources and white backgrounds (B), (2) varying lighting (L), (3) varying lighting and specular surface reflectance (LR) or (4) varying lighting, reflectance and background (LRBg) (see Fig. 6). All models are trained under the Basic setting and are then tested on novel objects from all four settings.

Our findings show the expected result that the performance of the non-oracle models degrades when tested on data with variable lighting, and exhibit only slight performance decreases when reflectance is also added (see Tbl. 5). Interestingly, the model using an intermediate 2.5D representation does not suffer a performance drop when tested on images with random backgrounds, perhaps due to the ability to use the silhouette for foreground segmentation. Results in Sec. 4.3 show good performance for models trained under all sources of variability (LRBg). Note that the rendering

²We investigate the effect of object surface reflectance properties while the surface texture remains constant

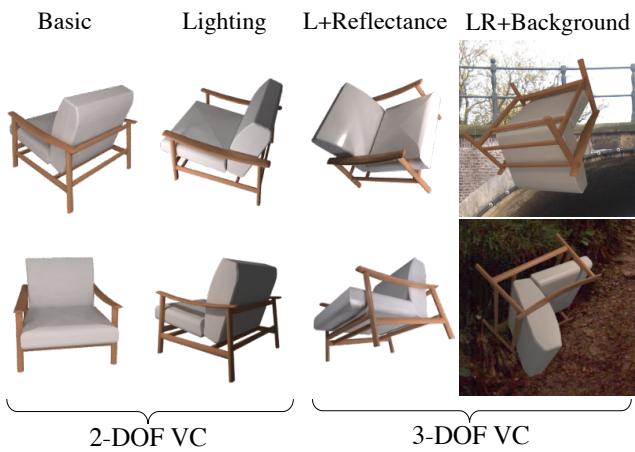


Figure 6: Rendering variability in terms of object appearance, object pose and background.

settings used in previous works are similar to our Basic setting. Given the poor generalization of models trained in this way, our findings suggest that models should be trained on data with high visual variability in order achieve effective generalization.

4.6. Cross Dataset Shape Reconstruction

In this section we further investigate general 3D shape reconstruction through experiments evaluating cross-dataset generalization. Tatarchenko et al. [38] discuss the data leakage issue, which arises when objects in the testing set are similar to objects in the training set. Zhang et al. [49] propose testing on unseen categories as a more effective test of generalization. In this section, we go beyond testing generalization on novel classes by experimenting with two inherently different datasets: ShapeNetCore.v2 and ABC, il-

Train data →	ShapeNet			ABC				
	CD	IoU	NC	FS@1	CD	IoU	NC	FS@1
Test data ↓								
Vis. ShapeNet	0.033	N/A	0.88	0.63	0.038	N/A	0.87	0.57
Occ. ShapeNet	0.058	N/A	0.82	0.38	0.062	N/A	0.81	0.41
Vis. ABC	0.643	N/A	0.73	0.54	0.026	N/A	0.89	0.67
Occ. ABC	0.658	N/A	0.66	0.34	0.044	N/A	0.82	0.55
ShapeNet	0.044	0.75	0.83	0.51	0.047	0.74	0.83	0.50
ABC	0.65	0.64	0.51	0.44	0.035	0.79	0.84	0.62

Table 6: Cross-dataset comparison of generalization to unseen classes of ShapeNet and test samples from ABC and vice versa. All models are trained on 3-DOF viewpoint data with ground truth 2.5D sketches as inputs. The first row parity reports visible (Vis.) and occluded (Occ.) surface performances of each testing dataset; the second row parity reports reconstruction performance overall.



Figure 7: Example objects from the most numerous categories in ShapeNet (top) and illustrative examples of ABC (bottom).

lustrated in Fig. 7. For this experiment we train 3-DOF VC SDFNet with ground truth 2.5D sketches as input. For this experiment’s analysis, we decompose the error into the visible and self-occluded object surface components, as shown in Tbl. 6. Our findings show that the performance of the model trained on ABC and evaluated on unseen ShapeNet categories is on par with the performance of the model trained on ShapeNet seen categories and tested on ShapeNet unseen categories for both visible and occluded surfaces. This is a surprising result since the we expect the ability to infer occluded surfaces to be biased towards the training data domain. The converse is not true, since 17% of the generated meshes when training on ShapeNet and testing ABC on are empty. This suggests that the ABC model learns a more robust shape representation that potentially comes from the fact that ABC objects are more diverse in shape. We believe we are the first to show that given access to ground truth 2.5D inputs and 3-DOF data, it is possible to generalize between significantly different shape datasets. Qualitative results are shown in Fig. 8 with further results in Supplement A.

5. Conclusion

This paper presents the first comprehensive exploration of generalization to unseen shapes in shape reconstruction, by generalizing to both unseen categories and a completely different dataset. Our solution consists of SDFNet, a novel architecture that combines a 2.5D sketch estimator with a 3D shape regressor that learns a signed distance function. Our findings imply that future approaches to single-view 3D shape reconstruction can benefit significantly from rendering with high visual variability (LRBg) and generating 3-DOF views of models (3-DOF VC representation). In ad-

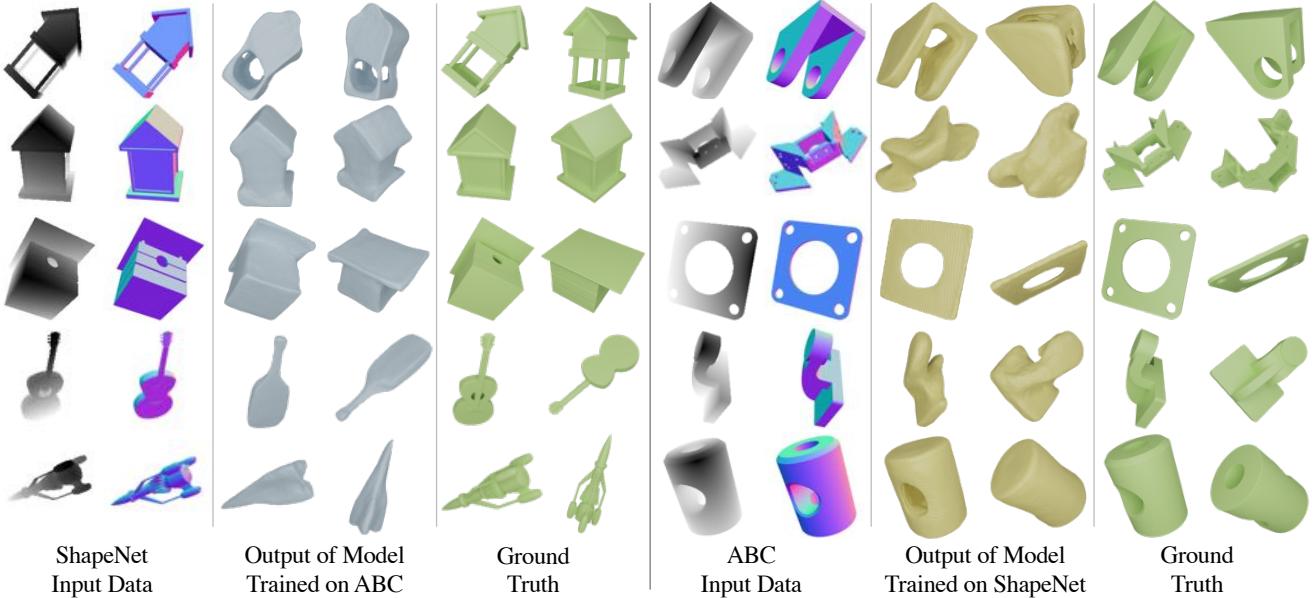


Figure 8: Cross-dataset generalization performance on unseen classes of ShapeNet of a model trained on ABC (left) and of a model trained on ShapeNet and tested on ABC (right). All models are trained on 3-DOF data with ground truth 2.5D sketches as inputs.

dition, testing on diverse shape datasets is an important future direction for effectively evaluating generalization performance.

6. Acknowledgement

We would like to thank Randal Michnovicz for his involvement in discussions and early empirical investigations. This work was supported by NSF Award 1936970 and NIH R01-MH114999.

References

- [1] Harry Barrow, J Tenenbaum, A Hanson, and E Riseman. Recovering intrinsic scene characteristics. *Comput. Vis. Syst.*, 2(3-26):2, 1978.
- [2] Anil S Baslamisli, Hoang-An Le, and Theo Gevers. Cnn based learning using reflection and retinex models for intrinsic image decomposition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6674–6683, 2018.
- [3] Blender Online Community. *Blender - a 3D modelling and rendering package*. Blender Foundation, Blender Institute, Amsterdam.
- [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.
- [5] Weifeng Chen, Zhao Fu, Dawei Yang, and Jia Deng. Single-image depth perception in the wild. In *Advances in neural information processing systems*, pages 730–738, 2016.
- [6] Zhiqin Chen and Hao Zhang. Learning implicit fields for generative shape modeling. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5939–5948, 2019.
- [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.
- [8] Harm De Vries, Florian Strub, Jérémie Mary, Hugo Larochelle, Olivier Pietquin, and Aaron C Courville. Modulating early visual processing by language. In *Advances in Neural Information Processing Systems*, pages 6594–6604, 2017.
- [9] Jose M Facil, Benjamin Ummenhofer, Huizhong Zhou, Luis Montesano, Thomas Brox, and Javier Civera. Cam-conv: camera-aware multi-scale convolutions for single-view depth. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 11826–11835, 2019.
- [10] Michael Firman, Oisin Mac Aodha, Simon Julier, and Gabriel J. Brostow. Structured prediction of unobserved voxels from a single depth image. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.
- [11] Silvio Giancola, Jesus Zarzar, and Bernard Ghanem. Leveraging shape completion for 3d siamese tracking. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1359–1368, 2019.
- [12] Thibault Groueix, Matthew Fisher, Vladimir G Kim, Bryan C Russell, and Mathieu Aubry. Atlasnet: A paper-

- m\`{a}ch\`{e} approach to learning 3d surface generation. *arXiv preprint arXiv:1802.05384*, 2018.
- [13] Paul Henderson and Vittorio Ferrari. Learning single-image 3d reconstruction by generative modelling of shape, pose and shading. *International Journal of Computer Vision*, pages 1–20, 2019.
 - [14] Jingwei Huang, Yichao Zhou, Thomas Funkhouser, and Leonidas J Guibas. Framenet: Learning local canonical frames of 3d surfaces from a single rgb image. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 8638–8647, 2019.
 - [15] Katsushi Ikeuchi and Berthold KP Horn. Numerical shape from shading and occluding boundaries. *Artificial intelligence*, 17(1-3):141–184, 1981.
 - [16] Dominic Jack, Jhony K Pontes, Sridha Sridharan, Clinton Fookes, Sareh Shirazi, Frederic Maire, and Anders Eriksson. Learning free-form deformations for 3d object reconstruction. In *Asian Conference on Computer Vision*, pages 317–333. Springer, 2018.
 - [17] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.
 - [18] Marian Kleineberg, Matthias Fey, and Frank Weichert. Adversarial generation of continuous implicit shape representations. *arXiv preprint arXiv:2002.00349*, 2020.
 - [19] Sebastian Koch, Albert Matveev, Zhongshi Jiang, Francis Williams, Alexey Artemov, Evgeny Burnaev, Marc Alexa, Denis Zorin, and Daniele Panozzo. Abc: A big cad model dataset for geometric deep learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 9601–9611, 2019.
 - [20] Yevhen Kuznetsov, Jorg Stuckler, and Bastian Leibe. Semi-supervised deep learning for monocular depth map prediction. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 6647–6655, 2017.
 - [21] Zhengqi Li and Noah Snavely. Cgintrinsics: Better intrinsic image decomposition through physically-based rendering. In *ECCV*, 2018.
 - [22] Zhengqi Li and Noah Snavely. Megadepth: Learning single-view depth prediction from internet photos. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.
 - [23] Zhengqin Li, Zexiang Xu, Ravi Ramamoorthi, Kalyan Sunkavalli, and Manmohan Chandraker. Learning to reconstruct shape and spatially-varying reflectance from a single image. *ACM Transactions on Graphics (TOG)*, 37(6):1–11, 2018.
 - [24] Shichen Liu, Tianye Li, Weikai Chen, and Hao Li. Soft rasterizer: A differentiable renderer for image-based 3d reasoning. In *The IEEE International Conference on Computer Vision (ICCV)*, October 2019.
 - [25] Shichen Liu, Shunsuke Saito, Weikai Chen, and Hao Li. Learning to infer implicit surfaces without 3d supervision. In *Advances in Neural Information Processing Systems*, pages 8293–8304, 2019.
 - [26] William E Lorensen and Harvey E Cline. Marching cubes: A high resolution 3d surface construction algorithm. *ACM siggraph computer graphics*, 21(4):163–169, 1987.
 - [27] Lars Mescheder, Michael Oechsle, Michael Niemeyer, Sebastian Nowozin, and Andreas Geiger. Occupancy networks: Learning 3d reconstruction in function space. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4460–4470, 2019.
 - [28] Jeong Joon Park, Peter Florence, Julian Straub, Richard Newcombe, and Steven Lovegrove. Deepsdf: Learning continuous signed distance functions for shape representation. *arXiv preprint arXiv:1901.05103*, 2019.
 - [29] Xiaojuan Qi, Renjie Liao, Zhengzhe Liu, Raquel Urtasun, and Jiaya Jia. Geonet: Geometric neural network for joint depth and surface normal estimation. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.
 - [30] Jiaxiong Qiu, Zhaopeng Cui, Yinda Zhang, Xingdi Zhang, Shuaicheng Liu, Bing Zeng, and Marc Pollefeys. Deeplidar: Deep surface normal guided depth prediction for outdoor scene from sparse lidar data and single color image. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.
 - [31] Jason Rock, Tanmay Gupta, Justin Thorsen, JunYoung Gwak, Daeyun Shin, and Derek Hoiem. Completing 3d object shape from one depth image. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2015.
 - [32] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer, 2015.
 - [33] Daeyun Shin, Charless C Fowlkes, and Derek Hoiem. Pixels, voxels, and views: A study of shape representations for single view 3d object shape prediction. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3061–3069, 2018.
 - [34] Edward Smith, Scott Fujimoto, Adriana Romero, and David Meger. GEOMetrics: Exploiting geometric structure for graph-encoded objects. In Kamalika Chaudhuri and Ruslan Salakhutdinov, editors, *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 5866–5876, Long Beach, California, USA, 09–15 Jun 2019. PMLR.
 - [35] Shuran Song, Fisher Yu, Andy Zeng, Angel X. Chang, Manolis Savva, and Thomas Funkhouser. Semantic scene completion from a single depth image. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.
 - [36] David Stutz and Andreas Geiger. Learning 3d shape completion under weak supervision. *International Journal of Computer Vision*, pages 1–20, 2018.
 - [37] Marshall F Tappen, William T Freeman, and Edward H Adelson. Recovering intrinsic images from a single image. In *Advances in neural information processing systems*, pages 1367–1374, 2003.
 - [38] Maxim Tatarchenko, Stephan R Richter, René Ranftl, Zhuwen Li, Vladlen Koltun, and Thomas Brox. What do single-view 3d reconstruction networks learn? In *Proceed-*

- ings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3405–3414, 2019.
- [39] Nanyang Wang, Yinda Zhang, Zhiwen Li, Yanwei Fu, Wei Liu, and Yu-Gang Jiang. Pixel2mesh: Generating 3d mesh models from single rgb images. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 52–67, 2018.
- [40] Diana Werner, Ayoub Al-Hamadi, and Philipp Werner. Truncated signed distance function: Experiments on voxel size. In Aurélio Campilho and Mohamed Kamel, editors, *Image Analysis and Recognition*, pages 357–364, Cham, 2014. Springer International Publishing.
- [41] Jiajun Wu, Yifan Wang, Tianfan Xue, Xingyuan Sun, Bill Freeman, and Josh Tenenbaum. Marrnet: 3d shape reconstruction via 2.5 d sketches. In *Advances in neural information processing systems*, pages 540–550, 2017.
- [42] Zhirong Wu, Shuran Song, Aditya Khosla, Fisher Yu, Lin-guang Zhang, Xiaoou Tang, and Jianxiong Xiao. 3d shapenets: A deep representation for volumetric shapes. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1912–1920, 2015.
- [43] Zhirong Wu, Shuran Song, Aditya Khosla, Fisher Yu, Lin-guang Zhang, Xiaoou Tang, and Jianxiong Xiao. 3d shapenets: A deep representation for volumetric shapes. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1912–1920, 2015.
- [44] Z. Zhang et al. X. Zhang. <https://github.com/xiumingzhang/GenRe-ShapeHD>.
- [45] Jianxiong Xiao, James Hays, Krista A Ehinger, Aude Oliva, and Antonio Torralba. Sun database: Large-scale scene recognition from abbey to zoo. In *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 3485–3492. IEEE, 2010.
- [46] Qiangeng Xu, Weiyue Wang, Duygu Ceylan, Radomir Mech, and Ulrich Neumann. Disn: Deep implicit surface network for high-quality single-view 3d reconstruction. 2019.
- [47] Bo Yang, Hongkai Wen, Sen Wang, Ronald Clark, Andrew Markham, and Niki Trigoni. 3d object reconstruction from a single depth view with adversarial learning. In *The IEEE International Conference on Computer Vision (ICCV) Workshops*, Oct 2017.
- [48] Jin Zeng, Yanfeng Tong, Yunmu Huang, Qiong Yan, Wenxiu Sun, Jing Chen, and Yongtian Wang. Deep surface normal estimation with hierarchical rgb-d fusion. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6146–6155, 2019.
- [49] Xiuming Zhang, Zhoutong Zhang, Chengkai Zhang, Joshua B Tenenbaum, William T Freeman, and Jiajun Wu. Learning to Reconstruct Shapes from Unseen Classes. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2018.

Appendix

A. Results for Cross Dataset Shape Reconstruction

Figure 9 contains qualitative results for SDFNet trained on ground truth 2.5D sketches of ABC and ShapeNet (seen classes) and tested on the ABC test set. The outputs suggest that the model trained on ABC has a better ability to capture shape detail and non-convexity e.g. hole in the fourth and the last row, spindle protruding from the cylinder the third row.

B. Comparison with GenRe

In this section we describe the steps we took to perform a comparison with GenRe on the complete ShapeNet-Core.v2 dataset. GenRe [49] is a three stage method consisting of depth estimation, spherical inpainting, and voxel refinement, that are trained sequentially one after the other. Each stage requires its own set of ground truth data for training, which the authors have released for the three largest classes of ShapeNet. For testing, the authors have released around 30 objects for each of the 9 unseen classes. In order to run GenRe [49] on our training and testing split of 13 seen and 42 unseen classes, we re-implemented the ground truth data generation pipeline for GenRe (referred to as GenRe GT Pipeline for the rest of this section). To generate RGB images and ground truth depth images, we adapted our Blender-based image rendering pipeline. To generate the full spherical maps, we partially adapted code from the authors’ release, in addition to writing new code to complete the procedure. To produce the voxel grids used during training, we employed code released by DISN [46] to extract 128^3 grids of signed distance fields for each mesh, which are then rotated, re-sampled and truncated to generate a voxel grid for each object view. The original GenRe voxel ground truth data is generated from inverted truncated unsigned distance (TuDF) fields [40]. In our GenRe experiments Section 4.3, we use a truncation value of $\frac{3}{128}$.

We validated our GenRe GT Pipeline by recreating the training and testing data released by the authors, and comparing the performance of GenRe trained on the released data [44] and data generated by our pipeline. Note that the GenRe authors focus on using CD for evaluation and report the best average CD after sweeping the isosurface threshold for mesh generation using marching cubes [26] prior to evaluation. In contrast, for CD we sample 100K surface points from the model output mesh and 300K surface points for the ground truth mesh, and use a fixed threshold of 0.25 to generate meshes from the model output. We used a fixed threshold to avoid biasing the performance based on the testing data. The ground truth meshes used to evaluate GenRe are obtained by running Marching Cubes on the

TuDF grids using a threshold of 0.5. The model and ground truth meshes are normalized to fit inside a unit cube prior to computing metrics, as in our other experiments.

In Tbl. 8 and Tbl. 7 we present the outcome of training and testing GenRe on the data provided by the authors, compared with training and testing GenRe on data from our GenRe GT Pipeline. Training on data from our GenRe GT Pipeline results in lower performance than originally reported, resulting in a mean 0.168 CD on unseen classes compared to the reported 0.106 in the paper (Table 1 [49]). This result is shown in the first column of Tbl. 7. This is potentially since our evaluation procedure is different from the one originally used. We do not sweep threshold values for isosurfaces, we scale the meshes to fit in a unit cube, and sample 100K+ points on the object surface in comparison to 1024 in the evaluation done in GenRe [49]. The last columns of Tbl. 8 and Tbl. 7 show comparable the testing performance for GenRe trained using our GenRe GT Pipeline and GenRe trained on the released data. The insignificant difference in performance between training GenRE on the released data and using our GenRe GT Pipeline demonstrates that our implementation is correct.

C. Further Discussion of Metrics

C.1. Issues with Current Metrics

Metrics for measuring the distance between two 3D shapes play an important role in shape reconstruction. While there are a variety of widely-used metrics, prior work [38, 33] has identified significant disadvantages with several standard metrics. IoU has been used extensively, and has the advantage of being straightforward to evaluate since it does not require correspondence between surfaces. However, while IoU is effective in capturing shape similarity at a coarse level, it is difficult to capture fine-grained shape details using IoU, since it is dominated by the interior volume of the shape rather than the surface details [38]. Figure 3 in the main text illustrates some of the issues that can arise in using shape metrics. On the right, the drum is progressively simplified from right to left. The IoU score exhibits very little change, reflecting its poor performance in capturing fine-grained details. In contrast, the F-Score (FS) at 1% of the side-length of reconstructed volume, in this case a unit side length bound cube, shows good sensitivity to the loss of fine-grained details. On the left of Fig. 3, the bowl is progressively thickened from right to left, while the shape of its surfaces remains largely constant. This example points out two issues. The first is the difficulty of interpreting IoU values for thin objects. An IoU of 0.15 would generally be thought to denote very poor agreement, while in fact the leftmost bowl is a fairly good approximation of the shape of the source bowl. In contrast, normal consistency (NC) is very sensitive to fine-grained shape details

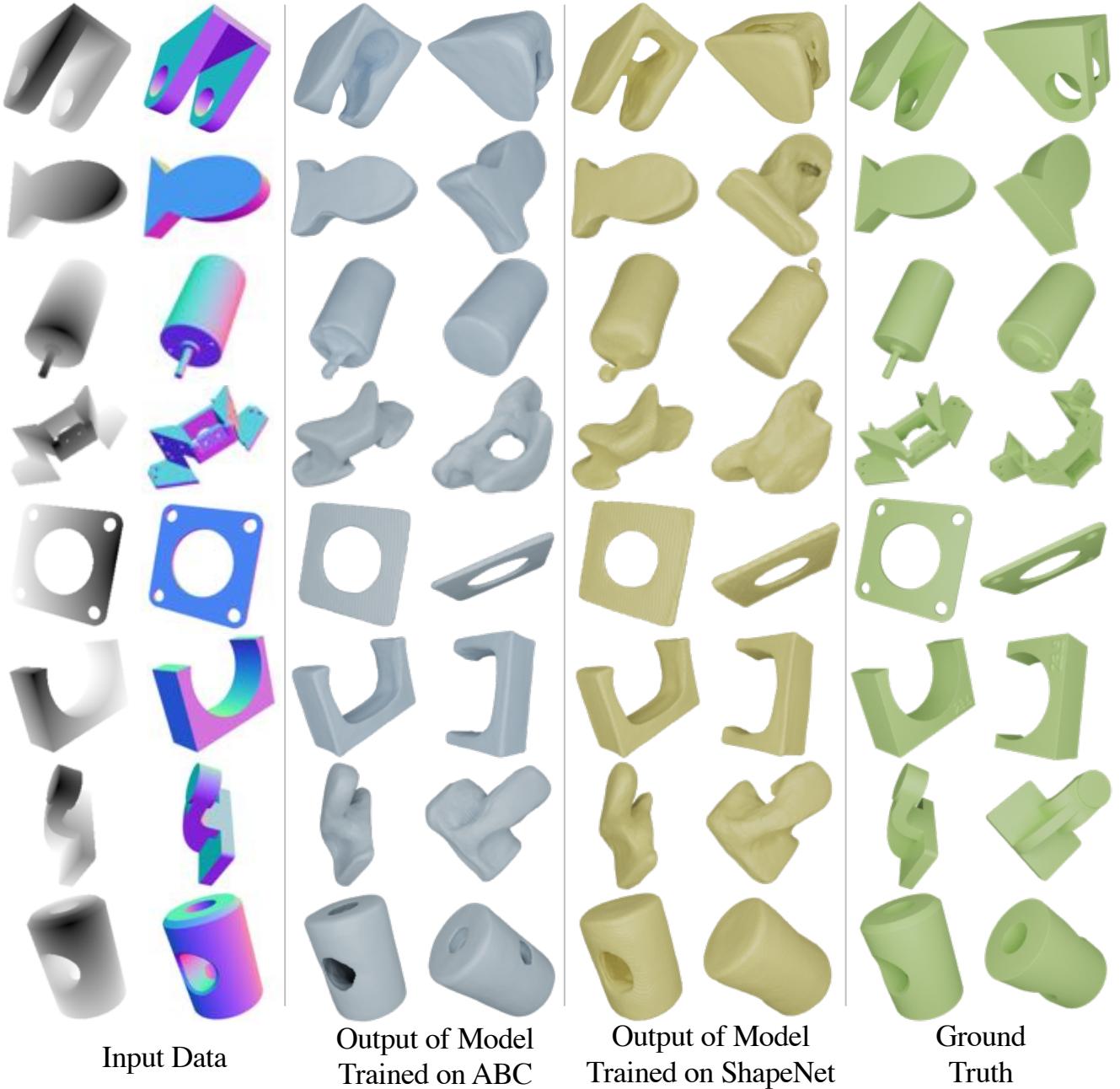


Figure 9: Performance on test samples of ABC of models trained on ABC (second column) and ShapeNet (third column) respectively. All models are trained on 3-DOF data with ground truth 2.5D sketches as inputs.

but fails to capture volumetric changes, as in this example NC exhibits almost no change despite the progressive thickening. While there is no ideal shape metric, we follow [38] in adopting FS@ d as the primary shape metric in this work.

The sampling floor issue is illustrated in Figs. 10, 11 and 12. To generate the curves in each figure, we take each object in ShapeNetCore.v2 and treat it as both the source and target object in computing the shape metrics. For example, with F-Score, we randomly sample the indicated number of

surface points twice, to obtain both source and target point clouds, and then compare the point clouds under the FS@ d shape metric for different choices of d (thresholds, along x axis). The average curves plot the average accuracy score (y axis, with error bars) for each choice of d . The minimum curves denote the minimum FS for the single worst-case mesh at each threshold. Since the source and target objects are identical meshes, the metric should always be 1, denoting perfect similarity. We can see that for 10K samples the

	CD Reported in Table 1 [49]	Our Evaluation w/o Scaling	Our Evaluation w/ Scaling	Our Implementation w/ Scaling
car	N/A	0.077	0.088	0.119
airplane	N/A	0.115	0.147	0.147
chair	N/A	0.105	0.130	0.132
avg	0.064	0.093	0.122	0.133

Table 7: Comparison of GenRe performance trained on data from our data generation pipeline with GenRe trained on the data released with the paper on seen categories. Our Evaluation w/o Scaling indicates evaluation without scaling the meshes up to fit in a unit cube, w/ Scaling indicates that meshes have been scaled up to fit inside a unit cube. The last column is trained using our GT generating pipeline and our evaluation code.

	CD Reported in Table 1 [49]	Our Evaluation w/o Scaling	Our Evaluation w/ Scaling	Our Implementation w/ Scaling
bench	0.089	0.114	0.132	0.135
display	0.092	0.108	0.222	0.216
lamp	0.124	0.177	0.225	0.196
loudspeaker	0.115	0.133	0.157	0.186
rifle	0.112	0.149	0.183	0.157
sofa	0.082	0.097	0.113	0.126
table	0.096	0.129	0.151	0.169
telephone	0.107	0.116	0.134	0.146
vessel	0.092	0.113	0.164	0.181
avg	0.106	0.137	0.165	0.168

Table 8: Comparison of GenRe performance trained on data from our data generation pipeline with GenRe trained on the data released with the paper on unseen categories. Our Evaluation w/o Scaling indicates evaluation without scaling the meshes up to fit in a unit cube, w/ Scaling indicates that meshes have been scaled up to fit inside a unit cube. The last column is trained using our GT generating pipeline and our evaluation code.

average FS@1 score is around 0.8, which is an upper bound on the ability to measure the reconstruction accuracy under this evaluation approach. A practical constraint on the use of a large number of samples is the time complexity of Nearest Neighbor matching ($O(N \log N)$). Evaluation times of around 2 hours are required for 100K points on 10K meshes on a Titan X GPU and 12 CPU threads. For 1M point samples, evaluation would take approximately 2 days, which is twice as long as the time required to train the model. Note that the value of the sampling floor will depend upon the choice of both the dataset and shape metric.

The sampling floor graphs (Figs. 10, 11, 12) can be used to select an appropriate number of points for all metrics (CD, NC and FS) and a corresponding threshold d for F-Score. For example, [38] has suggested using the F-Score metric at a threshold $\leq 2\%$ of the side length of the reconstructed volume. We can see that in order to achieve an average sampling floor of 0.9 or higher for ShapeNetCore55, 100K samples would be needed for FS@0.5 or higher, while 10K samples are acceptable for FS@1.5 or higher. Note that the significant gap between the worst and average cases in Figs. 10, 11, 12 implies that for a subset of the meshes,

the sampling error is significantly worse than in the average case. This suggests that it may be beneficial to compute and report the sampling floor along with their performance evaluation on any novel datasets in order to provide appropriate context.

C.2. Implementation of Metrics

In this work, we provide implementations for IoU, Chamfer distance (CD), normal consistency (NC) and F-score@ d (FS). For IoU we sample 3D points and compute $\text{IoU} = |O_1 \cap O_2| / |O_1 \cup O_2|$ where O_1 and O_2 are occupancy values obtained from checking whether the sampled points are inside or outside the meshes. For testing purposes, we sample points densely near the surface of the ground truth mesh with the same density as training. Note that this way of computing IoU will more strictly penalize errors made near the mesh surface. This therefore captures fine-grained shape details better than IoU computed after voxelization or using grid occupancies. IoU computed from grid occupancies requires high resolution to precisely capture thin parts of shapes which can be computationally expensive. Although we sample points densely closer to the

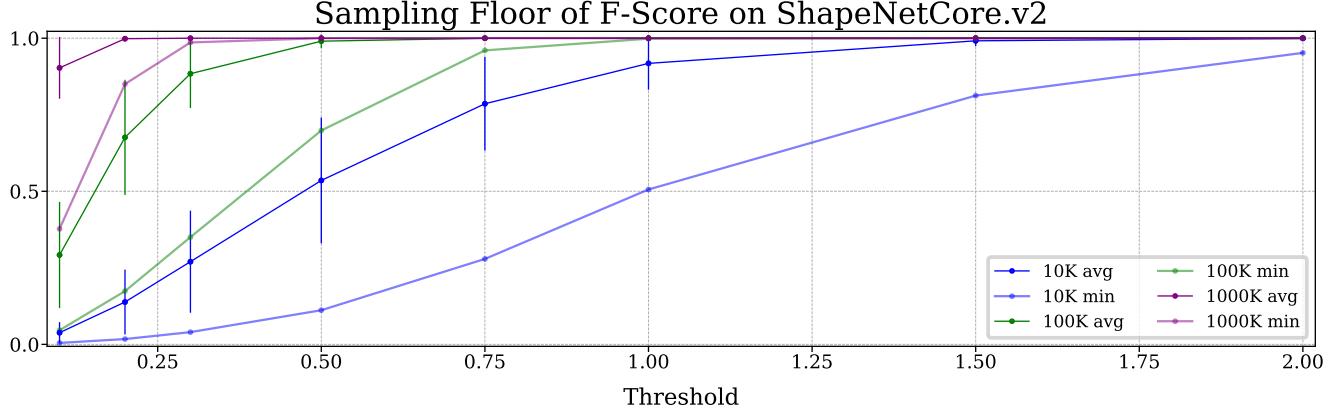


Figure 10: An illustration of the Sampling floor (maximum measurable accuracy in comparing two meshes as a function of the number of point samples) on ShapeNetCore.v2 using F-Score at thresholds $d \leq 2\%$. Curves with error bars give the average sampling floor for different numbers of samples from 10K to 1M. Curves without error bars denote the worst-case meshes. Higher is better for FS

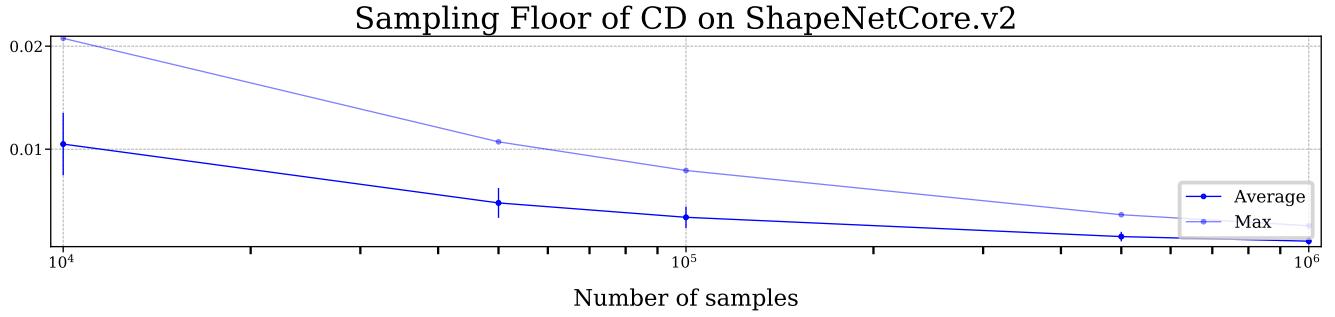


Figure 11: An illustration of the Sampling floor (maximum measurable accuracy in comparing two meshes as a function of the number of point samples) on ShapeNetCore.v2 using Chamfer Distance (CD). Curves with error bars give the average sampling floor for different numbers of samples from 10K to 1M. Curves without error bars denote the worst-case meshes. Lower is better for CD.

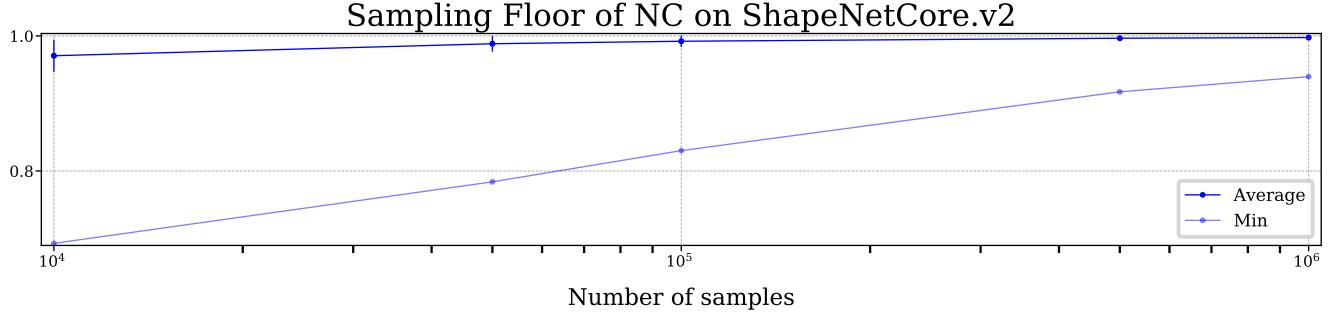


Figure 12: An illustration of the Sampling floor (maximum measurable accuracy in comparing two meshes as a function of the number of point samples) on ShapeNetCore.v2 using Normal Consistency (NC). Curves with error bars give the average sampling floor for different numbers of samples from 10K to 1M. Curves without error bars denote the worst-case meshes. Higher is better for NC.

mesh surface, to accurately measure IoU, it is important to also sample enough points uniformly in the cube volume.

For CD, NC and FS, we first sample 300K points and 100K points on the surface of predicted mesh (S_1) and

ground-truth mesh (S_2) respectively. The metrics are computed as follows

$$CD(S_1, S_2) = \frac{1}{|S_1|} \sum_{x \in S_1} \min_{y \in S_2} \|x - y\|_2 + \frac{1}{|S_2|} \sum_{y \in S_2} \min_{x \in S_1} \|x - y\|_2,$$

$$NC(S_1, S_2) = \frac{1}{2|S_1|} \sum_{x \in S_1} \min_{y \in S_2} |\langle \vec{n}_x, \vec{n}_y \rangle| + \frac{1}{2|S_2|} \sum_{y \in S_2} \min_{x \in S_1} |\langle \vec{n}_x, \vec{n}_y \rangle|,$$

where \vec{n}_x denotes the normal vector at point x , and

$$FS@d(S_1, S_2) = \frac{2 \cdot Precision@d \cdot Recall@d}{Precision@d + Recall@d},$$

where $Precision@d$ measures the portion of points from the predicted mesh that lie within a threshold d to the points from the ground truth mesh, and $Recall@d$ indicates the portion of points from the ground truth mesh that lie within a threshold d to the points from the predicted mesh.

D. Further Data Generation Details

Object Origin: Current single-view object shape reconstruction algorithms are not robust to changes in object translation, with training generally done with the object at the center of the image. This requires careful consideration of the placement of the object origin when performing rotation. Object meshes in ShapeNet have a predetermined origin. The GenRe algorithm is implemented for ShapeNet so that objects rotate around this object origin for VC training. For experiments with GenRe, we kept to this original design decision and rendered the objects after rotating them about the predetermined origin. For SDFNet, we rotate the object about the center of its bounding box. As a result of this distinction, GenRe and SDFNet are trained and tested on two distinct sets of object renderings, consisting of the same objects, with the same pose variability, rendered under the same lighting and reflectance variability settings, with the only difference being the object origin. This difference can be seen in Figure 4.

Pose Variability During 2-DOF VC and 3-DOF VC training: For 2-DOF training, we render views in the range of $[-50, 50]$ for elevation and $[0, 360]$ for azimuth. For 3-DOF VC, in order to include tilt, and achieve high variability in object pose, we initially apply a random pose to the object, and then generate 25 views using the same procedure and parameters as for the 2-DOF case.

Camera Parameters For all generated data, the camera distance is 2.2 from the the origin, where the object is placed. The focal length of the camera is 50mm with a 32mm sensor size. All images are rendered with a 1:1 aspect ratio and at a resolution of 256×256 pixels.

Background While it is possible is to use image based lighting techniques such as environment mapping to generate variability in backgrounds and lighting that is more realistic, this approach significantly slows down the ray-tracing based rendering process and requires environment map images. In order to generate large amounts of variable data we

use random backgrounds from the SUN [45] scenes dataset instead.