On the generalization of learning-based 3D reconstruction

Miguel Angel Bautista
Apple

Walter Talbott
Apple

Shuangfei Zhai
Apple

Nitish Srivastava
Apple

Joshua M. Susskind
Apple

{mbautistamartin,wtalbott,szhai,nitish_srivastava,jsusskind}@apple.com

Abstract

State-of-the-art learning-based monocular 3D reconstruction methods learn priors over object categories on the training set, and as a result struggle to achieve reasonable generalization to object categories unseen during training. In this paper we study the inductive biases encoded in the model architecture that impact the generalization of learning-based 3D reconstruction methods. We find that 3 inductive biases impact performance: the spatial extent of the encoder, the use of the underlying geometry of the scene to describe point features, and the mechanism to aggregate information from multiple views. Additionally, we propose mechanisms to enforce those inductive biases: a point representation that is aware of camera position, and a variance cost to aggregate information across views. Our model achieves state-of-the-art results on the standard ShapeNet 3D reconstruction benchmark in various settings.
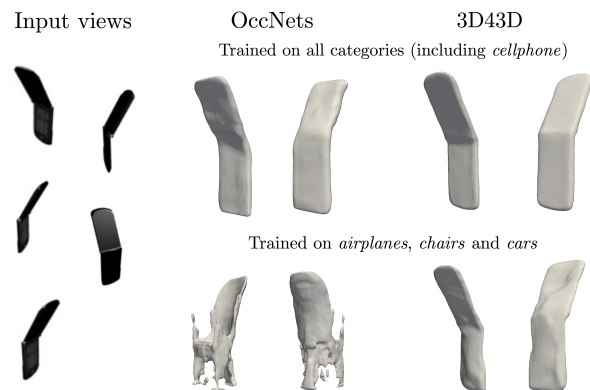
Figure 1. An example of reconstructing object categories unseen during training. State-of-the-art methods for learning-based reconstruction like OccNets [19] fail to generalize to categories unseen during training, mapping objects to their closest category in the training set (*e.g.* a chair). 3D43D improves generalization by using 3 inductive biases in the network design.

1. Introduction

Reconstructing the 3D shape of an object from monocular input views is a fundamental problem in computer vision. When the number of input views is small, reconstruction methods rely on priors over object shapes. Learning-based algorithms encode such priors from data. Recently proposed approaches [8, 34, 19, **?**, **?**, **?**, **?**] have achieved success in the single/multi view, seen category case when generalizing to novel objects within the seen categories. However, these approaches have difficulty generalizing to object categories not seen during training (cf. Fig. 1).

We present progress learning priors that generalize to unseen categories by incorporating a geometry-aware spatial feature map. Within this paradigm, we propose a point representation aware of camera position, and a variance cost to aggregate information across views.

A typical learning-based approach will take a single 2D view of an object as input, and a model to generate a 3D reconstruction. What should happen to the 3D ground truth as the viewpoint of the 2D input changes? An *object-centric* coordinate system would keep the ground truth fixed to a canonical coordinate system, regardless of the viewpoint of the 2D input view. In contrast, a *view-centric* coordinate system would rotate the ground truth coordinate frame in conjunction with the input view. An example of the two different coordinate systems [1] is shown in Fig. 2. Object-centric coordinate systems align shapes of the same category to an arbitrary, shared coordinate system. This introduces stable spatial relationships during training (*e.g.*, wheels of different car shapes generally occupy the same absolute area of $\mathbb{R}^3$). This makes the reconstruction task easier to learn, but these relationships are not necessarily

---

[1]In the graphics community the object-centric coordinate system is often referred as *world coordinates* and the view-centric coordinate system as *camera coordinates*.
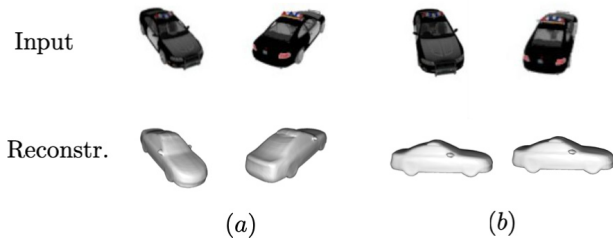
Input

Reconstr.

(a)   (b)

Figure 2. (a) View-centric coordinate system, where ground truth 3D objects are aligned to their respective input views. (b) Object-centric coordinate system, where all input views share the same ground-truth canonical 3D object orientation.

shared across categories. Similar to [26, 30], we show empirically that adopting a view-centric coordinate system improves generalization to unseen categories.

Another critical factor for achieving good generalization to unseen categories is the capacity of a model to encode geometrical features when processing the input view. Similar to [34], our model uses feature maps with spatial extent, rather than pooling across spatial locations to obtain a global representation. In [34] the motivation for using spatial feature maps is to preserve fine grained details of geometry (*e.g.* to better model categories in the training set). In contrast, in this paper we analyze generalization to unseen categories and how different encoding designs impact generalization capability. We argue that using a globally pooled representation encourages the model to perform reconstruction at the object level (since the pooling operation is invariant to spatial arrangements of the feature map), which makes it difficult to generalize to objects from unseen categories. By keeping the spatial extent of features, on the other hand, we process and represent an object at the part level. Critically, in contrast to [34], we model the scene geometry across different views by explicitly embedding information about camera poses in the spatial feature maps. We show empirically that using these geometry-aware spatial feature maps increases generalization performance.

Finally, we use multi-view aggregation to improve generalization performance. Traditional approaches to 3D reconstruction, such as multi-view stereo (MVS) [25] or structure-from-motion (SfM) [32], exploit the geometry of multiple views via cost volumes instead of priors learned from data. These approaches fail in single-view cases. Single-view reconstruction models, though, must rely entirely on priors for occluded regions. We propose a model that combines learned priors with the complimentary information gained from multiple views. We aggregate information from multiple views by taking inspiration from cost volumes used in MVS and compute a variance cost across views. By refining its single-view estimates with additional

views, our model shows improved generalization performance.

Individually, these factors are important as backed by literature and our empirical results and addressing them leads to compounding effects on generalization. The view-centric coordinate system has been shown to improve generalization [26, 30]. However, the need to aggregate information from multiple views is also paramount to reconstruct categories not seen during training time, since the prior learned over the training categories is not trustworthy in this unseen category case. In order to maximally benefit from aggregating information from multiple views, we require features that encode information about parts of objects rather than encoding the object as a whole entity without preserving spatial information. Otherwise, aggregation can be counterproductive by reinforcing the wrong object prior. We show empirically that by compounding these three factors, 3D43D outperforms state-of-the-art 3D reconstruction algorithms when tested on both categories seen and unseen during training. Contrary to suggestions from previous work [36, 30], we achieve these gains in generalization without a drop in performance when testing on categories seen during training.

2. Related Work

Object Coordinate System. Careful and extensive experimental analysis in [26, 30] has revealed that the object shape priors learned by most 3D reconstruction approaches act in a categorization regime rather than in the expected 3D dense reconstruction regime. In other words, reconstructing the 3D object in these models happens by categorizing the input into a blend of the known 3D objects in the training set. This phenomenon has been mainly explained by the choice of the object-centric coordinate system.

Results by [26, 30] showed that object-centric coordinate systems perform better for reconstructing viewpoints and categories seen during training, at the cost of significantly hampering the capability to generalize to categories or viewpoints unseen at training time. The converse result was also observed for view-centric coordinate systems, which generalized better to unseen objects and viewpoints at the cost of degraded reconstruction performance for known categories/viewpoints.

Feature Representation. Single-view 3D reconstruction approaches have recently obtained impressive results [19, 24, 34, 31, 8, 18, 13] despite the ill-posed nature of the problem. In order for these approaches to perform single-view reconstruction successfully, priors are required to resolve ambiguities. Not surprisingly, recent works show that using local features that retain spatial information [24, 34] improve reconstruction accuracy. However, none of these approaches analyze their performance on object categories unseen during training time. A recent exception is [36]

	MV Consist. [31]	Diff. PCs. [10]	L-MVS [14]	PiFU [24]	DISN [34]	OccNet [19]	3D43D
Geometry	Voxel	Point cloud	Voxel/Depth	Func.	Func.	Func.	Func.
Coordinate Sys.	Viewer	Viewer	Object	Object	Object	Object	Viewer
Features	Global	Global	Spatial+Global	Spatial+Global	Spatial+Global	Global	Spatial+Geometric
Multi-view	No	No	Yes	Yes	Yes	No	Yes
Generalization	No	No	Yes	Yes	Yes	No	Yes

Table 1. Summary of design choices of different approaches. We describe each method in terms of their choice of: geometry representation (**Geometry**), coordinate system (**Coordinate Sys.**), feature representation (**Features**), capacity to use multiple views (**Multi-view**) and if they analyze generalization to unseen categories (**Generalization**).

where the authors propose a *non* fully differentiable approach for single view reconstruction that relies on depth in-painting of spherical maps [6]. [36] also differs from 3D43D because 3D voxel grids are used as ground-truth, and extra supervision at the level of depth maps is available at training time.

View Aggregation. Multi-view 3D reconstruction has been traditionally addressed by stereopsis approaches like MVS [25, 16] or SfM [32]. Modern learning-based approaches to MVS [35, 16] have incorporated powerful convolutional networks to the MVS pipeline. These networks focus on visible regions and do not make inferences about the geometry of occluded object parts.

Another interesting trend has been to exploit the multi-view consistency inductive bias from MVS to learn 3D shape and pose from pairs of images [31, 10, 29]. However, these approaches either predict a very sparse set of keypoints [29], a sparse point cloud [10], or a voxel grid [31], limiting the approaches to fixed resolution representations.

Conceptually close to our approach is [14]. The authors propose differentiable proxies to the operations in the standard MVS pipeline, allowing end-to-end optimization. Although [14] addresses multi-view aggregation, there are critical design choices in other aspects of the method that limit the performance. First, the geometry representation produced is a voxel grid, making the estimation of high resolution geometry unfeasible. Second, the cost-volume optimization happens via a large 3D auto-encoder which has a non trivial geometric interpretation. Third, view aggregation is performed in a recurrent fashion, making the model sensitive to permutation of the views.

Properly extending the previously discussed single-view works [19, 24, 34, 31, 8, 18, 13] to the multi-view case is not trivial, although simple extensions to aggregate multiple views are briefly outlined in [24, 34]. Inspired by cost volume computation used in MVS [35] we aggregate information from different views by computing a variance cost (Sect. 3.3.)

Geometry Representation. The choice of representation scheme for 3D objects has been at the core of 3D reconstruction research from the early beginning. Voxels [31, 14] have been used as a natural extension of 2D image representation, showing great results in low resolution regimes. However, memory and computation require-

ments to scale voxel representations to higher resolution prevent them from being widely used. Circumventing this problem, point clouds are a more frequently used representation [10, 29]. Point clouds deal with the computational cost problem of voxels by sparsifying the representation and eliminating the neighbouring structure information. Meshes [13, 15, 18] add the neighboring structure back into point cloud representations. However, to make mesh estimation efficient, neighbouring structure has to be predefined (usually in the form of connectivity of a convex polyhedron with a variable number of faces) and only deformations of that structure can be modelled. Finally, functional/implicit representations have recently gained interest [19, 34, 22, 20, 4, 12]. This representation encodes geometry as the level set of a function that can be evaluated for any point in $\mathbb{R}^3$. Such a function can generate geometry at arbitrary resolutions by evaluating as many points as desired. As a summary, Tab. 1 shows the contributions of the most relevant and related literature in comparison to 3D43D.

3. Model

We now describe our approach in terms of the choice of geometry representation, the use of geometry-aware feature maps, and the multi-view aggregation strategy. Our model is shown in Fig. 3.

3.1. Functional Geometry Representation

3D43D takes the form of a functional estimator of geometry [19, 34, 8, 22]. Given a view of an object, our goal is to predict the object occupancy function indicating whether a given point $\mathbf{p} \in \mathbb{R}^3$ lies inside the mesh of the object. In order to do so, we learn a parametric scalar field $f_\theta : \mathbb{R}^3 \times \mathbf{V} \longrightarrow [0, 1]$ where $\mathbf{V} \in \mathbb{R}^{H \times W \times 3}$ is an monocular (RGB) view of the object. In the remainder of the text the parameter subscript θ is dropped for ease of notation. This scalar field f is implemented by a fully connected deep neural network with residual blocks [2].

3.2. Encoding Geometry-aware Features

Our goal is to learn a prior over occupancy functions that generalizes well to unseen categories, which we address by giving our model the capacity to reason about local

---

[2]Details of the implementation can be found in the supplementary material

object cues and scene geometry. In order to do so, we process input views with a convolutional U-Net[23] encoder with skip connections (refer to the supplementary material for implementation details). This results in a feature map $\mathbf{C} \in \mathbb{R}^{H \times W \times C}$ for a given RGB view. This is in contrast to the approach taken in [19, 8] where a view is represented by a global feature which pools across the spatial dimensions of the feature map. Our hypothesis is that preserving the spatial dimensions in the latent representation is crucial to retain local object cues which greatly improve generalization to unseen views and categories, as demonstrated in the experiments. To predict the occupancy value for a 3D point $\mathbf{p}$ in world coordinates we project this point into its location (u, v) in $\mathbf{C}$ by using the extrinsic camera parameters $\mathbf{T} = [\mathbf{R}|\mathbf{t}] \in \mathbb{R}^{3 \times 4}$ and intrinsic parameters $\mathbf{K} \in \mathbb{R}^{3 \times 3}$ [3] (cf. Eq. (1)), and sample the corresponding feature vector $\mathbf{c} \in \mathbb{R}^{C}$. We use bi-linear sampling as in [11] to make the sampling operation differentiable.

$$(u', v', w') = \mathbf{KTp}, \ \ u = u'/w', \ \ v = v'/w'. \quad (1)$$

The feature vector $\mathbf{c}$ encodes localized appearance information but lacks any notion of the structural geometry of the underlying scene. This causes ambiguities in the representation since, for example, all points in a ray shot from the camera origin get projected to the same location in the image plane. Thus, the sampled feature vector $\mathbf{c}$ cannot uniquely characterize the points in the ray (*e.g.*, to predict occupancy). Recent works [24, 34] mitigate this issue by augmenting $\mathbf{c}$ to explicitly encode coordinates of 3D points $\mathbf{p}$. This is often done by concatenating $\mathbf{p}$ (or a latent representation of $\mathbf{p}$ [34]) and $\mathbf{c}$, and further processing it via additional fully connected blocks.

However, $\mathbf{p}$ (or its representation) is sensitive to the choice of coordinate system [26]. Recent approaches [19, 34] use a canonical object-centric coordinate system for each object category, which has been shown to generalize poorly to categories not seen during training [30]. On the other hand, expressing $\mathbf{p}$ in a view-centric coordinate system[4] improves generalization to unseen categories [26, 30, 36]. Note that if $\mathbf{p}$ is expressed in the view-centric coordinate system the characterization of the scene is incomplete since it lacks information about the points where the rays passing through $\mathbf{p}$ originated in the image capturing process (*e.g.* the representation is not aware of the origin of the view-centric coordinate system w.r.t. the scene).

To tackle this issue we represent $\mathbf{p}$ using the camera coordinate system (denoted as $\mathbf{p}'$), and give the representation access to the origin of the camera coordinate system $\mathbf{t} \in \mathbb{R}^3$ with respect to the world (*e.g.* the camera position with respect to the world coordinate system). Therefore, after sam-

pling $\mathbf{c}$ we concatenate it with $\mathbf{p}'$ and $\mathbf{t}$, and process it with an MLP $g_\theta : \mathbb{R}^n \to \mathbb{R}^n$ with residual blocks, resulting in feature representation $\mathbf{g}$ that is aware of the scene geometry $\mathbf{g} = g_\theta([\mathbf{c}, \mathbf{p}', \mathbf{t}])$. This feature representation $\mathbf{g}$ is then input to the occupancy field f. Note that this does not require additional camera information compared to [34, 24] since the camera position is already used to project $\mathbf{p}$ into the image plane to sample the feature map. In our model we explicitly condition the representation using the camera position instead of only implicitly using the camera position to sample feature maps. Fig. 3 shows our model.

3.3. Multi-View Aggregation

We now turn to the task of aggregating information from multiple views to estimate occupancy. Traditionally, view aggregation approaches for geometry estimation require the explicit computation of a 3D cost volume which is either refined using graph optimization techniques in traditional MVS approaches [25] or processed by 3D convolutional auto-encoders with a large number of parameters in learned models [14, 35, 16]. Here we do not explicitly construct a cost volume for the whole scene, instead, we compute pointwise estimates of that cost volume. One key observation is that our model is able to estimate geometry for parts of the object that are occluded in the input views, as opposed to MVS approaches that only predict geometry for visible parts of a scene (*e.g.* depth). As a result our approach integrates *reconstruction of visible geometry* and *generation of unseen geometry* under the same framework.

Our task is to predict the ground truth occupancy values $\mathbf{o} \in \{0, 1\}^p$ of points $\mathcal{P} = \{\mathbf{p}_j\}_{j=0}^p$, given a set of posed RGB views $\{\mathbf{V}_i, \mathbf{T}_i\}_{i=0}^n$. In order to do so, we independently compute geometry aware representations $\mathbf{g}_i$ across views for each point $\mathbf{p}_j$ as show in Sect. 3.2. In order for our model to deal with a variable number of views a pooling operation over $\mathbf{g}_i$ is required. Modern approaches to estimate the complete geometry of a scene (visible and occluded) from multiple views rely on element-wise pooling operators like mean or max [34, 24]. These element-wise operators can be suitable for categories seen at training time. However, in order to better generalize to unseen categories it is oftentimes beneficial to rely on the alignment cost between features as done in purely geometric (*e.g.* non learning-based) approaches [25]. Inspired by traditional geometric approaches we propose to use an approximation to the alignment cost between local features $\{\mathbf{g}_i\}_{i=0}^n$. We approximate the alignment cost on the set of local features $\{\mathbf{g}_i\}_{i=0}^n$ by computing the variance $\hat{\mathbf{g}}$ as follows,

$$\hat{\mathbf{g}} = \mathcal{V}(\{\mathbf{g}_i\}_{i=0}^n) = \frac{\sum\limits_{i=0}^n \|\bar{\mathbf{g}} - \mathbf{g}_i\|_2}{n}, \quad (2)$$

where $\bar{\mathbf{g}}$ is the average of $\{\mathbf{g}_i\}_{i=0}^n$. A key design choice

---

[3]We assume camera intrinsics to be constant.
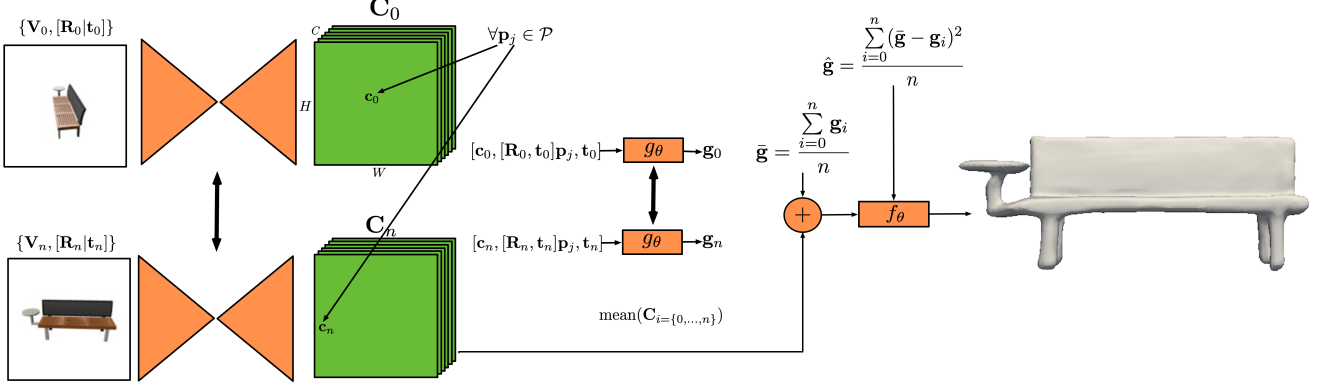[4]Also known as camera coordinate system

Figure 3. Overview of our model. Input views $\mathbf{V}_i$ are processed by our UNet encoder producing feature maps $\mathbf{C}_i$ that are sampled at spatial locations corresponding to a 3D point $\mathbf{p}_j$. Those features are then concatenated with the point $\mathbf{p}_j$ and the location of the camera origin of the corresponding input view and process through an MLP g_θ that produces geometry-aware point representations (one for each view). Those representations are used to compute a mean and variance cost across views that is used by another MLP f_θ to predict occupancy.

is that we do not use the variance $\hat{\mathbf{g}}$ as the sole input to our functional occupancy decoder f since the variance will be zero everywhere and uninformative when only a single view is available. Instead, we add a conditional input branch to our decoder f, which takes as input $\bar{\mathbf{g}}$ conditioned on $\hat{\mathbf{g}}$. We also give the model access to a global object representation by introducing a residual path that performs additive conditioning on $\bar{\mathbf{g}}$. We perform average pooling on feature maps $\mathbf{C}_i$ both spatially and across views to obtain $\bar{\mathbf{c}}$ that is added to $\bar{\mathbf{g}}$. Conditioning in f is implemented via conditional batch normalization[5] layers [3]. This formulation naturally handles the single view case, where $\hat{\mathbf{g}} = 0$. Finally, our objective function is shown in Eq. 3.

$$L(\{\mathbf{V}_i, \mathbf{T}_i\}_{i=0}^n, \mathcal{P}, \mathbf{o}) = -\sum_{j=1}^p o_j \log f(\bar{\mathbf{g}} + \bar{\mathbf{c}}, \hat{\mathbf{g}}) \quad (3)$$

4. Experiments

We present empirical results that show how 3D43D performs in two experimental setups: reconstruction of categories seen during training and generalization to categories unseen during training. In the first setup our goal is to show that 3D43D is competitive with state-of-the-art 3D reconstruction approaches. In the second setup we show that 3D43D generalizes better to categories unseen at training time. Finally, we conduct ablation experiments to show how the proposed contributions impact the reconstruction accuracy.

4.1. Settings

Dataset: For all of our experiments, we use the ShapeNet [1] subset of Choy et al. [2], together with their renderings. For a fair comparison with different methods we

use the same train/test splits and occupancy ground truth as [19], which provides an in depth comparison with several approaches [6].

Metrics: We report the following metrics, following [19]: volumetric IoU, Chamfer-L1 distance, and normal consistency score. In addition, as recently suggested by [30] we report the F-score. Volumetric IoU is defined as the intersection over union of the volume of two meshes. An estimate of the volumetric IoU is computed by randomly sampling 100k points and determining whether points reside in the meshes [19]. The Chamfer-L_1 is a relaxation of the symmetric Hausdorff distance measuring the average of reconstruction accuracy and completeness. The normal consistency score is the mean absolute dot product of the normals in one mesh and the normals at the corresponding nearest neighbors in the other mesh [19]. Finally, the F-score can be understood as the percentage of correctly reconstructed surface [30].

Implementation: We resize our input images to $(224, 224)$ pixels. For our encoder, we choose a U-Net with a ResNet-50[9] encoder, the final feature maps $\mathbf{C}$ have 256 channels and are of the same spatial size as the input. The function g_θ that computes geometric features is an MLP with 3 ResNet blocks and that takes a vector of $262 = 256 + 3 + 3$ dimensions and outputs a 256-dimensional representation $\mathbf{g}$. Our occupancy function f is an MLP with 5 ResNet blocks where all layers have 256 hidden units except the output layer. To train the occupancy function we sample 2048 points with their respective occupancy value from a pool of 100k points and use 4 input views [7]. Details of different sampling strategies can be found in [19]. We train our network with batches of 128

---

[5]Implementation details in the supplementary material

---

[6]Readers interested in the ground-truth generation process are referred to [19, 28].

[7]This was due to memory limitations. Nonetheless, the method generalizes to an arbitrary number of input views during inference

samples and use Adam[17] with default Pytorch parameters as our optimizer. We use a learning rate of 10^{-4} and train our network for 2000 epochs. To obtain meshes at inference time from the occupancy function we follow the process in [19].

4.2. Categories seen during training

In this section, we compare our method to various baselines on single-view and multi-view 3D reconstruction experiments. For the single-view setup we report our results on standard experiments for reconstructing unseen objects of categories seen during training (cf. Tab. 2). We compare 3D43D with 3D-R2N2[2], Pix2Mesh[33], PSGN[7], AtlasNet[8] and OccNets[19]. Encouragingly, 3D43D performs on par with the state-of-the-art OccNets[19] approach, which uses global features that are able to encode semantics of the objects in the training set. In addition, OccNets make use of an object-centric coordinate system, which aligns all shapes to the same canonical orientation, making the reconstruction problem simpler. Our results indicate that spatial feature maps are able to encode useful information for reconstruction despite being spatially localized. This result backs up our initial hypothesis and is critical to establish a good baseline performance. From this baseline, we explore the performance of our model when generalizing to unseen categories in different scenarios (Sect. 4.3).

In order to further validate our contribution, we provide results on multi-view reconstruction. We randomly sample 5 views of the object and compare our method with OccNets[19] (the top performer for single-view reconstruction method). To provide a fair comparison, we extend the trained model provided by OccNets (the best runner up) to the multiple view case by average pooling their conditional features across views at inference time. Since our method uses spatial features that are aware of scene geometry, we expect our aggregation mechanism to obtain more accurate reconstruction. Results shown in Tab. 3 and qualitative results in Fig. 4 consistently agree with this observation.

4.3. Generalization to unseen categories

We now turn to our second experimental setup were we evaluate the ability of 3D43D to generalize to categories not seen during training. In order to do so, we restrict the training set to the top-3 most frequent categories in ShapeNet (e.g. Car, Chair and Airplane) following [36], and test on the remaining categories. Tab. 4 compares the performance of 3D43D with two strong baselines: OccNets [19] and OccNets trained with a view-centric coordinate system ([19]-v). We extend OccNets to use view-centric coordinates in order to validate observations in recent papers [26, 30] reporting that using a view-centric coordinate system improves generalization to unseen categories. We find empirically

that this observation holds for models that do not aggregate information from multiple views. As discussed in Sec. 3.3, [19]-v suffers from systematic drawbacks due to the use of global features, and this results in degraded performance. Additionally, using a view-centric coordinate system only partially tackles the generalization problem, and further improvements can be obtained from the geometry aware features, and the mean and variance aggregation used by 3D43D.

We show sample reconstructions from this experiment in Fig. 5. The visualizations reveal that OccNets tend to work in a categorization regime, often mapping unseen categories to their closest counterparts in the training set. This is clearly visible in Fig. 5. This problem is not solved solely by using multiple views, which can be counterproductive by giving OccNets more confidence to reconstruct the wrong object.

4.4. Ablation

We perform an ablation study to show how the main design choices of our approach affect performance (ie. a point representation that is aware of camera position and a variance cost to aggregate information across views). We take as our baseline a model equivalent to DISN [34] but with a view-centric coordinate frame. We have already shown that spatial feature maps provide substantial improvements over 1D features (e.g. improvements over OccNet [19] shown in Tab. (2)(3)) for seen categories. Note that DISN [34] also reports similar results. However, in this paper we focus on analyzing the generalization of the model to categories unseen during training and show in our ablation that spatial feature maps are not the only critical design choice and our novel contributions improve reconstruction accuracy for unseen categories.

For all our model ablations our encoder (ie. a UNet with a ResNet50 encoder) outputs spatial feature maps for each view that are sampled at locations corresponding to a particular point $\mathbf{p} \in \mathbb{R}^3$ for which occupancy is predicted. Our ablation is divided in three models: *(i)* **Point model (P)**: Here we take the sampled features across views (*e.g.* the $\mathbf{c}_i$) and concatenate them with $\mathbf{p}$ before feeding through our MLP g_θ, so that $\mathbf{g}_i = g_\theta(\mathbf{c}_i, \mathbf{p})$. We take these feature representations $\mathbf{g}_i$ across views and aggregate them using average pooling, where the resulting vector is used as input to f_θ. *(ii)* **Point+Camera model (P+C)**: In this version we concatenate also the camera location $\mathbf{t}_i$ before processing the vector with g_θ, so that $\mathbf{g}_i = g_\theta(\mathbf{c}_i, \mathbf{p}, \mathbf{t}_i)$. We then average pool the resulting features and use them as input to f_θ. *(iii)* **Point+Camera+Variance model (P+C+V)**: In this model we take the same encoding as in P+C ($\mathbf{g}_i = g_\theta(\mathbf{c}_i, \mathbf{p}, \mathbf{t}_i)$). However, we now compute the mean and variance of $\mathbf{g}_i$ and use them as input and conditioning, respectively, for f_θ. This is our full 3D43D model. These models are trained

Seen category	IoU				↓Chamfer-L_1						Normal Consistency				
	[2]	[33]	[19]	3D43D	[2]	[7]	[33]	[8]	[19]	3D43D	[2]	[33]	[8]	[19]	3D43D
airplane	0.426	0.420	**0.591**	0.571	0.227	0.137	0.187	0.104	0.134	**0.096**	0.629	0.759	0.836	**0.845**	0.825
bench	0.373	0.323	0.492	**0.502**	0.194	0.181	0.201	0.138	0.150	**0.112**	0.678	0.732	0.779	**0.814**	0.809
cabinet	0.667	0.664	0.750	**0.761**	0.217	0.215	0.196	0.175	0.153	**0.119**	0.782	0.834	0.850	0.884	**0.886**
car	0.661	0.552	**0.746**	0.741	0.213	0.169	0.180	0.141	0.149	**0.122**	0.714	0.756	0.836	**0.852**	0.844
chair	0.439	0.396	0.530	**0.564**	0.270	0.247	0.265	0.209	0.206	**0.193**	0.663	0.746	0.791	0.829	**0.832**
display	0.440	0.490	0.518	0.548	**0.605**	0.284	0.239	0.198	0.258	**0.166**	0.720	0.830	0.858	0.857	**0.883**
lamp	0.281	0.323	0.400	**0.453**	0.778	0.314	0.308	**0.305**	0.368	0.561	0.560	0.666	0.694	0.751	**0.766**
loudspeaker	0.611	0.599	0.677	**0.729**	0.318	0.316	0.285	0.245	0.266	**0.229**	0.711	0.782	0.825	0.848	**0.868**
rifle	0.375	0.402	0.480	**0.529**	0.183	0.134	0.164	**0.115**	0.143	0.248	0.670	0.718	0.725	0.783	**0.798**
sofa	0.626	0.613	0.693	**0.718**	0.229	0.224	0.212	0.177	0.181	**0.125**	0.731	0.820	0.840	0.867	**0.875**
table	0.420	0.395	0.542	**0.574**	0.239	0.222	0.218	0.190	0.182	**0.146**	0.732	0.784	0.832	0.860	**0.864**
telephone	0.611	0.661	**0.746**	0.740	0.195	0.161	0.149	0.128	0.127	**0.107**	0.817	0.907	0.923	**0.939**	0.935
vessel	0.482	0.397	0.547	**0.588**	0.238	0.188	0.212	**0.151**	0.201	0.175	0.629	0.699	0.756	0.797	**0.799**
mean	0.493	0.480	0.593	**0.621**	0.278	0.215	0.216	**0.175**	0.194	0.184	0.695	0.772	0.810	0.840	**0.845**

Table 2. Performance of different approaches on the test set of categories seen during training, trained with single views. Our results show that 3D43D is comparable with state-of-the-art models trained on a object-centric coordinate system in the single view setting. Compared models are: 3D-R2N2[2], Pix2Mesh[33], PSGN[7], AtlasNet[8] and OccNets[19].

Seen category	IoU		↓ Chamfer-L_1		Normal Consistency		F-score	
	[19]	3D43D	[19]	3D43D	[19]	3D43D	[19]	3D43D
airplane	0.600	**0.736**	0.096	**0.021**	0.853	**0.899**	0.735	**0.841**
bench	0.547	**0.663**	0.176	**0.027**	0.834	**0.881**	0.691	**0.789**
cabinet	0.770	**0.831**	0.125	**0.073**	0.893	**0.925**	0.853	**0.898**
car	0.759	**0.797**	0.109	**0.090**	0.861	**0.873**	0.852	**0.878**
chair	0.568	**0.716**	0.187	**0.063**	0.846	**0.911**	0.704	**0.824**
display	0.593	**0.752**	0.168	**0.089**	0.884	**0.935**	0.723	**0.851**
lamp	0.415	**0.625**	1.083	**0.256**	0.764	**0.858**	0.546	**0.752**
loudspeaker	0.699	**0.807**	0.360	**0.143**	0.856	**0.912**	0.801	**0.883**
rifle	0.466	**0.745**	0.112	**0.012**	0.789	**0.903**	0.625	**0.851**
sofa	0.731	**0.809**	0.171	**0.054**	0.886	**0.927**	0.831	**0.886**
table	0.569	**0.689**	0.588	**0.058**	0.873	**0.921**	0.703	**0.805**
telephone	0.785	**0.861**	0.103	**0.017**	0.948	**0.971**	0.866	**0.922**
vessel	0.592	**0.708**	0.163	**0.053**	0.818	**0.868**	0.730	**0.821**
mean	0.621	**0.749**	0.265	**0.073**	0.854	**0.906**	0.743	**0.846**

Table 3. Performance metrics for multi-view reconstruction using 5 random views of objects from categories seen at training time, where we see that 3D43D achieves consistently better reconstruction.
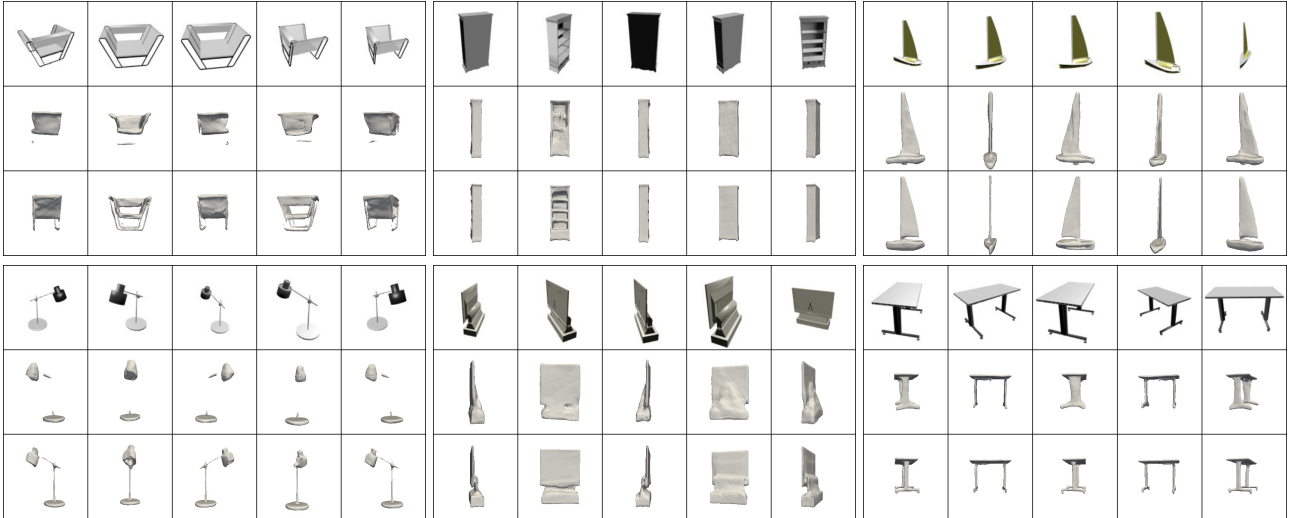

Figure 4. Reconstructions from categories seen during training time using 5 input views. For each object: (**Top row**) Input views. (**Middle row**) OccNets [19] prediction (orbit of 5 views of the predicted mesh). (**Bottom row**) 3D43D prediction (orbit of 5 views of the predicted mesh). We can qualitatively see that 3D43D produces better results than OccNets [19] in terms of high-frequency geometry. Note that input views and reconstructions are not presented from the shown viewpoint.

on 3 ShapeNet categories: plane, chair and car. We then report results on the test set of 10 unseen categories. We train and evaluate our model with 4 views and report the average IoU across the unseen classes in Tab. 5, where we show that our novel contributions contribute to improve the reconstruction accuracy.

Figure 5. Reconstruction of objects from unseen categories when training OccNets[19] and 3D43D on *Cars*, *Chairs* and *Airplanes*. For each object: (**Top row**) Input views. (**Middle row**) OccNets [19] prediction (orbit of 5 views of the predicted mesh). (**Bottom row**) 3D43D prediction (orbit of 5 views of the predicted mesh). OccNets [19] commonly map unseen categories to categories seen at training time. In comparison, 3D43D reconstructions are more accurate and less biased towards training categories. Note that input views and reconstructions are not shown from the same viewpoint.

	IoU			↓Chamfer-L_1			Normal Consistency			F-score		
Unseen category	[19]	[19]-v	3D43D	[19]	[19]-v	3D43D	[19]	[19]-v	3D43D	[19]	[19]-v	3D43D
(1 view)												
bench	0.251	0.291	**0.302**	0.752	**0.323**	0.357	0.714	**0.733**	0.706	0.374	0.426	**0.447**
cabinet	0.282	0.404	**0.502**	1.102	0.621	**0.529**	0.662	0.739	**0.759**	0.418	0.551	**0.647**
display	0.117	0.162	**0.243**	3.213	1.836	**1.389**	0.546	0.612	**0.638**	0.197	0.260	**0.364**
lamp	0.100	0.150	**0.223**	3.482	2.276	**1.997**	0.582	**0.625**	0.618	0.166	0.241	**0.340**
loudspeaker	0.311	0.405	**0.507**	1.649	0.860	**0.744**	0.655	0.731	**0.749**	0.452	0.552	**0.649**
rifle	0.155	0.150	**0.236**	2.465	2.206	**0.707**	0.539	0.527	**0.588**	0.255	0.252	**0.3737**
sofa	0.493	0.552	**0.559**	0.915	**0.399**	0.421	0.761	**0.799**	0.784	0.625	0.688	**0.699**
table	0.172	0.214	**0.313**	1.304	0.861	**0.583**	0.686	0.722	**0.731**	0.275	0.331	**0.461**
telephone	0.052	0.155	**0.271**	1.673	1.062	**0.996**	0.654	0.682	**0.700**	0.096	0.256	**0.403**
vessel	0.324	0.378	**0.401**	0.849	0.592	**0.521**	0.648	**0.691**	0.690	0.463	0.525	**0.553**
mean	0.226	0.286	**0.356**	1.740	1.104	**0.824**	0.645	0.686	**0.696**	0.332	0.408	**0.494**
(5 views)												
bench	0.288	0.147	**0.463**	0.508	1.960	**0.113**	0.729	0.625	**0.800**	0.421	0.242	**0.617**
cabinet	0.295	0.312	**0.629**	0.917	1.273	**0.250**	0.674	0.655	**0.844**	0.430	0.458	**0.756**
display	0.120	0.127	**0.409**	2.868	3.179	**0.428**	0.560	0.534	**0.770**	0.200	0.213	**0.558**
lamp	0.100	0.138	**0.369**	3.365	2.653	**2.057**	0.586	0.623	**0.738**	0.167	0.224	**0.513**
loudspeaker	0.315	0.333	**0.627**	1.460	1.344	**0.392**	0.660	0.677	**0.829**	0.457	0.480	**0.753**
rifle	0.180	0.095	**0.498**	1.866	2.610	**0.115**	0.567	0.444	**0.760**	0.290	0.169	**0.655**
sofa	0.525	0.356	**0.679**	0.732	1.445	**0.147**	0.776	0.663	**0.858**	0.656	0.508	**0.795**
table	0.186	0.177	**0.455**	1.122	1.771	**0.255**	0.694	0.700	**0.827**	0.295	0.285	**0.609**
telephone	0.036	0.131	**0.549**	1.588	1.457	**0.184**	0.689	0.592	**0.861**	0.066	0.226	**0.691**
vessel	0.347	0.256	**0.521**	0.683	1.524	**0.145**	0.661	0.603	**0.776**	0.489	0.390	**0.669**
mean	0.239	0.207	**0.520**	1.511	1.922	**0.409**	0.660	0.612	**0.806**	0.347	0.319	**0.662**

Table 4. Performance metrics for single and multi-view reconstruction when generalizing to unseen object categories after training only on the **car**, **chair** and **plane** categories.

	P	P+C	P+C+V (3D43D)
IoU	0.453	0.476	**0.491**

Table 5. Results of our ablation experiments.

5. Conclusions

In this paper, we studied factors that impact the generalization of learning-based 3D reconstruction models to unseen categories during training. We argued that to generalize successfully to unseen classes all these factors need to be addressed together. We empirically showed that by taking this into when designing our model, we obtain large improvements over state-of-the-art methods when reconstructing objects of on unseen categories. These improvements in generalization are a step forward for learned 3D reconstruction methods, which will enable recent Neural Rendering approaches [21, 27, 5] to go beyond the constrained scenario of training category-specific models. We believe that having a clear understanding of these factors and their compound effects will enrich this promising avenue of research.

References

[1] 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.

[2] 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] 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.

[4] Boyang Deng, Kyle Genova, Soroosh Yazdani, Sofien Bouaziz, Geoffrey Hinton, and Andrea Tagliasacchi. Cvxnet: Learnable convex decomposition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 31–44, 2020.

[5] Emilien Dupont, Miguel Angel Bautista, Alex Colburn, Aditya Sankar, Carlos Guestrin, Josh Susskind, and Qi Shan. Equivariant neural rendering. *arXiv preprint arXiv:2006.07630*, 2020.

[6] Carlos Esteves, Christine Allen-Blanchette, Ameesh Makadia, and Kostas Daniilidis. Learning so (3) equivariant representations with spherical cnns. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 52–68, 2018.

[7] Haoqiang Fan, Hao Su, and Leonidas J Guibas. A point set generation network for 3d object reconstruction from a single image. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 605–613, 2017.

[8] Thibault Groueix, Matthew Fisher, Vladimir G Kim, Bryan C Russell, and Mathieu Aubry. Atlasnet: A papier-m\^ ach\'e approach to learning 3d surface generation. *arXiv preprint arXiv:1802.05384*, 2018.

[9] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[10] Eldar Insafutdinov and Alexey Dosovitskiy. Unsupervised learning of shape and pose with differentiable point clouds. In *Advances in Neural Information Processing Systems*, pages 2802–2812, 2018.

[11] Max Jaderberg, Karen Simonyan, Andrew Zisserman, et al. Spatial transformer networks. In *Advances in neural information processing systems*, pages 2017–2025, 2015.

[12] Timothy Jeruzalski, Boyang Deng, Mohammad Norouzi, John P Lewis, Geoffrey Hinton, and Andrea Tagliasacchi. Nasa: Neural articulated shape approximation. *arXiv preprint arXiv:1912.03207*, 2019.

[13] Angjoo Kanazawa, Shubham Tulsiani, Alexei A Efros, and Jitendra Malik. Learning category-specific mesh reconstruction from image collections. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 371–386, 2018.

[14] Abhishek Kar, Christian Häne, and Jitendra Malik. Learning a multi-view stereo machine. In *Advances in neural information processing systems*, pages 365–376, 2017.

[15] Hiroharu Kato, Yoshitaka Ushiku, and Tatsuya Harada. Neural 3d mesh renderer. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3907–3916, 2018.

[16] Tejas Khot, Shubham Agrawal, Shubham Tulsiani, Christoph Mertz, Simon Lucey, and Martial Hebert. Learning unsupervised multi-view stereopsis via robust photometric consistency. *arXiv preprint arXiv:1905.02706*, 2019.

[17] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[18] Shichen Liu, Weikai Chen, Tianye Li, and Hao Li. Soft rasterizer: Differentiable rendering for unsupervised single-view mesh reconstruction. *arXiv preprint arXiv:1901.05567*, 2019.

[19] 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.

[20] Mateusz Michalkiewicz, Jhony K Pontes, Dominic Jack, Mahsa Baktashmotlagh, and Anders Eriksson. Deep level sets: Implicit surface representations for 3d shape inference. *arXiv preprint arXiv:1901.06802*, 2019.

[21] Ben Mildenhall, Pratul P Srinivasan, Matthew Tancik, Jonathan T Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. *arXiv preprint arXiv:2003.08934*, 2020.

[22] 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.

[23] 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.

[24] Shunsuke Saito, Zeng Huang, Ryota Natsume, Shigeo Morishima, Angjoo Kanazawa, and Hao Li. Pifu: Pixel-aligned implicit function for high-resolution clothed human digitization. *arXiv preprint arXiv:1905.05172*, 2019.

[25] Steven M Seitz, Brian Curless, James Diebel, Daniel Scharstein, and Richard Szeliski. A comparison and evaluation of multi-view stereo reconstruction algorithms. In *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*, volume 1, pages 519–528. IEEE, 2006.

[26] 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.

[27] Vincent Sitzmann, Michael Zollhöfer, and Gordon Wetzstein. Scene representation networks: Continuous 3d-structure-aware neural scene representations. In *Advances in Neural Information Processing Systems*, pages 1121–1132, 2019.

[28] David Stutz and Andreas Geiger. Learning 3d shape completion from laser scan data with weak supervision. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1955–1964, 2018.

[29] Supasorn Suwajanakorn, Noah Snavely, Jonathan J Tompson, and Mohammad Norouzi. Discovery of latent 3d keypoints via end-to-end geometric reasoning. In *Advances in Neural Information Processing Systems*, pages 2059–2070, 2018.

[30] Maxim Tatarchenko, Stephan R Richter, René Ranftl, Zhuwen Li, Vladlen Koltun, and Thomas Brox. What do single-view 3d reconstruction networks learn? In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3405–3414, 2019.

[31] Shubham Tulsiani, Alexei A Efros, and Jitendra Malik. Multi-view consistency as supervisory signal for learning shape and pose prediction. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2897–2905, 2018.

[32] Shimon Ullman. The interpretation of structure from motion. *Proceedings of the Royal Society of London. Series B. Biological Sciences*, 203(1153):405–426, 1979.

[33] Nanyang Wang, Yinda Zhang, Zhuwen 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.

[34] Qiangeng Xu, Weiyue Wang, Duygu Ceylan, Radomir Mech, and Ulrich Neumann. Disn: Deep implicit surface network for high-quality single-view 3d reconstruction. *arXiv preprint arXiv:1905.10711*, 2019.

[35] Yao Yao, Zixin Luo, Shiwei Li, Tian Fang, and Long Quan. Mvsnet: Depth inference for unstructured multi-view stereo. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 767–783, 2018.

[36] Xiuming Zhang, Zhoutong Zhang, Chengkai Zhang, Josh Tenenbaum, Bill Freeman, and Jiajun Wu. Learning to reconstruct shapes from unseen classes. In *Advances in Neural Information Processing Systems*, pages 2257–2268, 2018.