Learning 3D Shape Completion from Laser Scan Data with Weak Supervision

David Stutz[1,2] Andreas Geiger[1,3]
[1]Autonomous Vision Group, MPI for Intelligent Systems and University of Tübingen
[2]Computer Vision and Multimodal Computing, Max-Planck Institute for Informatics, Saarbrücken
[3]Computer Vision and Geometry Group, ETH Zürich
david.stutz@mpi-inf.mpg.de, andreas.geiger@tue.mpg.de

Abstract

3D shape completion from partial point clouds is a fundamental problem in computer vision and computer graphics. Recent approaches can be characterized as either data-driven or learning-based. Data-driven approaches rely on a shape model whose parameters are optimized to fit the observations. Learning-based approaches, in contrast, avoid the expensive optimization step and instead directly predict the complete shape from the incomplete observations using deep neural networks. However, full supervision is required which is often not available in practice. In this work, we propose a weakly-supervised learning-based approach to 3D shape completion which neither requires slow optimization nor direct supervision. While we also learn a shape prior on synthetic data, we amortize, i.e., learn, *maximum likelihood fitting using deep neural networks resulting in efficient shape completion without sacrificing accuracy. Tackling 3D shape completion of cars on ShapeNet [5] and KITTI [18], we demonstrate that the proposed amortized maximum likelihood approach is able to compete with a fully supervised baseline and a state-of-the-art data-driven approach while being significantly faster. On ModelNet [49], we additionally show that the approach is able to generalize to other object categories as well.*

1. Introduction

3D shape perception is a long-standing problem both in human [35, 36] and computer vision [17]. In both disciplines, a large body of work focuses on 3D reconstruction, e.g., reconstructing objects or scenes from one or multiple views, which is an inherently ill-posed inverse problem where many configurations of shape, color, texture and lighting may result in the very same image [17]. Both human and computer vision are related through insights regarding the cues and constraints used by humans to perceive 3D shapes. Motivated by results from human vision [35, 36], these priors are usually built into 3D reconstruction pipelines through explicit assumptions. Recently, however – leveraging the success of deep learning – researchers started to *learn* shape models from data. Predominantly generative models have been used to learn how to generate, manipulate and reason about 3D shapes, e.g., [4, 20, 41, 48, 49], thereby offering many interesting possibilities for a wide variety of problems.

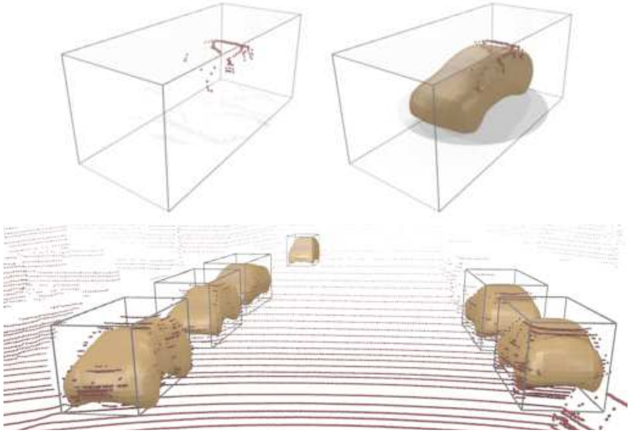

Figure 1: **Illustration of the 3D Shape Completion Problem.** Top: Given a 3D bounding box and an incomplete point cloud (left, red), our goal is to predict the complete shape of the object (right, beige). Bottom: Shape completion results on a street scene from KITTI [18]. Learning shape completion on real-world data is challenging due to sparse / noisy observations and missing ground truth.

In this paper, we focus on the problem of inferring and completing 3D shapes based on sparse and noisy 3D point observations as illustrated in Fig. 1. This problem occurs when only a single view of an individual object is provided or large parts of the object are occluded as, e.g., in autonomous driving applications. Existing approaches to shape completion can be roughly categorized into data-driven and learning-based methods. The former usually rely on learned shape priors and formulate shape comple-

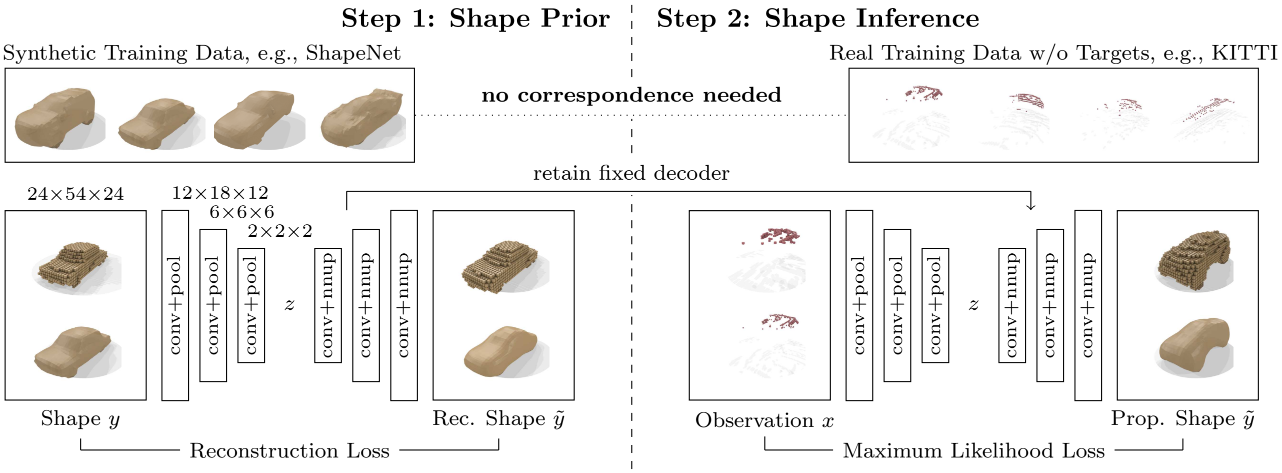

Figure 2: **Proposed Amortized Maximum Likelihood (AML) Approach to 3D Shape Completion.** We illustrate our amortized maximum likelihood (AML) approach on KITTI [18]. We consider two steps. In step 1 (left), we use car models from ShapeNet [5] to train a variational auto-encoder (VAE) [26]. In our case, the car models are encoded using occupancy grids and signed distance functions (SDFs) at a resolution of 24 × 54 × 24 voxels. In step 2 (right), we retain the pre-trained decoder (with fixed weights) and train a novel deterministic encoder. This network can be trained using a maximum likelihood loss without requiring further supervision. The pre-trained decoder constrains the predictions to valid car shapes while the maximum likelihood loss aligns the predictions with the observations. See text for further details.

tion as optimization problem over the corresponding (lower-dimensional) latent space [3, 10, 13, 22]. These approaches have demonstrated impressive performance on real data, e.g., on KITTI [18]. Learning-based approaches, in contrast, assume a fully supervised setting in order to directly learn shape completion on synthetic data [9, 15, 37, 39, 41, 42]. As full supervision is required, the applicability of these approaches to real data is limited. However, learning-based approaches offer advantages in terms of efficiency: a forward pass of the learned network is usually sufficient. In practice, both problems – the optimization problem of data-driven approaches and the required supervision of learning-based approaches – limit the applicability of state-of-the-art shape completion methods to real data.

To tackle these problems, this work proposes an amortized maximum likelihood approach for 3D shape completion. More specifically, we first learn a shape model on synthetic data using a variational auto-encoder [26] (cf. Figure 2, step 1). Shape completion can then be formulated as maximum likelihood problem – in the spirit of [13]. Instead of maximizing the likelihood independently for distinct observations, however, we follow the idea of amortized inference [19] and *learn* to predict the maximum likelihood solutions directly given the observations. Towards this goal, we train a new encoder which embeds the observations in the same latent space using an unsupervised maximum likelihood loss (cf. Figure 2, step 2). This allows us to learn 3D shape completion in challenging real-world situations, e.g., on KITTI. Using signed distance functions to represent shapes, we are able to obtain sub-voxel accuracy while applying regular 3D convolutional neural networks to voxel grids of limited resolution, yielding a highly efficient inference method. For experimental evaluation, we introduce two novel, synthetic shape completion benchmarks based on ShapeNet and ModelNet. On KITTI, we further compare our approach to the work of Engelmann et al. [13] – the only related work which addresses shape completion on KITTI. Our experiments demonstrate that we obtain shape reconstructions which rival data-driven techniques while significantly reducing inference time. Our code and datasets will be made publicly available[1].

This paper is structured as follows: we discuss related work in Section 2. In Section 3 we describe our amortized maximum likelihood framework for weakly-supervised shape completion. We present experimental results in Section 4 and conclude in Section 5.

2. Related Work

Symmetry-based and Data-driven Methods: Shape completion is usually performed on partial scans of individual objects. Following [44], classical shape completion approaches can roughly be categorized into symmetry-based methods and data-driven methods. The former leverage observed symmetry to complete shapes; representative works include [27, 29, 34, 46, 51]. The data-driven case is more interesting in relation to the proposed approach. In early work, Pauly et al. [33] pose shape completion as retrieval and alignment problem. In [3, 10, 13, 14, 21, 30, 32] shape

[1] https://avg.is.tuebingen.mpg.de/research_projects/3d-shape-completion.

retrieval is avoided by learning a latent shape space. The alignment task is then posed as an optimization problem over the latent shape variables. Data-driven approaches are applicable to real data assuming knowledge about the category of shapes in order to learn the shape prior. However, they require costly optimization at inference time. In contrast, we propose an approach which amortizes the inference procedure by means of a deep neural network allowing for efficient completion of 3D shapes.

Learning-based Methods: With the recent success of deep learning, several learning-based approaches have been proposed [8, 15, 16, 23, 37, 39, 41, 42]. Strictly speaking, those techniques are data-driven as well, however, shape retrieval and fitting is avoided by learning shape completion under full supervision on synthetic datasets such as ShapeNet [5] or ModelNet [49] – usually using deep neural networks. Some approaches [24, 39, 45] use octrees to predict high-resolution shapes via supervision provided at multiple scales. However, full supervision for the 3D shape is often not available in real-world situations (e.g., KITTI [18]), thus existing models are primarily evaluated on synthetic datasets. In this paper, we propose to train a shape prior on synthetic data, but leverage unlabeled real-world data for learning shape completion.

Amortized Inference: The notion of amortized inference was introduced in [19] and exploited repeatedly in recent work [38, 40, 47]. Generally, it describes the idea of *learning how to infer*; in our case, we learn, i.e. amortize, the maximum likelihood inference problem by training a network to directly predict maximum likelihood solutions.

3. Method

In the following, we first introduce the mathematical formulation of the weakly-supervised 3D shape completion problem. Subsequently, we briefly discuss the concept of variational auto-encoders (VAEs) [26] which we use to learn a shape prior. Finally, we formally derive our proposed amortized maximum likelihood (AML) approach. The overall framework is also illustrated in Figure 2.

3.1. Problem Formulation

In a supervised setting, our task can be described as follows: given a set of partial observations $\mathcal{X} = \{x_n\}_{n=1}^N \subseteq \mathbb{R}^R$ and corresponding ground truth shapes $\mathcal{Y}^* = \{y_n^*\}_{n=1}^N \subseteq \mathbb{R}^R$, learn a mapping $x_n \mapsto y_n^*$ that is able to generalize to previously unseen observations. Here, we assume $\mathbb{R}^R$ to be a suitable vector representation of observations and shapes; in practice, we resort to occupancy grids or signed distance functions (SDFs) defined on regular grids, i.e., $x_n, y_n^* \in \mathbb{R}^{H \times W \times D} \simeq \mathbb{R}^R$. SDFs represent the distance of each voxel's center to the closest point on the surface; we use negative signs for interior voxels. For the (partial) observations, we write $x_n \in \{0, 1, \perp\}^R$ to make missing information explicit; in particular, $x_{n,i} = \perp$ corresponds to unobserved voxels, while $x_{n,i} = 1$ and $x_{n,i} = 0$ correspond to occupied and unoccupied voxels, respectively.

On real data, e.g., KITTI [18], supervised learning is often not possible as obtaining ground truth annotations is labor intensive (e.g., [31,50]). Therefore, we target a weakly-supervised variant of the problem instead. Given observations $\mathcal{X}$ and a set of reference shapes $\mathcal{Y} = \{y_m\}_{m=1}^M \subseteq \mathbb{R}^R$ both of the same, known object category, learn a mapping $x_n \mapsto \tilde{y}(x_n)$ such that the predicted shape $\tilde{y}(x_n)$ matches the unknown ground truth shape y_n^* as close as possible. Here, supervision is provided in the form of the known object category, allowing to derive the reference shapes from (watertight) triangular meshes; on real data, we also assume the object locations to be given in the form of 3D bounding boxes in order to extract the observations $\mathcal{X}$.

3.2. Shape Prior

We propose to use the provided reference shapes $\mathcal{Y}$ to learn a model of possible 3D shapes over the latent space $\mathcal{Z} = \mathbb{R}^Q$ with $Q \ll R$. The prior model is learned using a VAE where the joint distribution $p(y, z)$ decomposes into $p(y, z) = p(y|z)p(z)$ with $p(z)$ being a unit Gaussian, i.e., $p(z) = \mathcal{N}(z; 0, I_Q)$ with $I_Q \in \mathbb{R}^{R \times R}$ being the identity matrix. Sampling from the model is then performed by choosing $z \sim p(z)$ and subsequently sampling $y \sim p(y|z)$. For training the generative model, we also need to approximate the posterior $q(z|y) \approx p(z|y)$, i.e., the inference model. In the framework of the variational auto-encoder, both the so-called recognition model $q(z|y)$ and the generative model $p(y|z)$ – corresponding to encoder and decoder – are represented by neural networks. In particular,

$$q(z|y) = \mathcal{N}(z_i; \mu_i(y), \mathrm{diag}(\sigma_i^2(y))) \tag{1}$$

where $\mu(y), \sigma^2(y) \in \mathbb{R}^Q$ are predicted using the encoder neural network and $p(y_i|z)$ is assumed to be a Bernoulli distribution when working with occupancy grids, i.e., $p(y_i|z) = \mathrm{Ber}(y_i; \theta_i(z))$ while a Gaussian distribution is used when predicting SDFs, i.e., $p(y_i|z) = \mathcal{N}(y_i; \mu_i(z), \sigma^2)$. In both cases, the parameters, i.e., $\theta_i(z)$ or $\mu_i(z)$, are predicted using the decoder neural network. For SDFs, we neglect the variance ($\sigma^2 = 1$) as it merely scales the training objective.

In the framework of variational inference, the parameters of the encoder and the decoder are found by maximizing the likelihood $p(y)$. In practice, the likelihood is often intractable. Instead, the evidence lower bound is maximized, resulting in the following loss to be minimized:

$$\mathcal{L}_{\text{VAE}}(w) = -\mathbb{E}_{q(z|y)}[\ln p(y|z)] + \mathrm{KL}(q(z|y)|p(z)). \tag{2}$$

where w are the weights of the encoder and decoder. The Kullback-Leibler divergence KL can be computed analytically; the expectation corresponds to a binary cross-entropy error for occupancy grids or a scaled sum-of-squared error for SDFs. The loss $\mathcal{L}_{\text{VAE}}$ is minimized using stochastic gradient descent (SGD). We refer to [26] for details.

3.3. Shape Inference

After learning the shape prior $p(y, z) = p(y|z)p(z)$, shape completion can be formulated as a maximum likelihood (ML) problem over the lower-dimensional latent space $\mathcal{Z} = \mathbb{R}^Q$. The corresponding negative log-likelihood, i.e., $-\ln p(y, z)$, can be written as

$$\mathcal{L}_{\text{ML}}(z) = -\sum_{x_i \neq \perp} \ln p(y_i = x_i|z) - \ln p(z). \tag{3}$$

where $x_i \neq \perp$ expresses that the summation ranges only over observed voxels. As the prior $p(z)$ is Gaussian, the corresponding negative log-probability $-\ln p(z) \propto \|z\|_2^2$ results in a quadratic regularizer. As before, the generative model $p(y|z)$ decomposes over voxels. Instead of solving Equation (3) for each observation $x \in \mathcal{X}$ individually, we follow the idea of amortized inference [19] and train an encoder $z(x; w)$ to *learn* ML. To this end, we keep the generative model $p(y|z)$ fixed and train the weights w of the encoder $z(x; w)$ using the ML objective as loss:

$$\mathcal{L}_{\text{AML}}(w) = -\sum_{x_i \neq \perp} \ln p(y_i = x_i|z) - \lambda \ln p(z). \tag{4}$$

where we added an additional parameter λ controlling the importance of the shape prior. The exact form of the probabilities $p(y_i = x_i|z)$ depends on the used shape representation. In the case of occupancy grids, this term results in a cross-entropy error (as both y_i and x_i are, for $x_i \neq \perp$, binary). However, when using SDFs, the term is not well-defined as $p(y_i|z)$ is modeled with a continuous Gaussian distribution, while the observations x_i are binary, i.e., it is unclear how to define $p(y_i = x_i|z)$. Alternatively, we could derive distance values along the rays corresponding to observed points (e.g., following [43]). However, as illustrated in Figure 3, noisy rays lead to invalid observations along the whole ray. This problem can partly be avoided when relying on occupancy to represent the observations.

In order to still work with SDFs (to achieve sub-voxel accuracy) we propose to define $p(y_i = x_i|z)$ through a simple transformation. In particular, as $p(y_i|z)$ is modeled as Gaussian distribution $p(y_i|z) = \mathcal{N}(y_i; \mu_i(z), \sigma^2)$ where $\mu_i(z)$ is predicted using the fixed decoder (and $\sigma^2 = 1$) and x_i is binary (for $x_i \neq \perp$), we introduce a mapping $\theta_i(\mu_i(z))$ transforming the predicted Gaussian distribution to a Bernoulli distribution with occupancy probability $\theta_i(\mu_i(z))$, i.e., $p(y_i = x_i|z)$ becomes $\text{Ber}(y_i = x_i; \theta_i(\mu_i(z)))$. As we defined occupied voxels to have negative sign in the SDF, we can derive the occupancy probability $\theta_i(\mu_i(z))$ as the probability of a negative distance:

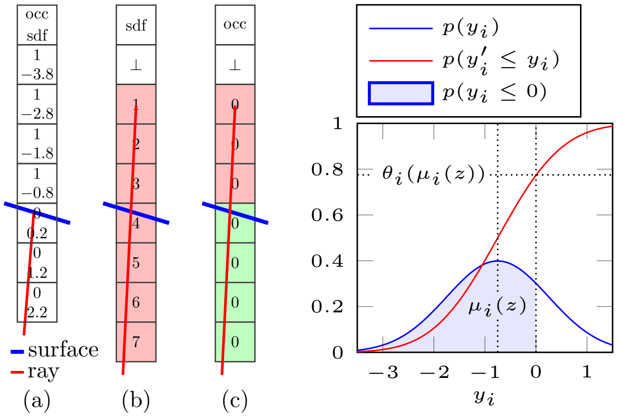

Figure 3: **Left: Problem when Predicting SDFs.** Illustration of a ray (red) correctly hitting a surface (blue) causing the SDF values and occupancy values in the underlying voxel grid to be correct (cf. (a)). A noisy ray, however, causes all voxels along the ray to get invalid distances assigned (marked red; cf. (b)). When using occupancy, in contrast, only the voxels behind the surface are assigned invalid occupancy states (marked red); the remaining voxels are labeled correctly (marked green; cf. (c)). **Right: Proposed Gaussian-to-Bernoulli Transformation.** For $p(y_i) := p(y_i|z) = \mathcal{N}(y_i; \mu_i(z), \sigma^2)$ (blue), we illustrate the transformation discussed in Section 3.3, allowing to use the binary observations x_i (for $x_i \neq \perp$) to supervise the SDF predictions. This is achieved, by transforming the predicted Gaussian distribution to a Bernoulli distribution with occupancy probability $\theta_i(\mu_i(z)) = p(y_i \leq 0)$ (blue area).

$$\theta_i(\mu_i(z)) = \mathcal{N}(y_i \leq 0; \mu_i(z), \sigma^2) \tag{5}$$

$$= \frac{1}{2}\left(1 + \text{erf}\left(\frac{-\mu_i(z)}{\sigma\sqrt{\pi}}\right)\right). \tag{6}$$

Here, erf is the error function which, in practice, is approximated following [1]. Equation (5) is illustrated in Figure 3 where the occupancy probability $\theta_i(\mu_i(z))$ is computed as the area under the Gaussian bell curve for $y_i \leq 0$. This per-voxel transformation can easily be implemented as non-linearity layer and its derivative wrt. $\mu_i(z)$ is – by construction – a Gaussian distribution. Overall, this transformation allows to predict SDFs while using binary observations.

4. Experimental Evaluation

In this section, we present quantitative and qualitative experimental results. First, we derive a synthetic benchmark for 3D shape completion of cars based on ShapeNet [5]. Second, we present results on KITTI [18] and compare the proposed amortized maximum likelihood (AML) approach to the data-driven approach of [13]. We also consider regular maximum likelihood (ML) and a fully-supervised model (Sup; following related work [8, 23, 39, 41]) as baselines. Finally, we consider additional object categories on ModelNet [49]. We provide complementary details and experiments in the supplementary material.

4.1. Datasets

ShapeNet: On ShapeNet, we took 3253 car models, and simplified them using the approach outlined in [22] to obtain watertight meshes. After random translation, rotation and scaling, we extract two sets: the reference shapes $\mathcal{Y}$ and the ground truth shapes $\mathcal{Y}^*$ (such that $\mathcal{Y} \cap \mathcal{Y}^* = \emptyset$). To train the shape prior using the reference shapes $\mathcal{Y}$, we derive signed distance functions (SDFs) and occupancy grids at a resolution of $24 \times 54 \times 24$ voxels. The ground truth shapes $\mathcal{Y}^*$ are rendered to obtain the observations $\mathcal{X}$. In particular, we identify occupied voxels, i.e., $x_{n,i} = 1$, by back-projecting pixels from the rendered depth image and perform ray tracing to identify free space, i.e., $x_{n,i} = 0$ (all other voxels are unknown, i.e., $x_{n,i} = \bot$).

In order to benchmark 3D shape completion, we consider two difficulties: a “clean” – or easy – version with depth images rendered at a resolution of 48×64 pixels and a “noisy” – or hard – version using a resolution of 24×32. On average, this results in 411 and 106 observed points (not necessarily voxels), respectively. For the latter variant, we additionally inject noise by (randomly) perturbing pixels or setting them to the maximum depth value to simulate rays (e.g., from a LiDAR sensor) passing through objects (e.g., due to specular or transparent surfaces). We refer to the created datasets as SN-clean and SN-noisy and show examples in Figure 4. Overall, we obtain 14640/14640/1950 samples for the prior training/inference training/validation set with roughly $1.06\%/0.32\%$ observed voxels and $7.04\%/4.8\%$ free space voxels for SN-clean/SN-noisy. As can be seen, SN-clean and SN-noisy include a large variety of car models and SN-noisy, in particular, captures the difficulty of real data, e.g. from KITTI, by explicitly modeling sparsity and noise.

KITTI: On KITTI, we extract observations using the provided ground truth 3D bounding boxes to avoid the inaccuracies of 3D object detectors. We used KITTI's Velodyne point clouds from the 3D object detection benchmark and the training/validation split of [6] (7140/7118 samples). Based on the average aspect ratio of cars in the dataset, we voxelize the points within the 3D bounding boxes into occupancy grids of size $24 \times 54 \times 24$ and perform ray tracing to obtain the observations $\mathcal{X}$. We filtered the observations to contain at least 50 observed points to avoid overly sparse observations. On average, we obtained 0.3% observed voxels and 3.35% free space voxels. For the bounding boxes in the validation set, we generated partial ground truth by considering 10 future and 10 past frames and accumulating the corresponding 3D points according to the ground truth bounding boxes. In Figure 4, we show examples of the extracted observations and ground truth. Overall, the extracted observations are very sparse and noisy and ground truth is not available for every observation.

ModelNet: On ModelNet, we consider the object categories bathtub, dresser, monitor, nightstand, sofa and toilet. We use a resolution of $32 \times 32 \times 32$ (similar to [49]) and rely purely on occupancy grids as thin structures make SDFs unreliable in low resolution. Reference shapes $\mathcal{Y}$, ground truth shapes $\mathcal{Y}^*$ and observations $\mathcal{X}$ are obtained following the procedure for SN-clean (without simplification of the models). This results in – on average – 1.04% observed voxels and 7.24% free space voxels. Overall, we obtained a minimum of 700/700/150 samples for prior training/inference training/validation per category. The large intra-category variations contribute to the difficulty of the task on ModelNet; we show examples in Figure 5.

4.2. Architecture and Training

We rely on a simple, shallow architecture to predict both occupancy grids and (if applicable) SDFs in separate channels. Instead of predicting SDFs directly, we predict log-transformed SDFs, i.e., for signed distance y_i we compute $\text{sign}(y_i) \log(1 + |y_i|)$. As in depth prediction [11, 12, 28], this transformation reduces the overall range while enlarging the relative range around the boundaries. On ShapeNet, the encoder and the decoder of the variational auto-encoder (VAE) [26] comprise three convolutional stages including batch normalization, ReLU activations and max pooling/nearest neighbor upsampling with 3^3 kernels and 24, 48 and 96 channels; the resolution is reduced to 2^3. On ModelNet, we use four stages with 24, 48, 96 and 96 channels. When predicting occupancy probabilities we use Sigmoid activations in the last layer of the decoder. We use stochastic gradient descent (SGD) with momentum and weight decay for training.

The encoder $z(x; w)$ trained for shape inference follows the architecture of the recognition model $q(z|x)$ and takes occupancy grids and (if applicable) DFs of the observations as input. The code z, however, is directly predicted. While training the encoder $z(x; w)$, the generative model is kept fixed. In order to obtain well-performing models for shape inference, we found that it is of crucial importance that the encoder predicts high-probability codes (i.e., under the Gaussian prior $p(z)$). Therefore, we experimentally set $\lambda = 15$ on SN-clean and ModelNet, $\lambda = 50$ on SN-noisy and $\lambda = 10$ on KITTI (cf. Equation (4)). As before, we train the encoder using SGD with momentum and weight decay. On SN-noisy and KITTI, we additionally weight the per-voxel terms in Equation (4) for $x_i = 0$ by the probabil-

	SN-clean (val)			SN-noisy (val)			KITTI (val)	
	Ham	Acc [vx]	Comp [vx]	Ham	Acc [vx]	Comp [vx]	Comp [m]	t [s]
VAE	**0.014**	**0.283**	**0.439**					
ML	0.04	0.733	0.845	0.059	1.145	1.331		30
Sup (on KITTI GT)	**0.022**	**0.425**	**0.575**	**0.027**	**0.527**	**0.751**	**0.176 (0.174)**	**0.001**
AML Q=5	**0.041**	0.752	0.877	**0.061**	1.203	1.39	**0.091**	
AML w/o weighted free space	0.043	**0.739**	**0.845**	**0.061**	1.228	1.327	0.117	**0.001**
AML (on KITTI GT)				0.062	**1.161**	**1.203**	0.1 (**0.091**)	
[13] (on KITTI GT)		1.164	0.99		1.713	1.211	0.131 (0.129)	0.168*

Table 1: **Quantitative Results.** On SN-clean and SN-noisy, we report Hamming distance (Ham), accuracy (Acc) and completeness (Comp) (cf. Section 4.3). Both Acc and Comp are in voxels, i.e. as multiples of the voxel edge length. On KITTI [18], we only report Comp in meters. For all metrics, **lower is better**. We also report the average runtime per sample. All results were obtained on the corresponding validation sets. * Runtimes on an Intel® Xeon® E5-2690 @2.6Ghz using (multi-threaded) Ceres [2]; remaining runtimes on a NVIDIA™ GeForce® GTX TITAN using Torch [7].

ity of free space at voxel i on the training set of the shape prior, i.e., SN-clean. We found that this reduces the impact of noise.

4.3. Evaluation

For evaluation, we consider metrics reflecting the employed shape representations. For occupancy grids, we use the Hamming distance (Ham) between the (thresholded) predictions and the ground truth. For SDFs, we consider a mesh-to-mesh distance on SN-clean and SN-noisy and a mesh-to-point distance on KITTI. In both cases, we follow [25] and consider accuracy (Acc) and completeness (Comp). To measure accuracy, we sample roughly $10k$ points on the reconstructed mesh; for each point, we then compute the distance to the target mesh and report the average. Analogously, completeness is the distance from the target mesh (or the ground truth points on KITTI) to the reconstructed mesh. Note that for both Acc and Comp, lower is better. On SN-clean and SN-noisy, we report both Acc and Comp in voxels, i.e., in multiples of the voxel edge length (as we do not know the absolute scale of ShapeNet's car models); on KITTI, we only report Comp in meters.

4.4. Baselines

We consider regular ML as well as a fully-supervised model (Sup) as baselines. For the former, we applied SGD on an initial code of $z = 0$ until the change in objective is insignificant. As supervised baseline we train the VAE shape prior architecture, using the very same training procedure, to directly perform 3D shape completion – i.e., to predict completed shapes given the observations. Note that in contrast to the proposed approach, this baseline has access to full supervision during training (i.e., full shapes, not only the observations). This baseline also represents related learning-based approaches [8, 15, 23, 39, 41, 42] which are unsuitable for a fair comparison due to our low-dimensional bottleneck and as architectures are not trivially adjustable to our setting (e.g., resolution and SDFs). Additionally, we consider the data-driven method proposed in [13] which iteratively optimizes both the pose and the shape based on a principal component analysis (PCA) shape prior with latent space dimensionality $Q = 5$[2]. On KITTI, we adapted the method to only optimize the shape, as the pose is provided through the ground truth 3D bounding boxes. On SN-clean and SN-noisy, in contrast, we optimize both pose and shape as [13] expects a common ground plane, which is not the case on SN-clean or SN-noisy by construction.

4.5. Results

Our results on ShapeNet and KITTI are summarized in Table 1 and Figure 4; results on ModelNet are presented in Table 2 and Figure 5. For our experiments, choosing Q is of crucial importance – large Q allows to capture details and variation, but the latent space is more likely to contain unreasonable shapes; small Q prevents the model from reconstructing shapes in detail. On SN-clean, we determined $Q = 10$ to be suitable; for fair comparison to [13] we also report selected results for $Q = 5$. On ModelNet, in contrast, we use $Q = 25$ and $Q = 100$ for category-specific (i.e., one model per category) and -agnostic (i.e., one model for all six categories) models, respectively.

4.5.1 Shape Completion on ShapeNet

On SN-clean and SN-noisy, we follow Table 1, demonstrating that AML outperforms related work [13] and performs on par with ML while significantly reducing runtime. As reference point, we also report the reconstruction performance of the VAE shape prior as lower bound on the achievable Ham, Acc and Comp. Sup, in contrast, performs well and represents the achievable performance under full supervision. Interestingly, ML performs reasonably well; on SN-clean and SN-noisy, ML exhibits less than double the error compared to Sup while using only 8% supervi-

[2] Code and shape prior (without models for training) from https://github.com/VisualComputingInstitute/ShapePriors_GCPR16.

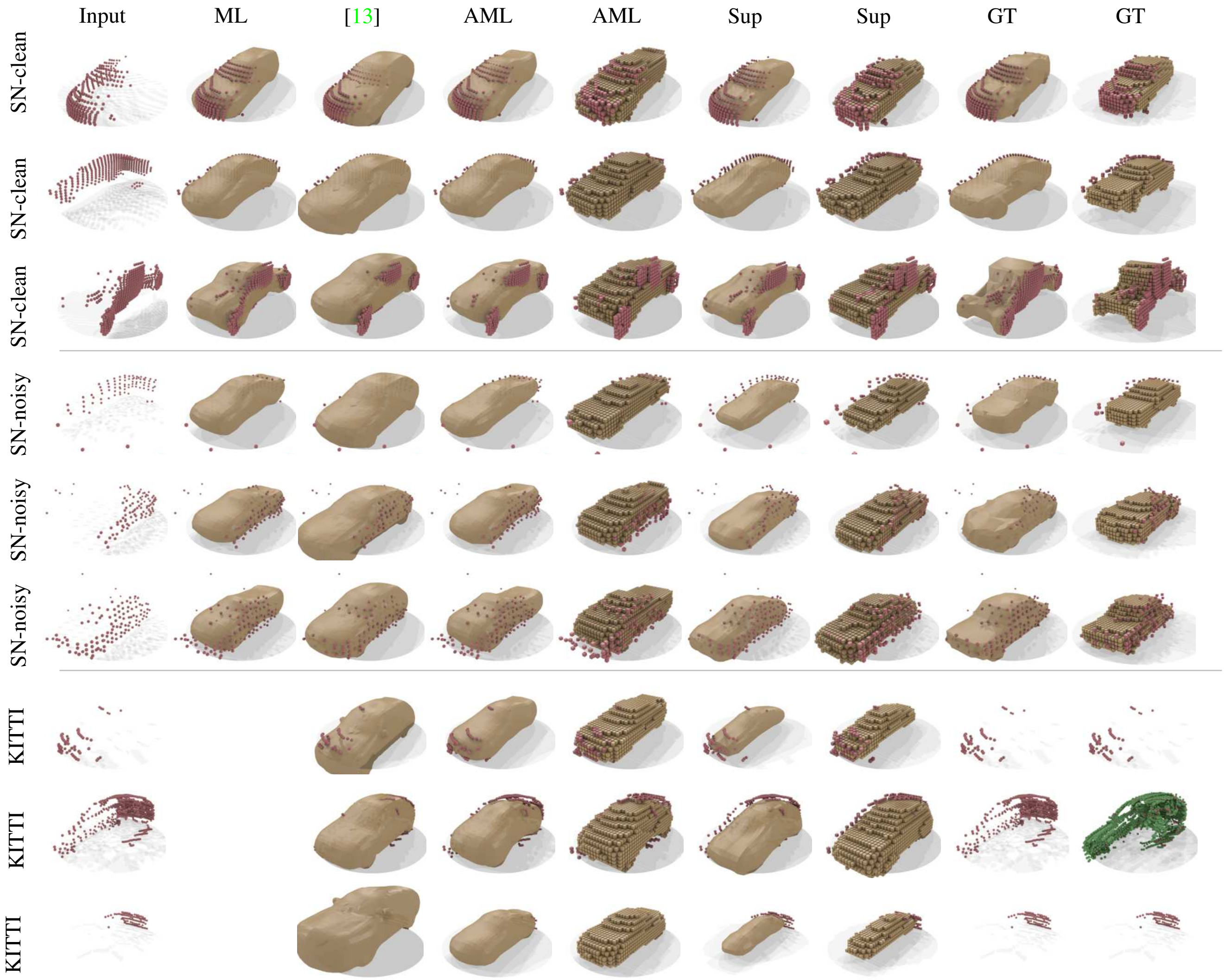

Figure 4: **Qualitative Results.** On SN-clean and SN-noisy we show results for ML, [13], AML and Sup as well as ground truth shapes. On KITTI, ground truth shapes are not available; we show results for [13], AML and Sup as well as accumulated ground truth points (green). We present predicted shapes (meshes and occupancy grids, beige) and observations (red).

sion or less. AML demonstrates performance on par with ML; this means that amortized inference is able to preserve performance while reducing runtime significantly. On SN-clean and SN-noisy, AML easily outperforms related work [13], even for $Q = 5$. However, we note that [13] was originally proposed for KITTI. Overall, AML demonstrates good shape completion performance at low runtime and without full supervision.

We also consider qualitative results in Figure 4 showing meshes and occupancy grids for ML, [13], AML and Sup. On SN-clean, the high number of observed points ensures that all methods predict reasonable shapes. In the second row, we notice that Sup is not always able to predict the correct shape while AML and ML are and that [13] has difficulties predicting the correct size of the car. Surprisingly, ML comes most closely to the ground truth car in row three. We suspect that ML is able to overfit to these exotic cars while AML is required to generalize based on the cars seen during training. On SN-noisy, all methods have significant difficulties predicting reasonable cars. Interestingly, we notice that [13] has a bias towards larger station wagons or cars with hatchback while AML, ML and Sup prefer to predict thinner cars. This illustrates that the shape prior takes over more responsibility when less observations are available. Overall, we notice that SN-clean is – by construction – considerably easier than SN-noisy. Based on both quantitative and qualitative results, we find that AML outperforms related work [13] while being significantly faster and allowing – in contrast to Sup – to be trained on unannotated real data as we demonstrate in the next section.

	Ham		
	VAE	AML	Sup
bathtub	0.015	0.037	0.025
dresser	0.018	0.069	0.036
monitor	0.013	0.036	0.023
nightstand	0.03	0.099	0.065
sofa	0.011	0.028	0.019
toilet	0.02	0.053	0.033
all	0.016	0.065	0.035

Table 2: **Quantitative Results on ModelNet.** We report Hamming distance (Ham) for both category-specific as well as -agnostic (cf. "all") models on ModelNet; **lower is better**. Results were obtained on the validation sets.

4.5.2 Shape Completion on KITTI

On KITTI, considering Table 1, we focused on AML, Sup and related work [13]. We note that completeness (Comp) is reported in meters. Sup as well as the method by Engelmann et al. [13] come close to an average of 10cm, while only AML is able to actually reduce Comp to 9.1cm. We also report results for AML, Sup and [13] applied to KITTI's ground truth, i.e., using the ground truth points as input. In this case, performance slightly increases, but AML still outperforms Sup showing that Sup is not able to generalize well. As the ground truth is noisy, however, the performance differences are not significant enough. Therefore, runtime and the level of supervision gain importance. Regarding the former, AML exhibits significantly lower runtime compared to [13]; regarding the latter, AML requires considerably less supervision compared to Sup. Overall, this shows the advantage of being able to amortize, i.e., *learn*, shape completion under weak supervision.

Finally, we consider the qualitative results on KITTI as presented in Figure 4. As full ground truth shapes are not available, reasoning about qualitative performance is difficult. For example, AML and [13] make similar predictions for the first sample. For the second and third one, however, the predictions differ significantly. Here, we argue that [13] has difficulties predicting reasonably sized cars while AML is not able to recover details such as wheels. We also notice, that Sup is clearly biased towards very thin cars not matching the observed points. Overall, we find it difficult to judge shape completion on KITTI – which motivated the creation of SN-clean and SN-noisy; both [13] and AML are able to predict reasonable shapes.

4.5.3 Shape Completion on ModelNet

On ModelNet, we compare AML and Sup against the VAE shape prior (note that [13] is not applicable), considering both category-specific and -agnostic models, see Table 2. As on SN-clean, AML is able to achieve reasonable performance compared to Sup while using 9% or less supervision.

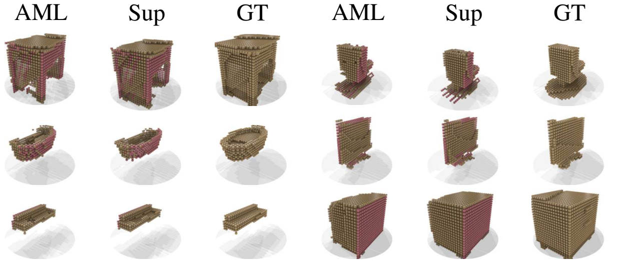

Figure 5: **Qualitative Results on ModelNet.** We present results for AML (category-agnostic, cf. "all" in Table 2) and Sup. We show shapes (occupancy grids, beige) and observations (red).

Additionally, Figure 5 shows that AML is able to distinguish object categories reasonably well without access to category information during training (in contrast to Sup); more results are discussed in the supplementary material.

5. Conclusion

In this paper, we presented a weakly-supervised, learning-based approach to 3D shape completion. After using a variational auto-encoder (VAE) [26] to learn a shape prior on synthetic data, we formulated shape completion as maximum likelihood (ML) problem. We fixed the learned generative model, i.e. the VAE's decoder, and trained a new, deterministic encoder to amortize, i.e. *learn*, the ML problem. This encoder can be trained in an unsupervised fashion. Compared to related data-driven approaches, the proposed amortized maximum likelihood (AML) approach offers fast inference and, in contrast to related learning-based approaches, does not require full supervision.

On newly created, synthetic 3D shape completion benchmarks derived from ShapeNet [5] and ModelNet [49], we demonstrated that AML outperforms a state-of-the-art data-driven method [13] (while significantly reducing runtime) and generalizes across object categories. Motivated by related learning-based approaches, we also compared our approach to a fully-supervised baseline. We showed that AML is able to compete with the fully-supervised model both quantitatively and qualitatively while using 9% or less supervision. On real data from KITTI [18], both AML and [13] predict reasonable shapes. However, AML demonstrates significantly lower runtime, and runtime is independent of the observed points. Additionally, AML allows to learn from KITTI's unlabeled data and, thus, outperforms the fully-supervised baseline which is not able to generalize well. Overall, our experiments demonstrate the benefits of the proposed AML approach: reduced runtime compared to data-driven approaches and training on unlabeled, real data compared to learning-based approaches.

Acknowledgements: We thank Francis Engelmann for help with the approach of [13].

References

[1] M. Abramowitz. *Handbook of Mathematical Functions, With Formulas, Graphs, and Mathematical Tables*. Dover Publications, 1974. 4

[2] S. Agarwal, K. Mierle, and Others. Ceres solver. http://ceres-solver.org, 2012. 6

[3] S. Bao, M. Chandraker, Y. Lin, and S. Savarese. Dense object reconstruction with semantic priors. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2013. 2

[4] A. Brock, T. Lim, J. M. Ritchie, and N. Weston. Generative and discriminative voxel modeling with convolutional neural networks. *arXiv.org*, 1608.04236, 2016. 1

[5] A. X. Chang, T. A. Funkhouser, L. J. Guibas, P. Hanrahan, Q. Huang, Z. Li, S. Savarese, M. Savva, S. Song, H. Su, J. Xiao, L. Yi, and F. Yu. Shapenet: An information-rich 3d model repository. *arXiv.org*, 1512.03012, 2015. 1, 2, 3, 4, 8

[6] X. Chen, K. Kundu, Y. Zhu, H. Ma, S. Fidler, and R. Urtasun. 3d object proposals using stereo imagery for accurate object class detection. *arXiv.org*, 1608.07711, 2016. 5

[7] R. Collobert, K. Kavukcuoglu, and C. Farabet. Torch7: A matlab-like environment for machine learning. In *BigLearn, NIPS Workshop*, 2011. 6

[8] A. Dai, C. R. Qi, and M. Nießner. Shape completion using 3d-encoder-predictor cnns and shape synthesis. *arXiv.org*, abs/1612.00101, 2016. 3, 5, 6

[9] A. Dai, C. R. Qi, and M. Nießner. Shape completion using 3d-encoder-predictor cnns and shape synthesis. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2017. 2

[10] A. Dame, V. Prisacariu, C. Ren, and I. Reid. Dense reconstruction using 3D object shape priors. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2013. 2

[11] D. Eigen and R. Fergus. Predicting depth, surface normals and semantic labels with a common multi-scale convolutional architecture. In *Proc. of the IEEE International Conf. on Computer Vision (ICCV)*, pages 2650–2658, 2015. 5

[12] D. Eigen, C. Puhrsch, and R. Fergus. Depth map prediction from a single image using a multi-scale deep network. In *Advances in Neural Information Processing Systems (NIPS)*, 2014. 5

[13] F. Engelmann, J. Stückler, and B. Leibe. Joint object pose estimation and shape reconstruction in urban street scenes using 3D shape priors. In *Proc. of the German Conference on Pattern Recognition (GCPR)*, 2016. 2, 5, 6, 7, 8

[14] F. Engelmann, J. Stückler, and B. Leibe. SAMP: shape and motion priors for 4d vehicle reconstruction. In *Proc. of the IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 400–408, 2017. 2

[15] H. Fan, H. Su, and L. J. Guibas. A point set generation network for 3d object reconstruction from a single image. *arXiv.org*, abs/1612.00603, 2016. 2, 3, 6

[16] M. Firman, O. Mac Aodha, S. Julier, and G. J. Brostow. Structured prediction of unobserved voxels from a single depth image. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2016. 3

[17] Y. Furukawa and C. Hernandez. Multi-view stereo: A tutorial. *Foundations and Trends in Computer Graphics and Vision*, 9(1-2):1–148, 2013. 1

[18] A. Geiger, P. Lenz, and R. Urtasun. Are we ready for autonomous driving? The KITTI vision benchmark suite. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2012. 1, 2, 3, 4, 6, 8

[19] S. Gershman and N. D. Goodman. Amortized inference in probabilistic reasoning. In *Proc. of the Annual Meeting of the Cognitive Science Society*, 2014. 2, 3, 4

[20] R. Girdhar, D. F. Fouhey, M. Rodriguez, and A. Gupta. Learning a predictable and generative vector representation for objects. In *Proc. of the European Conf. on Computer Vision (ECCV)*, 2016. 1

[21] S. Gupta, P. A. Arbeláez, R. B. Girshick, and J. Malik. Aligning 3D models to RGB-D images of cluttered scenes. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2015. 2

[22] F. Güney and A. Geiger. Displets: Resolving stereo ambiguities using object knowledge. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2015. 2, 5

[23] X. Han, Z. Li, H. Huang, E. Kalogerakis, and Y. Yu. High-resolution shape completion using deep neural networks for global structure and local geometry inference. In *Proc. of the IEEE International Conf. on Computer Vision (ICCV)*, pages 85–93, 2017. 3, 5, 6

[24] C. Häne, S. Tulsiani, and J. Malik. Hierarchical surface prediction for 3d object reconstruction. *arXiv.org*, 1704.00710, 2017. 3

[25] R. R. Jensen, A. L. Dahl, G. Vogiatzis, E. Tola, and H. Aanæs. Large scale multi-view stereopsis evaluation. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2014. 6

[26] D. P. Kingma and M. Welling. Auto-encoding variational bayes. *arXiv.org*, abs/1312.6114, 2013. 2, 3, 4, 5, 8

[27] O. Kroemer, H. B. Amor, M. Ewerton, and J. Peters. Point cloud completion using extrusions. In *IEEE-RAS International Conference on Humanoid Robots (Humanoids)*, pages 680–685, 2012. 2

[28] I. Laina, C. Rupprecht, V. Belagiannis, F. Tombari, and N. Navab. Deeper depth prediction with fully convolutional residual networks. In *Proc. of the International Conf. on 3D Vision (3DV)*, pages 239–248, 2016. 5

[29] A. J. Law and D. G. Aliaga. Single viewpoint model completion of symmetric objects for digital inspection. *Computer Vision and Image Understanding (CVIU)*, 115(5):603–610, 2011. 2

[30] Y. Li, A. Dai, L. J. Guibas, and M. Nießner. Database-assisted object retrieval for real-time 3d reconstruction. *Computer Graphics Forum*, 34(2):435–446, 2015. 2

[31] M. Menze and A. Geiger. Object scene flow for autonomous vehicles. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2015. 3

[32] L. Nan, K. Xie, and A. Sharf. A search-classify approach for cluttered indoor scene understanding. *ACM TG*, 31(6):137:1–137:10, Nov. 2012. 2

[33] M. Pauly, N. J. Mitra, J. Giesen, M. H. Gross, and L. J. Guibas. Example-based 3d scan completion. In *Eurographics Symposium on Geometry Processing (SGP)*, 2005. 2

[34] M. Pauly, N. J. Mitra, J. Wallner, H. Pottmann, and L. J. Guibas. Discovering structural regularity in 3d geometry. *ACM Trans. on Graphics*, 27(3):43:1–43:11, 2008. 2

[35] Z. Pizlo. Human perception of 3d shapes. In *Proc. of the International Conf. on Computer Analysis of Images and Patterns (CAIP)*, pages 1–12, 2007. 1

[36] Z. Pizlo. *3D shape: Its unique place in visual perception*. MIT Press, 2010. 1

[37] D. J. Rezende, S. M. A. Eslami, S. Mohamed, P. Battaglia, M. Jaderberg, and N. Heess. Unsupervised learning of 3d structure from images. *arXiv.org*, 1607.00662, 2016. 2, 3

[38] D. J. Rezende and S. Mohamed. Variational inference with normalizing flows. In *Proc. of the International Conf. on Machine learning (ICML)*, pages 1530–1538, 2015. 3

[39] G. Riegler, A. O. Ulusoy, H. Bischof, and A. Geiger. OctNetFusion: Learning depth fusion from data. In *Proc. of the International Conf. on 3D Vision (3DV)*, 2017. 2, 3, 5, 6

[40] D. Ritchie, P. Horsfall, and N. D. Goodman. Deep amortized inference for probabilistic programs. *arXiv.org*, abs/1610.05735, 2016. 3

[41] A. Sharma, O. Grau, and M. Fritz. Vconv-dae: Deep volumetric shape learning without object labels. *arXiv.org*, 1604.03755, 2016. 1, 2, 3, 5, 6

[42] E. Smith and D. Meger. Improved adversarial systems for 3d object generation and reconstruction. *arXiv.org*, abs/1707.09557, 2017. 2, 3, 6

[43] F. Steinbrucker, C. Kerl, and D. Cremers. Large-scale multi-resolution surface reconstruction from rgb-d sequences. In *Proc. of the IEEE International Conf. on Computer Vision (ICCV)*, 2013. 4

[44] M. Sung, V. G. Kim, R. Angst, and L. J. Guibas. Data-driven structural priors for shape completion. *ACM Trans. on Graphics*, 34(6):175:1–175:11, 2015. 2

[45] M. Tatarchenko, A. Dosovitskiy, and T. Brox. Octree generating networks: Efficient convolutional architectures for high-resolution 3d outputs. In *Proc. of the IEEE International Conf. on Computer Vision (ICCV)*, 2017. 3

[46] S. Thrun and B. Wegbreit. Shape from symmetry. In *Proc. of the IEEE International Conf. on Computer Vision (ICCV)*, pages 1824–1831, 2005. 2

[47] D. Wang and Q. Liu. Learning to draw samples: With application to amortized MLE for generative adversarial learning. *arXiv.org*, abs/1611.01722, 2016. 3

[48] J. Wu, C. Zhang, T. Xue, B. Freeman, and J. Tenenbaum. Learning a probabilistic latent space of object shapes via 3d generative-adversarial modeling. In *Advances in Neural Information Processing Systems (NIPS)*, pages 82–90, 2016. 1

[49] Z. Wu, S. Song, A. Khosla, F. Yu, L. Zhang, X. Tang, and J. Xiao. 3d shapenets: A deep representation for volumetric shapes. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2015. 1, 3, 5, 8

[50] J. Xie, M. Kiefel, M.-T. Sun, and A. Geiger. Semantic instance annotation of street scenes by 3d to 2d label transfer. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2016. 3

[51] Q. Zheng, A. Sharf, G. Wan, Y. Li, N. J. Mitra, D. Cohen-Or, and B. Chen. Non-local scan consolidation for 3d urban scenes. *ACM Trans. on Graphics*, 29(4):94:1–94:9, 2010. 2