Weakly-supervised 3D Shape Completion in the Wild

Jiayuan Gu[1,2], Wei-Chiu Ma[1,3], Sivabalan Manivasagam[1,4], Wenyuan Zeng[1,4], Zihao Wang[1], Yuwen Xiong[1,4], Hao Su[2], and Raquel Urtasun[1,4]

[1] Uber Advanced Technologies Group
[2] University of California, San Diego
{jigu, haosu}@eng.ucsd.edu
[3] Massachusetts Institute of Techonology
[4] University of Toronto
{wichiu, manivasagam, wenyuan, yuwen, urtasun}@uber.com

Abstract. 3D shape completion for real data is important but challenging, since partial point clouds acquired by real-world sensors are usually sparse, noisy and unaligned. Different from previous methods, we address the problem of learning 3D complete shape from unaligned and real-world partial point clouds. To this end, we propose a weakly-supervised method to estimate both 3D canonical shape and 6-DoF pose for alignment, given multiple partial observations associated with the same instance. The network jointly optimizes canonical shapes and poses with multi-view geometry constraints during training, and can infer the complete shape given a single partial point cloud. Moreover, learned pose estimation can facilitate partial point cloud registration. Experiments on both synthetic and real data show that it is feasible and promising to learn 3D shape completion through large-scale data without shape and pose supervision.

1 Introduction

We are interested in the problem of 3D shape completion, which estimates the complete geometry of objects from partial observations. This task is a prerequisite for many important real-world applications. For example, complete shapes can facilitate automated vehicles to track objects [12] and robots to figure out the best pose to grasp objects [29]. Previous works [8, 14, 33] have successfully applied deep learning methods to learn shape priors from large-scale synthetic data, which results in improvement of the 3D shape completion task. However, most these prior works have two major limitations: 1) they require the ground-truth shape for learning, and 2) they assume the input partial point clouds are aligned and normalized to the canonical frame, in which the object faces forward and are centered at the origin. In addition, models trained on synthetic data do not transfer well to the real world due to the domain gap.

We aim to use real data for the 3D shape completion task. However, since there is a lack of real 3D data that comes with sufficient high-quality ground-truth 3D shapes, we cannot directly adopt these supervised learning methods

developed in the synthetic domain. Although there are a few datasets containing real scans, such as KITTI [11] and ScanNet [7], no efforts are made to explore the possibility of learning 3D shape completion in a weakly-supervised fashion. There are three challenges to work on real 3D data, unique from the synthetic ones: 1) No or few ground-truth complete shapes are available for full supervision. Note that annotating 3D shapes are more difficult and expensive than annotating 2D images; 2) Partial point clouds acquired by real-world sensors like RGB-D cameras or LiDAR are sparse and noisy; 3) Poses and sizes of objects are more diverse, and partial observations may be occluded by other objects.

In this paper, we address the problem of learning 3D shape completion from real, unaligned partial point clouds without shape and pose supervision (Sec 3). The proposed method is weakly-supervised by multi-view consistency of instances (Sec 4). The key contributions of our work are as follows:

1. We propose a weakly-supervised[5] approach to learn 3D shape completion from unaligned point clouds. Our promising results show that it is feasible to learn 3D shape completion from large-scale 3D data without shape and pose supervision.
2. We showcase the extension of our method to tackle the challenging partial point cloud registration problem.

2 Related work

3D reconstruction from single images 3D shape completion is highly related to 3D reconstruction from single images, since a partial point cloud can be obtained from a RGB-D image. Since the problem is ill-posed by nature, many learning-based approaches are developed to learn shape priors from large-scale data. [6] uses a recurrent 3D CNN to predict a 3D occupancy grid given one or more images of an object. [27] proposes a differentiable 'view consistency' loss and a probabilistic occupancy grid. [10] pioneers the representation of point clouds as output. However, they all require full supervision from synthetic images rendered from ShapeNet [4]. Performance on real datasets like Pascal 3D+ [31] suffer from unrealistic ground truth shapes from aligned CAD models.

Thus, [32, 37, 26, 14] focus on reconstructing 3D shapes with weak supervision. Especially, [26] enforces geometric consistency between the independently predicted shape and pose from two views of the same instance. Differential point clouds (DPC) [14] uses a similar strategy to reconstruct point clouds and devises differentiable projection of point clouds. However, it is non-trivial to extend these methods to real-world data, which will be discussed in Sec 5.3.

3D reconstruction from multiple frames By leveraging consecutive frames, 3D shapes can be reconstructed from RGB images [25, 1, 9] or depth images [17]. The problem is also known as Structure-from-Motion (SfM). [28, 35, 23] are proposed to tackle it with deep learning. Although poses are estimated in both SfM

[5] We use the term "weakly-supervised" instead of "unsupervised learning of shape and pose" [14] to avoid confusion, which are in fact equivalent.

and our 3D shape completion, the main difference is that unseen 3D points are hallucinated in our 3D shape completion while depths are estimated in the SfM.

KinectFusion [17] fuses all the depth data streamed from a Kinect sensor into a single global implicit surface model of the observed scene in real-time. It demonstrates the advantages of maintaining a full surface model compared to frame-to-frame tracking. Our method benefits from the similar idea, but differs from it in 2 aspects: 1) The proposed approach is a learning-based framework based on 3D point clouds only. 2) The trained model can predict the complete shape from a single point cloud and the relative pose between two distant views during inference, which is demonstrated in our experiments.

3D shape completion 3D shape completion is usually performed on partial scans of individual objects. With the success of deep learning, learning-based approaches show more flexibility and better performance compared with geometry-based and alignment-based methods. [8] combines a data-driven shape predictor and analytic 3D shape synthesis. [33] proposes a variant of PointNet [19] to directly process point clouds and generate high-resolution outputs. [24] devises a tree-style neural network to generate structured point clouds.

3D shape completion without full supervision is of increasing interest to the community. [22] finetunes the encoder on the target dataset, like KITTI [11], with a fixed generator pretrained on the ground truth SDF representation of synthetic data, like ShapeNet [4]. [13] generates half-to-half sequence pairs from the ground truth complete point clouds of ShapeNet, and learns features by half-to-half prediction and self-reconstruction. [5] trains autoencoders to learn embedding features of shape on clean, complete synthetic data and noisy, partial target data. An adaption network is learned to transform the embedding space of noisy point clouds to that of clean point clouds with a GAN setup. However, none of those works deals with unaligned point clouds and relies on complete synthetic data to pretrain.

Deep learning for point clouds PointNet [19] is the pioneer to directly process point clouds with a deep neural network, followed by many variants [20, 30, 15]. It extracts features for each point with a shared multi-layer perceptron (MLP), and outputs with an aggregation function invariant to permutation. Any point-cloud-based neural network can work as the encoder of our method.

3 Problem

The goal of 3D shape completion is to predict a complete shape Y given a partial observation X. In this work, we represent the partial observation and the complete shape as point clouds: $X \in \mathcal{R}^{n\times 3}$ and $Y \in \mathcal{R}^{m\times 3}$, where n and m are the number of partial and complete points respectively.

Previous approaches [33, 22] have assumed that partial observations are normalized according to ground-truth bounding boxes and transformed into a predefined canonical frame, *e.g.*, the forward-facing object centered at the origin. Past works may also assume the ground-truth shape $Y^{gt} \in \mathcal{R}^{m_{gt}\times 3}$ is available and train a model in a supervised setting via a permutation-invariant loss

function $L(Y, Y^{gt}; X)$, such as Chamfer Distance (CD) or Earth Mover Distance (EMD) [10] to evaluate reconstruction quality. While these ground truth information may be available on synthetic data, they may not be available in the real-world setting. Thus, we build on past works and propose a more general and challenging setting:

- We do not assume knowledge of the ground-truth canonical frame for normalizing and aligning the partial observations. Instead, we maintain the partial observations in the *sensor* coordinate system.
- We do not assume knowledge of the ground-truth shape.
- The point cloud observation is captured by a sensor (*e.g.*, a LiDAR) from the real world and can therefore be sparse and also noisy.
- Instead of ground truth, we have access to a set of unaligned partial observations of the instance, captured at different viewpoints by the sensor.

We call this more realistic setting "weakly-supervised shape completion in the wild". This setting is especially applicable to the real-world setting such as in autonomous driving or indoor scene navigation, where the robot may observe other moving agents from multiple viewpoints and needs to reason about the shape and pose to perform shape completion. In the next section, we propose our method for tackling weakly-supervised shape completion in the wild.

4 Method

4.1 Overview

We tackle weakly-supervised shape completion in the wild by jointly learning the canonical shape and pose of the object. The underlying idea is that predicting a complete shape Y_{sen} in the *sensor* coordinate system is equivalent to predicting a complete shape Y_{can} in the *canonical* coordinate system[6] and then transforming it according to a 6-DoF pose T_{can}^{sen}, where $Y_{sen} = T_{can}^{sen} Y_{can}$. But a key question remains: how do we learn the complete shape and pose when we do not have access to the ground-truth for either? We leverage the fact that, during training, we have access to multiple observations of the object from different viewpoints. We know that these observations, while noisy, accurately represent different views of the GT shape. By enforcing that predicted shapes and poses are consistent with recorded observations, we can train the network in a weakly-supervised fashion to estimate both the shape and pose from a single observation.

Our training approach is as follows: Given a set of sensor observations of the object of interest, $\{X_{sen}^1, X_{sen}^2, \cdots, X_{sen}^M\}$, we apply a deep autoencoder network to each observation X_{sen}^i and predict a canonical shape Y_{can} and pose $T_{can}^{sen_i}$. We then apply two loss terms based on these outputs to guide the network to learn the correct shape and pose: (1) the partial observation points should match the completed shape transformed by the estimated pose (observation-matching-shape), and (2) the surface points of the completed shape as viewed by the

[6] The *canonical* frame in our method is not predefined, but emerges during training.

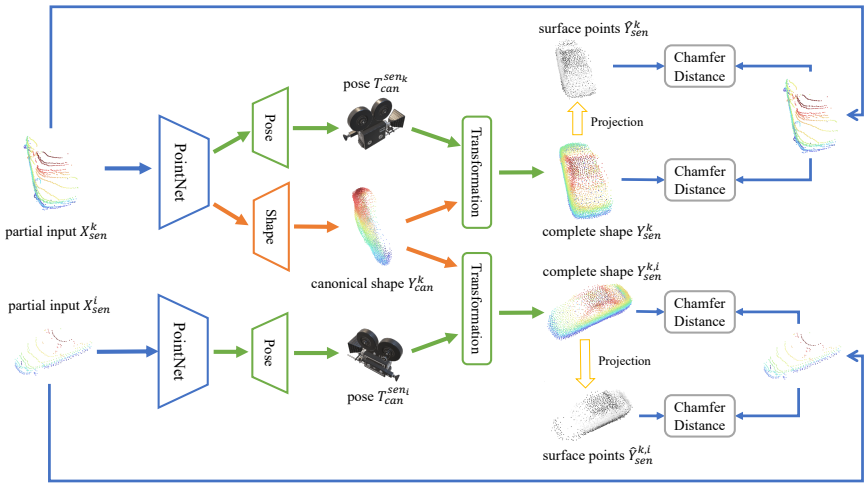

Fig. 1: Overview of our weakly-supervised shape completion pipeline at training time. This illustration is for a pair of partial inputs and can be extended to partial input of a shape from multiple views, by optimizing the averaged loss among selected pairs of inputs.

estimated sensor pose should match the observation points with self-occlusion taken into consideration (shape-projection-matching-observation). Because we have access to multiple observations and multiple pose predictions, we can encourage the network to generate a completed shape estimate that minimizes these two loss terms with respect to all observations and poses. We call this *multi-view consistency*.

During inference, the trained network takes as input a single partial point cloud, and outputs the estimated pose and completed shape. Our pipeline is illustrated in Fig 1. The following sections describe in detail our approach. Sec 4.2 describes the network architecture to predict 3D canonical shapes and 6-DoF poses. Sec 4.3 and Sec 4.4 present the loss terms (observation-matching-shape and shape-silhouette-matching-observation). Sec 4.5 describes how we extend the two loss terms to work on multiple observations.

4.2 Predict canonical shape and pose

Given a partial observation X_{sen}, we employ a deep encoder-decoder network to predict both the 3D canonical shape and 6-DoF pose. The encoder encodes the input X_{sen} to a latent code z. We use the same encoder architecture as PCN's [33], which is a variant of PointNet [19]. The shape decoder is a 3-layer MLP, which decodes z to a fixed number of 3D coordinates $Y_{can} \in R^{m\times 3}$. The pose decoder, also a 3-layer MLP, outputs a rotation $\hat{R}$ and a translation t. Following [36],the rotation is represented as a 6D vector, and the translation a

3D vector. The inferred rotation matrix and translation form T_{can}^{sen}. Thus, the predicted complete shape in the *sensor* coordinate system is calculated as:

$$Y_{sen} = T_{can}^{sen} Y_{can} = f_T(z) f_{shape}(z) \tag{1}$$

To alleviate the issue of local minima and overcome bad initialization, we follow prior art and have multiple pose decoder branches in our network and train them with the hindsight loss introduced in DPC [14]. In brief, hindsight loss is where, for each batch, gradients are only backpropagated to the branch with the lowest loss.

4.3 Match partial observation with canonical shape

We now describe the observation-matching-shape loss, which we implement as an asymmetric Chamfer-Distance (CD) between the observation point cloud and the completed shape point cloud in the sensor-coordinate space. The asymmetric CD (Eq 2) between the input observation X_{sen} and the output shape Y_{sen} is

$$CD(X_{sen} \mapsto Y_{sen}) = \frac{1}{|X_{sen}|} \sum_{x \in X_{sen}} \min_{y \in Y_{sen}} ||x - y||_2 \tag{2}$$

This forces the output canonical shape to completely cover the input observation. However, it does not guarantee that Y_{sen} is close to X_{sen} — even a point cloud that fills the whole 3D space would minimize Eq 2, which is not desired. Thus, we need to develop a more sophisticated loss term to enforce how the sensor acquires the point cloud and compute the distance between the input observation and the projection of the output, which is described next.

4.4 Project canonical shape to partial observation

We now describe the shape-projection-matching-observation term. Using our knowledge of how the sensor acquires observations, we can "simulate" which points on the surface are observed based on the estimated complete shape point cloud and the estimated pose. We can then force the "generated" point cloud to match the true observation. We tailor this loss term based on knowledge of how the LiDAR sensor works.

Given a subset point cloud $\hat{Y}_{sen}$ of the predicted point cloud Y_{sen}, which are on the surface as viewed from the sensor (the "simulated" observation), another asymmetric CD (Eq 3) between the input X_{sen} and those *surface* points is optimized.

$$CD(\hat{Y}_{sen} \mapsto X_{sen}) = \frac{1}{|\hat{Y}_{sen}|} \sum_{\hat{y} \in \hat{Y}_{sen}} \min_{x \in X_{sen}} ||\hat{y} - x||_2 \tag{3}$$

We introduce a simple, flexible and efficient way to infer the *surface* points. The point cloud acquired by the LiDAR sensor can be projected to a range

image, which is essentially the polar-coordinate system of the LiDAR sensor. The polar coordinate (ϕ, θ, r) of a cartesian sensor observation point (x, y, z) is calculated as Eq 4:

$$r = \sqrt{x^2 + y^2 + z^2}, \phi = tan^{-1}\left(\frac{x}{y}\right), \theta = sin^{-1}\left(\frac{z}{r}\right) \tag{4}$$

where r is radial distance to the sensor and ϕ, θ are the azimuth and pitch angles, respectively, of the ray shot from the LiDAR sensor. According to the resolution of LiDAR (d_ϕ, d_θ), we can discretize (ϕ, θ) to $(\lfloor \frac{\phi}{d_\phi} \rfloor, \lfloor \frac{\theta}{d_\theta} \rfloor)$, which forms several bins. For each bin, the point with the smallest distance is considered to be on the *surface*. This is the depth buffer approach widely used for rasterization in the computer graphics literature.

This "projection" approach is differentiable and simple to implement, since we can just count the occupied bins and find the smallest distance in each. No voxelization or normalization of the points is needed like in DPC [14], which makes our approach more flexible, especially for real data without a normalized scale. Additionally, we can flexibly adjust the projection resolution to be coarser than the real resolution, which helps with noisy and occluded real-world data. Furthermore, this method is also efficient as the computation complexity is $O(m)$, where m is the number of predicted points.

4.5 Multi-view consistency

Both loss terms, observation-matching-shape and shape-projection-matching-observation, work not only for the input observation X^i_{sen}, but also works for all other observations of the instance in the set. Inspired by [26, 14], we leverage the consistency among multiple views associated with the same instance to supervise 3D shape prediction and 6-DoF pose estimation. During training, we sample M observations $\{X^1_{sen}, X^2_{sen}, \cdots, X^M_{sen}\}$ of one instance within a batch. One view is selected as the *reference*, denoted by index k. Intuitively, all observations share the same complete rigid shape in the *canonical* coordinate system. In other words, for any view i, Y^i_{can} should be close to Y^k_{can}. Therefore, we can replace Y^i_{sen} with $Y^{k,i}_{sen} = Y^k_{can}\hat{R}^T_i + t_i$, which forces the network to learn a complete canonical shape matching all the partial views.

The full loss for a given training example $\{X^i_{sen}, i \in 1...N\}$ with reference view k is calculated as Eq. 5:

$$L(\{X^i_{sen}\}) = \sum_{i=1}^{M} CD(X^i_{sen} \mapsto Y^{k,i}_{sen}) + \beta CD(\hat{Y}^{k,i}_{sen} \mapsto X^i_{sen}) \tag{5}$$

where β is a hyper-parameter, which can be adjusted according to the quality of data. While we could apply multi-view consistency between all possible pairs (*i.e.*, make each index in the observation set the reference index and sum all terms), we choose one randomly to reduce training complexity.

5 Experiments

In this section, we demonstrate our method and several baselines on this new setting of weakly-supervised shape completion in the wild. We first evaluate our method on the standard synthetic data benchmark, ShapeNet. We then showcase the performance of our method on two real-world self-driving datasets for which we construct ground-truth complete shapes. Furthermore, we demonstrate our method also works on the task of point cloud registration. Finally, we compare our approach against a fully-supervised oracle.

5.1 Datasets

ShapeNet [4] ShapeNet is a richly-annotated, large-scale dataset of 3D synthetic shapes. We focus on 3 categories: chairs, cars, and airplanes. We use the same data and split provided by DPC [14], where the data available for each training example is 5 random RGB-D views of the model. We note that this data only has random viewpoint/orientation, and the translation component of the view is fixed. To acquire partial point clouds in the camera coordinate system, we backproject depth maps according to the intrinsic matrix. The average number of points of the partial point clouds for chairs, cars and airplanes is 3018, 2956, 756 respectively. For evaluation, 8192 ground-truth points are sampled from the surface of the CAD models.

3D vehicle dataset [16] We build a collection of complete vehicle object point clouds from a large-scale LiDAR dataset for self-driving that contain bounding box instance annotations for over 1.2 million frames. We generate the ground-truth complete shape as follows: for each static object, we accumulate the LiDAR points inside the bounding box and determine the object relative coordinates for the LiDAR points based on the bounding box center. Since cars are usually symmetric, we postprocess data by mirroring the aggregate point cloud along the vehicle's heading axis, followed by Gaussian statistical outlier removal, to acquire complete shapes for annotated objects. Visualizations of the ground-truth shape can be seen in Fig 3. There are 13700 annotated objects in total, splitted into 10000/700/3000 for train/val/test. On average, each object is associated with 80 scans, and each scan contains 1163 points. We filter observations to include at least 100 points to avoid overly sparse observations.

SemanticKITTI [2] Instance and semantic annotations for the LiDAR point clouds are provided for all sequences of the Odometry Benchmark. We use SemanticKITTI's odometry localization to aggregate partial point clouds of the same parked vehicle instance (with at least 512 points on average) into a single vehicle frame and apply radius outlier removal. Following [2], we train our network on instances generated from sequences 00 to 10, except for sequence 08 instances which are used as test set. There are 659/229 instances and 51186/16299 observations for training/test. On average, each object is associated with 95 scans, and each scan contains 1377 points.

5.2 Tasks and metrics

3D shape completion For shape completion, the algorithm is required to predict shapes in the sensor coordinate system. Given the ground-truth canonical shape and pose, we can compute the ground-truth shape in the sensor coordinate system. Then, we adopt the standard metrics used in the literature [10, 33, 14, 26]. The main metric to evaluate shape completion against ground truth point cloud Y^{gt}_{sen} is the Chamfer Distance (Eq 6). The first term is called the *Precision* and the second term is called the *Coverage*.

$$\begin{aligned} CD(\hat{Y}_{sen} \leftrightarrow Y^{gt}_{sen}) = & \frac{1}{|\hat{Y}_{sen}|} \sum_{\hat{y} \in \hat{Y}_{sen}} \min_{y \in Y_{sen}} ||\hat{y} - y^{gt}||_2 \\ & + \frac{1}{|Y^{gt}_{sen}|} \sum_{y^{gt} \in Y^{gt}_{sen}} \min_{\hat{y} \in \hat{Y}_{sen}} ||y^{gt} - \hat{y}||_2 \end{aligned} \tag{6}$$

Partial point cloud registration Given two partial observations, the algorithm is required to predict the relative pose from one to the other. This task is more challenging than common point cloud registration. The algorithms are evaluated by calculating the quaternion distance θ, or *angle difference*, between the estimated pose q_{pred} and the GT pose q_{gt} for all instances in the dataset: $\theta = 2 \arccos\langle q_{pred}, q_{gt} \rangle$. Following DPC [14], we report the median of angle difference and accuracy (the percentage of samples for which $\theta \leq 30°$). In addition, if the translation is predicted, we also report the median of mean-square-error between the prediction and the ground truth.

5.3 Baselines

To our knowledge, there are currently no weakly-supervised methods for shape completion that take as input a single unaligned partial point cloud. We instead compare our method against the state-of-the-art single-image 3D reconstruction method, and standard point cloud alignment methods.

DPC [14] DPC is a weakly-supervised method, which is trained on image pairs to reconstruct 3D point clouds. We compare to their reported results of shape reconstruction on ShapeNet, and adapt their method to range images. We argue that it is non-trivial to adapt DPC to the "wild" setting, where both pose rotation and translation are unknown and the shape size is not bounded. We list their drawbacks as follows: 1) It is assumed that canonical shapes are normalized into a unit cube; 2) A fixed camera distance to the object is provided and only rotations are considered in the original paper; 3) The projection loss only is sensitive to density and occlusion. Despite these drawbacks, we modify DPC by replacing perspective transformation with polar transformation and scaling the normalized output by a fixed factor to match the real scale, denoted by DPC-LIDAR. We refer readers to the supplementary for more details.

ICP Since our ground-truth complete shapes are acquired by accumulating partial point clouds given ground-truth transformations, we introduce a baseline

based on Iterative Closest Point (ICP). For two consecutive frames, we calculate the rigid transformation between two partial point clouds by ICP. Thus, given a pair of partial point clouds, the transformation can be calculated by accumulating results from ICP. All the partial point clouds can be transformed into a certain frame and fused. To reduce the accumulated error, we choose the middle frame as the reference. For 3D shape completion, the fused point cloud in the reference frame is transformed according to the ground-truth pose to compare with the ground-truth complete shape. For point cloud registration, we compare the transformation from one frame to the reference one with the ground-truth transformation. Local ICP [3] and Global ICP [21] are the two ICP algorithms we compare against. We use the implementation of Open3D [34] and search the best hyper-parameters on the validation set (0.175 for the distance threshold in Local ICP and 0.125 for the voxel size in Global ICP).

5.4 Implementation details

As our synthetic and real datasets are of different sizes and come very different distributions, we slightly modify our data input and our implementation of the model for each setting:

Input To ease the learning requirements for our model, we preprocess input partial point clouds. Given a partial point cloud in the sensor coordinate system, we shift the point cloud to be centered at the origin without knowledge of the ground-truth shape or pose: For ShapeNet, an axis-aligned bounding box in the camera coordinate system is calculated, and its center is shifted to the origin; for real data, we first extract a bounding frustum of the input partial point cloud, and then centralize the frustum[7]. After converting to this origin-shifted coordinate system, the input point cloud is resampled with replacement to a fixed size, as done in the original PointNet [19]. For ShapeNet and for real LiDAR datasets, we uniformly resample 3096 and 1024 points, respectively, from the input partial point cloud.

Training During training, 4 observations per instance are sampled in a batch. Adam is used as the optimizer. For synthetic data, models are trained with an initial learning rate of $1e^{-4}$ for 300k iterations and a batch size of 32. The learning rate is decayed by 0.5 every 100K iterations. For real data, models are trained with an initial learning rate of $1e^{-4}$ for 400k iterations and a batch size of 32. The learning rate is decayed by 0.7 every 100K iterations. Especially, all the observations of one instance in a batch are within a window of 20 frames. It takes less than 16 hours to train our model with a GTX 1080Ti. The loss weight β is set to 0.25 and 0.05 for synthetic and real data respectively.

5.5 Results of 3D shape completion

ShapeNet We first demonstrate shape completion results on ShapeNet. The chamfer distance, precision and coverage are reported on the test set. We also

[7] The resulting coordinate system is similar to *3D mask coordinate* introduced in [18].

	CD	Precision	Coverage
DPC	6.26/3.54/4.85	4.19/1.95/2.59	2.07/1.59/2.26
DPC†	13.17/4.73/7.32	8.17/2.52/3.81	5.00/2.20/3.52
DPC (pre-aligned [14])	3.91/3.47/4.30	-	-
DPC† (pre-aligned [14])	5.07/4.09/5.86	-	-
Ours	**1.95/2.68/3.33**	**0.91/1.27/1.69**	**1.05/1.41/1.64**

Table 1: Quantitative results of 3D shape completion on the test set of ShapeNet (airplane/car/chair). All the values are multiplied by 100. We also report the original numbers from [14]. Note that they align predicted shapes according to the validation set before evaluating on the test set.

include the chamfer distance of DPC reported by [14]. Besides, we compare our approach with DPC†, which predicts both rotation and translation of the pose. Note that [14] evaluates *shape prediction*, rather than *shape completion*, by aligning the predicted canonical shape according to the ground truth of the validation set. We argue that this evaluation protocol assumes that all the objects share the same canonical space and it does not really disentangle shape and pose.

Table 1 shows the quantitative comparison between our method and the DPC variants. Despite not having access to the ground-truth translation or size of the object, our model is able to predict a more accurate complete shape. Fig 2 shows the qualitative results. Since planes are usually flat and result in sparse observations, DPC fails to learn a clean shape while our approach is more robust. We refer readers to the supplementary for ablation studies and more details.

Real LiDAR datasets We now apply our method to real-world partial LiDAR scans of vehicles. Table 2a and 2b show the comparison between our method and ICP baselines on real LiDAR datasets. Fig 3 showcases the qualitative results. DPC-LIDAR does not converge and performs much worser than our approach, which implies it is better to process point clouds directly rather than project them into 2D planes and rely on existing 2D methods. Moreover, compared to strong ICP baselines, our method shows higher precision and comparable coverage. More results and analysis are provided in the supplementary.

	CD	Precision	Coverage
DPC-LIDAR	0.928	0.489	0.439
Local-ICP	0.315	0.170	0.145
Global-ICP	0.309	0.174	**0.135**
Ours	**0.255**	**0.083**	0.172

(a) 3D vehicle dataset

	CD	Precision	Coverage
Local-ICP	0.246	0.152	0.094
Global-ICP	0.213	0.138	**0.075**
Ours	**0.194**	**0.087**	0.107

(b) SemanticKITTI

Table 2: 3D shape completion results on the test sets of real LiDAR datasets.

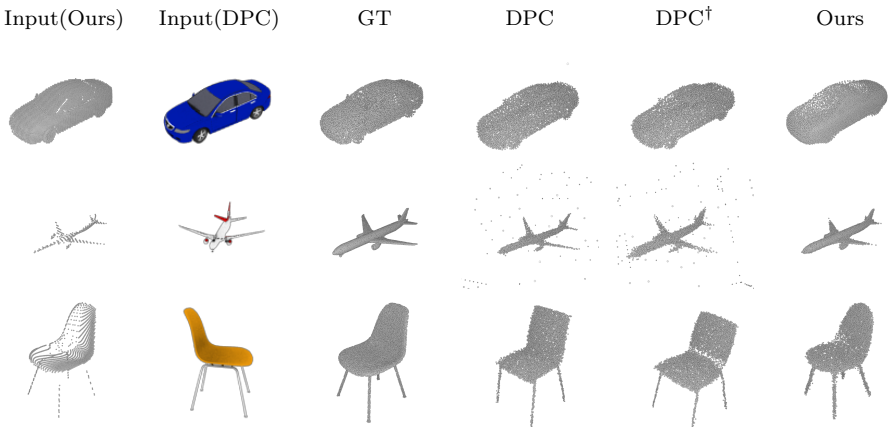

Fig. 2: Qualitative results of 3D shape completion on the test set of ShapeNet. All the (input and predicted) point clouds are transformed to the ground-truth canonical frame and visualized at a fixed viewpoint. *Note that the input of our method is the point cloud lifted from the depth map input of DPC.*

5.6 Partial point cloud registration

In this section, we showcase that our method can be applied to point cloud registration. It is challenging to align real-world partial point clouds for traditional methods like ICP, due to incompleteness, noise, and sparsity. Furthermore, even if the transformation between two consecutive frames is accurate, the error may accumulate across frames. However, our approach alleviates these issues since it implicitly encodes a complete canonical shape. To evaluate the performance of point cloud registration, we select the middle frame in a sequence as the target, and align other frames to the reference according to estimated poses. Given a source and a target point cloud, our method predicts T_{can}^{src} and T_{can}^{tgt}. Thus, the transformation from the source to the target is calculated as $T_{src}^{tgt} = T_{can}^{tgt}(T_{can}^{src})^{-1}$. We report the accuracy, median angle difference, and median MSE between the groundtruth vs. our method, Local-ICP, Global-ICP in Table 3a and 3b. Fig 3 demonstrates fused point clouds according to estimated poses. Our method outperforms standard alignment methods.

5.7 Comparison with fully-supervised counterparts

Our method does not rely on any ground-truth shape and pose, or any prior knowledge of where the object is located, *e.g.*, the bounding box. Yet, it can still reconstruct the 3D shape reliably and accurately. In order to understand the upper bound of our method for shape completion, we include an *oracle* baseline, where our model is trained with ground-truth complete shapes. Concretely, during training, given a partial point cloud in the *sensor* coordinate system as

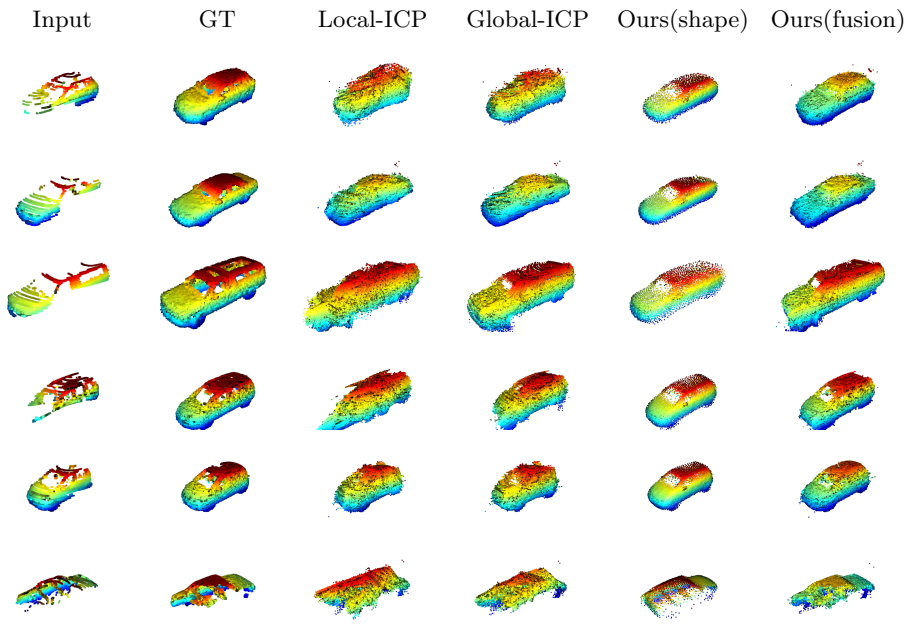

Fig. 3: Qualitative results of our method compared against ground-truth and ICP on the real datasets (row 1-3: 3D vehicle dataset; row 4-6: SemanticKITTI). All the point clouds are transformed to the ground-truth canonical frame and visualized at a fixed viewpoint. We denote our approach for 3D shape completion and point cloud registration by *Ours(shape)* and *Ours(fusion)*.

	Acc	Rot $\Delta\theta$	Trans Δt
Local-ICP	84.09	11.33	0.30
Global-ICP	83.83	10.69	0.26
Ours	**97.68**	**2.37**	**0.13**

(a) 3D vehicle dataset

	Acc	Rot $\Delta\theta$	Trans Δt
Local-ICP	85.29	13.04	0.31
Global-ICP	85.28	10.59	0.23
Ours	**89.37**	**2.86**	**0.17**

(b) SemanticKITTI

Table 3: Point cloud registration results on the test sets of real LiDAR datasets. We report the median of angle difference and accuracy (the percentage of samples for which $\Delta\theta \leq 30°$, as well as the median of translation error Δt.)

input, we employ the same network to encode the input and decode the canonical complete shape without estimating the pose. The Chamfer Distance is calculated between the output canonical shape and the ground-truth canonical complete shape. Note that our method with full supervision is identical to *PCN-FC* [33], except for unaligned point clouds as input.

Category	Method	CD	Precision	Coverage
Airplane	Ours	1.95	0.91	1.05
	Ours(Full-Sup)	1.65	0.77	0.88
Car	Ours	2.68	1.27	1.41
	Ours(Full-Sup)	2.04	1.01	1.03
Chair	Ours	3.33	1.69	1.64
	Ours(Full-Sup)	2.82	1.47	1.35
Real vehicle	Ours	0.255	0.083	0.172
	Ours (Full-Sup)	0.140	0.064	0.077

Table 4: Shape completion results on the test sets of ShapeNet and our 3D vehicle dataset. All the values for ShapeNet categories are multiplied by 100. The full-supervision oracle is denoted by *Ours(Full-Sup)*.

Table 4 shows that there exists a gap between our weakly-supervised approach and its fully-supervised counterpart. However, the gap is even smaller than that between ours and DPC [14]. The Chamfer Distance of the fully-supervised oracle is almost half of that of our weakly-supervised approach. This may be due to the fact that the LiDAR sensor will only see half of the car by which it passes, and therefore the partial LiDAR observations alone are insufficient to see the other side of the car. To solve this issue, prior knowledge of the category may help, which we leave for future work.

6 Discussion and Future Work

We have proposed a new setting, weakly-supervised 3D shape completion in the wild, which better captures the realistic scenario of being able to infer unknown shape from real world scans of objects. We demonstrate that this challenging problem can be tackled by jointly learning both shape and pose with multi-view consistency. However, there remains much space to improve and explore. From visualization, we observe that the model tends to generate coarse shapes and miss details, due to the noise of pose estimation. It is also observed in training that the loss calculated on point clouds is more sensitive to the density compared to that using 2D projection. More efforts can be made to improve visual quality and narrow the gap with fully-supervised methods. Besides, we use PointNet as the backbone for simplicity and efficiency in this work. Differently designed networks can be applied to predict shapes and poses separately. Furthermore, our approach currently requires knowing multiple views of a single rigid object. Such "annotations" can be acquired by a 3D detector and tracker. Thus, one future direction is to study self-supervised or weakly-supervised 3D detection and tracking.

References

1. Agarwal, S., Snavely, N., Seitz, S.M., Szeliski, R.: Bundle adjustment in the large. In: European conference on computer vision. pp. 29–42. Springer (2010)
2. Behley, J., Garbade, M., Milioto, A., Quenzel, J., Behnke, S., Stachniss, C., Gall, J.: SemanticKITTI: A Dataset for Semantic Scene Understanding of LiDAR Sequences. In: Proc. of the IEEE/CVF International Conf. on Computer Vision (ICCV) (2019)
3. Besl, P.J., McKay, N.D.: Method for registration of 3-d shapes. In: Sensor fusion IV: control paradigms and data structures. vol. 1611, pp. 586–606. International Society for Optics and Photonics (1992)
4. Chang, A.X., Funkhouser, T., Guibas, L., Hanrahan, P., Huang, Q., Li, Z., Savarese, S., Savva, M., Song, S., Su, H., et al.: Shapenet: An information-rich 3d model repository. arXiv preprint arXiv:1512.03012 (2015)
5. Chen, X., Chen, B., Mitra, N.J.: Unpaired point cloud completion on real scans using adversarial training. arXiv preprint arXiv:1904.00069 (2019)
6. Choy, C.B., Xu, D., Gwak, J., Chen, K., Savarese, S.: 3d-r2n2: A unified approach for single and multi-view 3d object reconstruction. In: European conference on computer vision. pp. 628–644. Springer (2016)
7. Dai, A., Chang, A.X., Savva, M., Halber, M., Funkhouser, T., Nießner, M.: Scannet: Richly-annotated 3d reconstructions of indoor scenes. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 5828–5839 (2017)
8. Dai, A., Ruizhongtai Qi, C., Nießner, M.: Shape completion using 3d-encoder-predictor cnns and shape synthesis. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 5868–5877 (2017)
9. Eigen, D., Puhrsch, C., Fergus, R.: Depth map prediction from a single image using a multi-scale deep network. In: Advances in neural information processing systems. pp. 2366–2374 (2014)
10. Fan, H., Su, H., Guibas, L.J.: 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. pp. 605–613 (2017)
11. Geiger, A., Lenz, P., Stiller, C., Urtasun, R.: Vision meets robotics: The kitti dataset. International Journal of Robotics Research (IJRR) (2013)
12. Giancola, S., Zarzar, J., Ghanem, B.: Leveraging shape completion for 3d siamese tracking. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 1359–1368 (2019)
13. Han, Z., Wang, X., Liu, Y.S., Zwicker, M.: Multi-angle point cloud-vae: unsupervised feature learning for 3d point clouds from multiple angles by joint self-reconstruction and half-to-half prediction. arXiv (2019)
14. Insafutdinov, E., Dosovitskiy, A.: Unsupervised learning of shape and pose with differentiable point clouds. In: Advances in Neural Information Processing Systems. pp. 2802–2812 (2018)
15. Li, Y., Bu, R., Sun, M., Wu, W., Di, X., Chen, B.: Pointcnn: Convolution on x-transformed points. In: Advances in Neural Information Processing Systems. pp. 820–830 (2018)
16. Manivasagam, S., Wang, S., Wong, K., Zeng, W., Sazanovich, M., Tan, S., Yang, B., Ma, W.C., Urtasun, R.: Lidarsim: Realistic lidar simulation by leveraging the real world. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 11167–11176 (2020)

17. Newcombe, R.A., Izadi, S., Hilliges, O., Molyneaux, D., Kim, D., Davison, A.J., Kohi, P., Shotton, J., Hodges, S., Fitzgibbon, A.: Kinectfusion: Real-time dense surface mapping and tracking. In: 2011 10th IEEE International Symposium on Mixed and Augmented Reality. pp. 127–136. IEEE (2011)
18. Qi, C.R., Liu, W., Wu, C., Su, H., Guibas, L.J.: Frustum pointnets for 3d object detection from rgb-d data. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 918–927 (2018)
19. Qi, C.R., Su, H., Mo, K., Guibas, L.J.: Pointnet: Deep learning on point sets for 3d classification and segmentation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 652–660 (2017)
20. Qi, C.R., Yi, L., Su, H., Guibas, L.J.: Pointnet++: Deep hierarchical feature learning on point sets in a metric space. In: Advances in neural information processing systems. pp. 5099–5108 (2017)
21. Rusu, R.B., Blodow, N., Beetz, M.: Fast point feature histograms (fpfh) for 3d registration. In: 2009 IEEE International Conference on Robotics and Automation. pp. 3212–3217. IEEE (2009)
22. Stutz, D., Geiger, A.: Learning 3d shape completion from laser scan data with weak supervision. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 1955–1964 (2018)
23. Tang, C., Tan, P.: Ba-net: Dense bundle adjustment network. arXiv preprint arXiv:1806.04807 (2018)
24. Tchapmi, L.P., Kosaraju, V., Rezatofighi, H., Reid, I., Savarese, S.: Topnet: Structural point cloud decoder. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 383–392 (2019)
25. Triggs, B., McLauchlan, P.F., Hartley, R.I., Fitzgibbon, A.W.: Bundle adjustment—a modern synthesis. In: International workshop on vision algorithms. pp. 298–372. Springer (1999)
26. Tulsiani, S., Efros, A.A., Malik, J.: Multi-view consistency as supervisory signal for learning shape and pose prediction. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 2897–2905 (2018)
27. Tulsiani, S., Zhou, T., Efros, A.A., Malik, J.: Multi-view supervision for single-view reconstruction via differentiable ray consistency. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 2626–2634 (2017)
28. Ummenhofer, B., Zhou, H., Uhrig, J., Mayer, N., Ilg, E., Dosovitskiy, A., Brox, T.: Demon: Depth and motion network for learning monocular stereo. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 5038–5047 (2017)
29. Varley, J., DeChant, C., Richardson, A., Ruales, J., Allen, P.: Shape completion enabled robotic grasping. In: 2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). pp. 2442–2447. IEEE (2017)
30. Wang, Y., Sun, Y., Liu, Z., Sarma, S.E., Bronstein, M.M., Solomon, J.M.: Dynamic graph cnn for learning on point clouds. ACM Transactions on Graphics (TOG) (2019)
31. Xiang, Y., Mottaghi, R., Savarese, S.: Beyond pascal: A benchmark for 3d object detection in the wild. In: IEEE Winter Conference on Applications of Computer Vision. pp. 75–82. IEEE (2014)
32. Yan, X., Yang, J., Yumer, E., Guo, Y., Lee, H.: Perspective transformer nets: Learning single-view 3d object reconstruction without 3d supervision. In: Advances in Neural Information Processing Systems. pp. 1696–1704 (2016)

33. Yuan, W., Khot, T., Held, D., Mertz, C., Hebert, M.: Pcn: Point completion network. In: 2018 International Conference on 3D Vision (3DV). pp. 728–737. IEEE (2018)
34. Zhou, Q.Y., Park, J., Koltun, V.: Open3D: A modern library for 3D data processing. arXiv:1801.09847 (2018)
35. Zhou, T., Brown, M., Snavely, N., Lowe, D.G.: Unsupervised learning of depth and ego-motion from video. In: CVPR (2017)
36. Zhou, Y., Barnes, C., Lu, J., Yang, J., Li, H.: On the continuity of rotation representations in neural networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 5745–5753 (2019)
37. Zhu, R., Kiani Galoogahi, H., Wang, C., Lucey, S.: Rethinking reprojection: Closing the loop for pose-aware shape reconstruction from a single image. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 57–65 (2017)