Journal of Intelligent & Robotic Systems (2022) 104: 45
https://doi.org/10.1007/s10846-022-01586-4

REGULAR PAPER

Improving Object Grasp Performance via Transformer-Based Sparse Shape Completion

Wenkai Chen[1] · **Hongzhuo Liang**[1] · **Zhaopeng Chen**[2] · **Fuchun Sun**[3] · **Jianwei Zhang**[1]

Received: 28 July 2021 / Accepted: 28 January 2022 / Published online: 26 February 2022

Abstract

Currently, robotic grasping methods based on sparse partial point clouds have attained excellent grasping performance on various objects. However, they often generate wrong grasping candidates due to the lack of geometric information on the object. In this work, we propose a novel and robust sparse shape completion model (TransSC). This model has a transformer-based encoder to explore more point-wise features and a manifold-based decoder to exploit more object details using a segmented partial point cloud as input. Quantitative experiments verify the effectiveness of the proposed shape completion network and demonstrate that our network outperforms existing methods. Besides, TransSC is integrated into a grasp evaluation network to generate a set of grasp candidates. The simulation experiment shows that TransSC improves the grasping generation result compared to the existing shape completion baselines. Furthermore, our robotic experiment shows that with TransSC, the robot is more successful in grasping objects of unknown numbers randomly placed on a support surface.

Keywords Robotic grasping · Point cloud · Sparse shape completion · Object segmentation

1 Introduction

Robotic grasping evaluation is a challenging task due to incomplete geometric information from single-view visual sensor data [28]. Many probabilistic grasp planning models have been proposed to address this problem, such as Motel Carlo, Gaussian Process and uncertainty analysis [10, 17, 26]. However, these analytic methods are always computationally expensive. With the development of deep learning techniques, data-driven grasp detection methods have shown great potential [4, 15, 22, 31] to solve this problem. They generate lots of grasp candidates and estimate the corresponding grasp quality, resulting in a better grasp performance and generalization. However, as most of these methods still rely on original sensor input like 2D (image) and 2.5D (depth map), there exists a physical grasping defect when the gripper interacts with real object surfaces or edges because of the incomplete pixel-wise and point-wise representations. Otherwise, traditional data-driven grasping algorithms [15, 21, 22]are mostly based on the partial point clouds. Due to the object's missing geometric and semantic information, these algorithms are easily to generate wrong grasp candidates and causing a research gap.

To improve grasp performance, the sparse point cloud is necessary to be restored or repaired to generate a better grasping interaction. Additional sensor input such as a tactile sensor is introduced to supplement original vision sensing [30]. However, object uncertainty still exists and extra sensor interference with the object will directly affect the final grasping result. Another strategy is to use shape completion to infer the original object shape while traditional grasping-based shape completion methods use a high-resolution voxelized grid as object representation [17, 18, 27], causing a high memory cost and information loss due to the sparsity of the sensory input. To avoid extra sensor cost and obtain complete object information,

✉ Hongzhuo Liang
liang@informatik.uni-hamburg.de

[1] Technical Aspects of Multimodal Systems (TAMS), Department of Informatics, Universität Hamburg, Hamburg, Germany

[2] Agile Robots AG, München, Germany

[3] Beijing National Research Center for Information Science and Technology (BNRist), State Key Lab on Intelligent Technology and Systems, Department of Computer Science and Technology, Tsinghua University, Beijing, China

a novel transformer-based shape completion module is proposed in this work based on an original sparse point cloud. Compared with the traditional convolutional network layer, the transformer has achieved state-of-the-art results in visual recognition and segmentation recently [11, 25], which enables our shape completion module to achieve better performance.

As illustrated in Fig. 1, we present a novel grasping pipeline that uses a sparse point cloud to execute the grasp directly, without converting it into discrete voxel grids during the shape completion process and then transforming it into a mesh in the grasp planning process. The pipeline consists of two sub-modules: The transformer-based shape completion module and the grasp evaluation module. In the first module, a non-synthetic segmented partial point cloud dataset based on YCB objects was constructed. Not cropping the object randomly or viewing the object in a physical simulator, our dataset contains many real cameras and environmental noise, which guarantees an improved grasping interaction in a real robot environment. Based on this dataset, we propose a novel point cloud completion network (TransSC), where the segmented partial point cloud of an object is input, and the complete point cloud is output. In the second module, our previous work [15] is involved. We use PointNet [24] to obtain feature representation of the repaired point cloud and build a grasp detection network to generate and evaluate a set of grasp candidates. The grasp with the highest score will be executed in a real robot experiment. The proposed pipeline is validated in a simulation experiment and robotic experiments, which demonstrate that our shape completion pipeline can significantly improve grasping performance.

Our contributions in this paper can be listed as:

- A large-scale non-synthetic partial point cloud dataset is constructed based on the YCB-Video dataset. As the dataset is based on 3D point cloud data captured by a real RGB-D camera, the noise that comes from it will facilitate the generalization of our work, especially in real robot environments.
- A novel point cloud completion network TransSC is proposed. The transformer-based encoder and

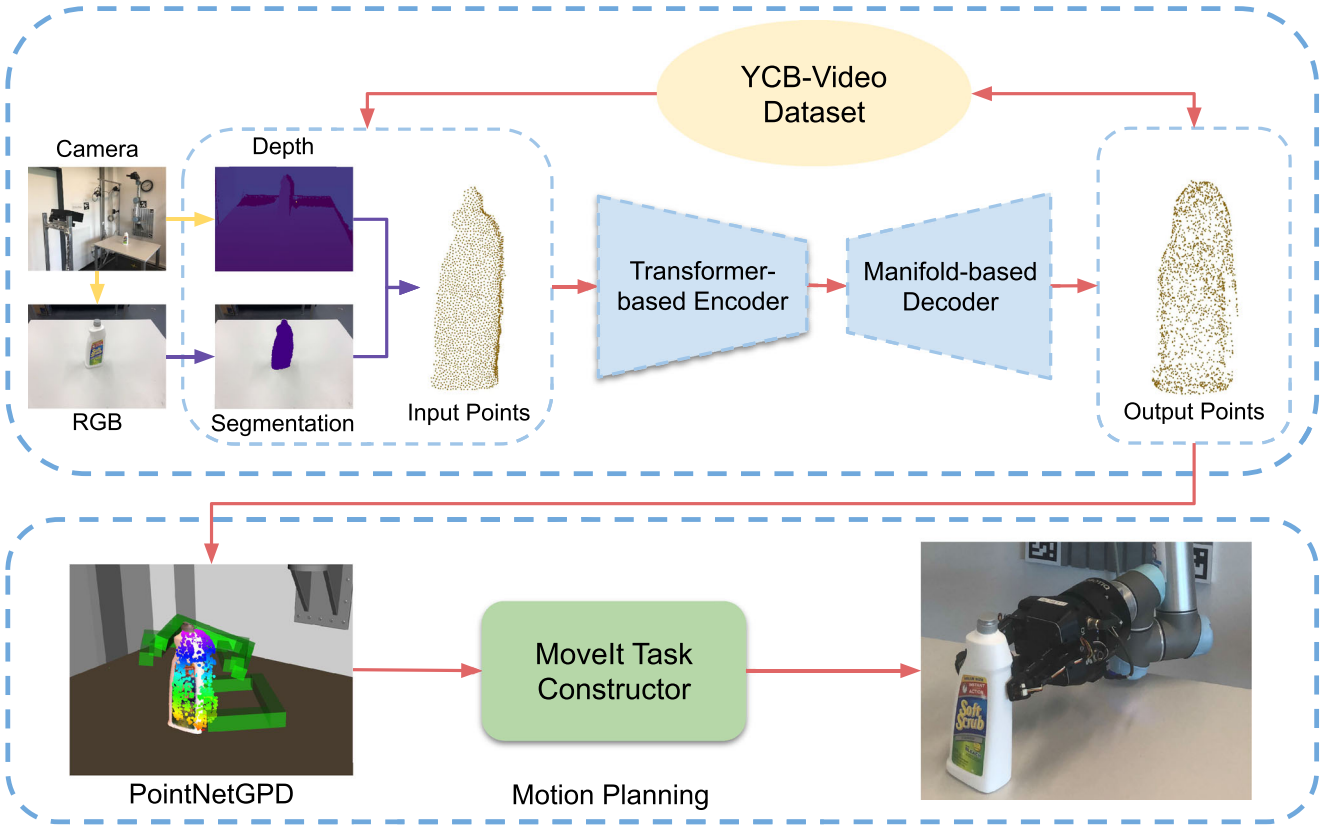

Fig. 1 Overview of our shape completion based grasp pipeline. The top row shows the shape completion module. In this module, a segmented partial point cloud ζ_p with n points is first input into a transformer-based encoder to extract point-wise and self-attention features, which outputs a latent vector with m dimensions. Then, the latent vector is concatenated with another latent feature from a flat/spatial point seed generator to predict multiple spatial surfaces in the manifold-based decoder. Finally, these surfaces are assembled into a complete point cloud ζ_c. The bottom row is the grasp evaluation module, the complete point cloud ζ_c is the input of our grasp detection pipeline PointNetGPD to compute the grasp quality $\mathcal{Q}_i$. The grasp with the highest score $\mathcal{G}_{best}$ will be sent to calculate a collision-free trajectory and will be executed in a real robot experiment

manifold-based decoder are introduced into the shape completion task to improve its performance.
– Combining our previous work PointNetGPD for grasp evaluation and the MoveIt Task Constructor for motion planning, we demonstrate that a robust grasp planning pipeline using the shape completion result as input can achieve a better grasp planning result compared to the single view without shape completion work.

The paper is organized as follows. We first contrast related approaches for visual grasping, dense point cloud completion and traditional robotic grasping strategies based shape completion in Section 2. Then, we propose our problem formulation in Section 3. Furthermore, we explain the different components of our grasping evaluation approach: dataset construction, transformer-based shape completion network architecture and grasping detection module in Section 4. After that, we evaluate our method through quantitative evaluation, simulation grasping experiments and real robotic experiments on single-object and object-occlusion scenes in Section 5. Finally, our conclusion and future work is drawn in Section 6.

2 Related Work

Deep Visual Robotic Grasping With the development of deep learning, many methods for deep visual grasping have been proposed. Similar to 2D object recognition, monocular camera images were firstly used to predict the probability that the input grasps were successful [14]. In [5] and [26], a single RGB-D image of the target object was used to generate a 6D-pose grasp and effective end-effector trajectories. However, this work is not suitable to deal with sparse 3D object information and spatial grasps. Compared with the 2D feature representations from images, 3D voxel or point cloud data could provide robotic grasping with more semantic and spatial information. Given a synthetic grasp dataset, [4] transformed scanning 3D object information into Truncated Signed Distance Function (TSDF) representations and passed them into a Volumetric Grasping Network (VGN) to directly output grasp quality, gripper orientation and gripper width at each voxel. Wu et al. [31] designed a special grasp proposal module that defines anchors of grasp centers and related 3D grid corners to predict a set of 6D grasps from a partial point cloud. Based on the scaled point cloud, [22] used hand-crafted outline features and a CNN-based method to build a grasp quality evaluation model. In our previous work [15], we used PointNet [24] to extract raw point cloud features and built a grasp evaluation network, which performs great in robotic grasping experiments. However, due to the lack of complete geometric information on the object, we found that some grasp candidates are still infeasible and cause a collision with the object.

Dense Point Cloud Completion The task of point cloud completion has been attracting more and more attention in the field of computer vision. Yuan et al. [36] firstly used Multi-layer Perceptrons (MLP) to extract the local geometric features of point clouds to accomplish the reconstruction. Groueix et al. [9] introduced a morphing learning strategy to generate different shapes of 3D surfaces, which shows great potential for point cloud and voxel reconstruction. Liu et al. [16] combined the above work and proposed a morphing and sampling network, which shows a higher fidelity and quality for the dense point cloud. Furthermore, [34] proposed a Gridding Residual Network to restore more structural details, especially for the dense point cloud. However, these methods cannot be applied to robotic research directly because all trained objects in their datasets are at the same pose and status. This would create dense point cloud models too complicated to pursue the details of the point cloud. It is better to restore a sparse completion of object surfaces for robotic tasks.

Shape Completion for Robotic Grasping For robotic grasping, the critical challenge is recognizing objects in 3D space and avoiding potential perception uncertainty. When the RGB-D camera captures an object from a particular viewpoint, the 3D information on the object is incomplete, which means a lot of semantic and spatial information is missing. The missing of complete 3D object information will lead to the grasp generation process generating wrong grasping poses.

Recently, researchers have proposed to use shape completion to enable robotic grasping. In [27], the observed object from 2.5D range sensors was firstly converted to occupancy voxel grid data. Then the voxelized data were input into a CNN and formed a high-resolution voxel output. Furthermore, the completion result was transformed into mesh and then loaded into Graspit! [20] to generate a grasp. Lundell et al. [17] used dropout layers to modify the network, which enabled the prediction of shape samples at runtime. Meanwhile, Monte Carlo Sampling and probabilistic grasp planning were used to generate grasp candidates. As traditional analytic grasping methods are computationally expensive, [18] combined the shape completion of a voxel grid and a data-driven grasping planning strategy (GQCNN) [19] to propose a structure called FC-GQCNN, where synthetic object shapes were obtained from a top-down physics simulator and grasps were generated from depth images. Traditional grasp-based shape completion solutions mainly concentrate on completing a single object from different camera views, while they hardly consider the lack of geometric information caused by occlusion from other objects.

In conclusion, traditional grasp shape completion methods mainly voxelized the 2.5D data into occupancy grids or distance fields to train a CNN. However, these high-resolution voxel grids will entail a high memory cost. Moreover, detailed semantic information is often lost in the form of occlusions of other objects, which causes meaningful geometric features of objects not to be learned from the neural network. We propose a transformer-based shape completion module to obtain the complete geometric features and retain original object information. Without converting the observed partial point cloud into the voxel grid and mesh, our completion method segments the sparse point cloud of the target object and outputs a repaired point cloud at arbitrary resolution, which outperforms existing methods. Furthermore, PointNet [24] is introduced for the representation learning of the repaired point cloud and a grasp evaluation network is constructed to generate grasp candidates. Finally, our grasp evaluation pipeline achieves a better grasping performance than the baseline method without point cloud completion.

3 Problem Formulation

We consider a setup consisting of a robotic arm with parallel-jaw grippers, an RGB-D camera, and objects of unknown number that are set on a flat support surface while we define a target object via user input. Meanwhile, we assume that the RGB-D camera could capture the depth map of objects, where a semantic segmentation network is used to extract the mask of the target object and convert it into a 2.5D partial point cloud $\mathcal{P} \in \mathcal{R}^{N\times 3}$. For simplicity, all spatial quantities are in camera coordinates.

Given a gripper configuration $\mathcal{C}$ and camera observation $\mathcal{O}$, our goal is firstly to extract the target object point cloud $\mathcal{P}$ using semantic segmentation. Then a point cloud completion network is used to repair the segmented 2.5D partial point cloud $\mathcal{P} \in \mathcal{R}^{N\times 3}$, turning it into a complete 3D point cloud $\mathcal{P}_c \in \mathcal{R}^{N\times 3}$. After that, a grasp evaluation network based on $\mathcal{P}_c$ is used to predict a set of grasp candidates $\mathcal{G}_i$ and compute the relative grasp quality $\mathcal{Q}_i$. The grasp with the highest score $\mathcal{G}_{best}$ and highest kinematic possible, i.e., a collision-free grasp, will be executed in the real robot experiment.

4 Robotic Grasping Evaluation via Shape Completion and Grasp Detection

4.1 Dataset Construction

Traditional shape completion models use synthetic CAD models from the ShapeNet [35] or ModelNet [32] datasets to generate partial and corresponding complete point cloud data, while these synthetic data contain no real-world noise. As a result, synthetic data often do not work well in the real world. To tackle this problem, we summarize a shape completion dataset from the YCB-Video Dataset [33]. Non-synthetic RGB-D video images ($\sim$ 133,827 frames) in the YCB-Video Dataset are firstly chosen, while most of them vary insignificantly. Thus, a preprocessed image dataset is obtained by reducing every five frames. Meanwhile, to cover distinguishable shapes with different levels of detail, 18 objects are also chosen from the YCB-Video dataset. In this work, the ground-truth point cloud of 18 objects is created by the farthest point sampling (FPS) of 2048 points on each object model. Not randomly sampling or cropping complete point clouds on the unit sphere to get partial point clouds, RGB-D images and related object label images in the preprocessed dataset are loaded to compute the matching partial point clouds using related camera intrinsic parameters. To approximate the distribution of point cloud data of real objects and retain the semantic information, a large number of cameras and environmental noise data are kept on, though a small radius is used to remove partial outliers. For the convenience of network training, the partial point clouds are also unified into the size of 2048 points by FPS or replicating points. To enable an accurate comparison with existing baselines, the canonical center of the partial point cloud of each object is transformed into the canonical center of the ground-truth point cloud using pose information. Finally, more than 70,000 partial point clouds are collected in our dataset. Compared to other synthetic point cloud datasets, our dataset also does well at preserving the real point cloud distribution of occluded objects.

4.2 Semantic Segmentation

As shown in Fig. 1, the scene of our grasping task is that objects of unknown number are set on a flat support surface. To obtain the target object point cloud, we first build a semantic segmentation network branch, where different YCB objects are assigned a particular semantic label value. It can be seen that the performance of the segmentation network is good enough that it can also be deployed in a grasping task of multi-object occlusion.

Our segmentation network [2] takes an RGB image as input and outputs a binary mask of the expected object. The network has an encoder-decoder architecture based on CNN, where the encoder consists of convolutional layers with ReLU activation followed by max-pooling layers. At the same time, the decoder utilizes unpooling operations whereby the pooling indices from the corresponding encoder layers are recalled. Convolutional layers again follow this upsampling strategy. Moreover, several data augmentation strategies like adjusting brightness, contrast

and saturation are used to make the network generalize well. After getting the expected object mask, the sparse 2.5D point cloud $\mathcal{P} \in \mathcal{R}^{N\times 3}$ of the target object could be extracted through the corresponding depth image. Meanwhile, we also remove the redundant background (support surface) point cloud by setting a threshold value of the z-axis (support surface height).

4.3 Transformer-Based Encoder Module

As shown in Fig. 2, we compare our proposed encoder module with several common competitive methods. Multi-layer Perception (MLP) is a simple baseline architecture to extract point features. This method maps each point into different dimensions and extracts the maximum value from the final K dimensions to formulate a latent vector. A simple generalization for MLP is to combine semantic features from a low-level dimension with those of a high-level dimension. The MSF (Multi-scale Fusion) [13] module inflates the dimension of the latent vector from 1024 to 1408 to obtain semantic features from different dimensions. To improve the performance of the feature extractor, L-GAN [1] proposed to use a Maxpooling layer appropriately. Concatenated Multiple Layer Perception (CMLP) [12] maxpools the output of the last k layers to guarantee that multi-scale feature vectors are concatenated directly. An overview of our proposed Transformer-based multi-layer perception (TMLP) module is shown in Fig. 2(d). Without an extra skip connection structure and a maxpooling operation from different layers, the Multi-head Self-attention (MHSA) [29] module is introduced to replace the traditional convolutional layer [$128 \times 256 \times 1$].

MHSA aims to transform (encode) the input point feature into a new feature space, which contains point-wise and self-attention features. Figure 2(e) shows a simple MHSA architecture used in TMLP, which includes two sub-layers. In our first layer, the multi-head number is set to 8 and the input feature dimension for each point is 128. Unlike natural language processing (NLP) problems, the 128-dimensional feature vector $\mathcal{A}_{in} \in \mathcal{R}^{2048\times 128}$ will enter into the multi-head attention module directly without positional encoding. This is because each point in the point cloud has its unique $x - y - z$ coordinates. The output feature $\mathcal{Z}$ is formed by concatenating the attention of each attention head. A residual structure is also used to add and normalize the output feature $\mathcal{Z}$ with $\mathcal{A}_{in}$. This process can be formulated as follows:

$$\mathcal{A}_i = SA_i(\mathcal{A}_{in}) \quad i = 1, 2, ..., 8 \tag{1}$$

$$\mathcal{Z} = concat(\mathcal{A}_1, \mathcal{A}_2, ..., \mathcal{A}_8) * W_0 \tag{2}$$

$$\mathcal{A}_{out} = Norm(\mathcal{A}_{in} + \mathcal{Z}) \tag{3}$$

where SA_i represents the i-th self-attention layer, each has the same output dimension size with input feature vector $\mathcal{A}_{in}$, and W_0 is the weight of the linear layer. $\mathcal{A}_{out}$ represents the output point-wise features of the first sub-layer.

The second sub-layer is called Feed-forward module, which is a fully connected network. Point-wise features $\mathcal{A}_{out}$ are processed through two linear transformations and one ReLU activation. Furthermore, a residual network is also used to fuse and normalize the output features. Finally, we can get the MHSA module output $\mathcal{FF}_{out} \in \mathcal{R}^{2048\times 128}$ as:

$$\mathcal{FF} = ReLU(\mathcal{A}_{out} * W_1 + b_1) * W_2 + b_2 \tag{4}$$

$$\mathcal{FF}_{out} = Norm(\mathcal{A}_{out} + \mathcal{FF}) \tag{5}$$

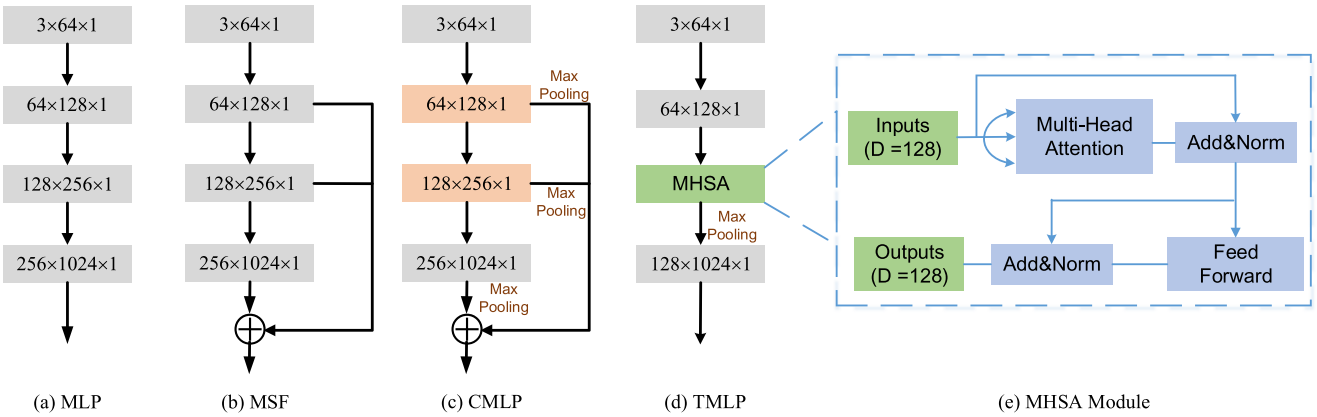

Fig. 2 Illustration of various encoder structures for point cloud completion. (**a**) is a simple multiple-layer perception (MLP) structure. (**b**) is a multi-scale fusion (MSF) module, which can fuse features from different layers directly. (**c**) is concatenated multiple layer perception (CMLP), which can also concatenate multi-dimensional latent features while the max pooling operation is used to extract latent features further. (**d**) shows our Transformer-based multiple layer perception (TMLP) module, which integrates the Multi-head Self-attention (MHSA) module into the MLP structure. (**e**) depicts the architecture of the MHSA module

where W_1, W_2 and b_1, b_2 represent the weight and bias value of the corresponding linear transformation, respectively.

4.4 Manifold-Based Decoder Module

Inspired by the AtlasNet [9], a manifold-based decoder module is designed to predict a complete point cloud from partial point cloud features. As shown in Fig. 3, a complete point cloud could be assumed to consist of multiple sub-surfaces. Therefore, we only concentrate on obtaining each sub-surface, then we gather them and make an appropriate montage to form the final complete point cloud. To obtain each sub-surface, a point seed generator is used to concatenate with global feature vector $\mathcal{P}_g \in \mathcal{R}^{2048\times 1024}$ output from the encoder, where point initialization values are computed from a flat (f) or spatial (g) sampler. As the coordinate values of the ground-truth point cloud are limited to between $[-1, 1]$, point initialization values are also limited in this range. After that, the concatenated feature vector $\mathcal{P}_{concat} \in \mathcal{R}^{2048\times M}$ $(M = 1026\ or\ 1027)$ is input into K convolutional layers, where all sampled 2D or 3D points will be mapped to 3D points on each sub-surface. In our decoder, the sub-surface number is set to 16. Unlike other voxel-based shape completion methods, our decoder module achieves an arbitrary resolution for the completion results.

Evaluation Metrics To evaluate our shape completion results, we used two permutation-invariant metrics called Chamfer Distance (CD) and Earth Mover's Distance (EMD) as our evaluation goal [7]. Given two arbitrary point clouds S_1 and S_2, CD measures the average distance from each point in one point cloud to its nearest point coordinates in the other point cloud.

$$d_{CD}(S_1, S_2) = \frac{1}{S_1}\sum_{x\in S_1}\min_{y\in S_2}\|x-y\|_2^2 + \frac{1}{S_2}\sum_{y\in S_2}\min_{x\in S_1}\|y-x\|_2^2 \tag{6}$$

While Earth Mover's Distance considers two equal point sets S_1 and S_2 and is defined as:

$$d_{EMD}(S_1, S_2) = \min_{\emptyset: S_1\to S_2}\frac{1}{S_1}\sum_{x\in S_1}\|x-\emptyset(x)\|_2 \tag{7}$$

CD has been widely used in most shape completion tasks because it is efficient to compute. However, EMD is chosen as our completion loss because CD is blind to some visual inferiority and ignores details easily [1]. With $\emptyset : S_1 \to S_2$ being bijective, EMD could solve the assignment and transformation problem in which one point cloud is mapped into another.

4.5 PointNetGPD Based Grasping Detection Module

Given the complete point cloud from the previous steps, we put the point cloud into a geometric-based grasp pose generation algorithm (GPG) [23], which outputs a set of grasp proposals $\mathcal{G}_i$. We then transform $\mathcal{G}_i$ into a gripper coordinate system and use points inside the gripper as the input of PointNetGPD, a data-driven grasp evaluation framework. The output grasp will then be sent to the MoveIt Task Constructor [8] to plan a feasible trajectory for a pick and place task.

PointNetGPD [15] is trained on a grasp dataset generated using a reconstructed YCB object mesh and evaluates the input grasp quality. The grasp candidates in the grasp dataset all proceeded collision-free to the target object. As a result, the grasp evaluation network assumes that all the input grasp candidates are not colliding with the object. If the object has occlusion due to the camera viewpoint, the current geometric-based grasp proposal algorithm will generate grasp candidates that collide with the object. Thus, using a complete point cloud could ensure that the grasp candidate generation algorithm generates grasp sets that do not collide with the graspable objects. Figure 4 shows the comparison

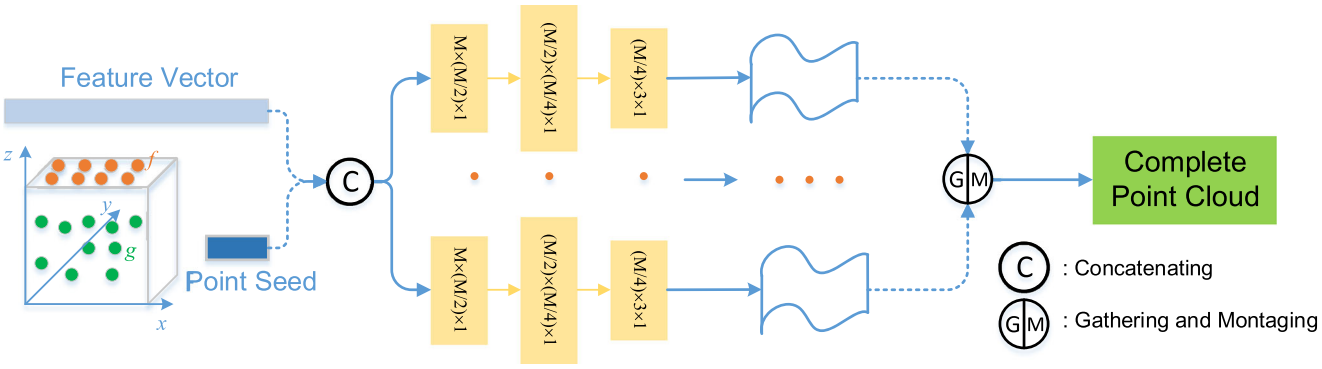

Fig. 3 Illustration of the decoder structure for point cloud completion. The feature vector with m dimensions from the encoder is firstly concatenated with a latent feature from a special point seed generator f or g. Then three convolutional layers as the backbone are used to extract features and form different manifold-based surfaces, respectively. Finally, these surfaces are gathered and montaged into a complete point cloud

Fig. 4 Comparison of grasp candidates generated using GPG. (**a**) RGB image to show the example environment, (**b**) grasp generated with partial point cloud, (**c**) grasp generated with complete point cloud

of the grasp generation result using GPG [23] with and without point cloud completion, where Fig. 4(b) shows a candidate generated using a partial point cloud and Fig. 4(c) shows a grasp candidate generated using a complete point cloud. We can see that the grasp in Fig. 4(b) collides with the real object while Fig. 4(c) avoids generating that kind of grasp.

5 Experiments

5.1 Quantitative Evaluation of Proposed Shape Completion Network

Training and Implementation Details To evaluate model performance and reduce training time, eight categories of different objects in our dataset are chosen to train the shape completion model. The training set and validation set are split into 0.8:0.2. We implement our network on PyTorch. All the building modules are trained using the Adam optimizer with an initial learning rate of 0.0001 and a batch size of 16. All the parameters of the network are initialized using a Gaussian sampler. Batch Normalization (BN) and ReLU activation units are all employed at the encoder and decoder module except the final tanh layer producing point coordinates, and Dropout operation is used in the MHSA module to suppress model overfitting.

5.1.1 Comparison with Existing Methods

In this subsection, we compare our method against several representative baselines that are also used for point cloud completion, including AtlasNet [9], MSN [16] and GRNet [34]. The Oracle method means that we randomly resample 2048 points from the original surface of different YCB objects. Corresponding EMD and CD distances between the resampled point cloud and the ground-truth point cloud provide an upper bound for the performance. Relative comparison results are shown in Tables 1 and 2. Our method is developed into two models based on the different point seed generators (f/g) in the decoder module. It can be seen that our method outperforms other methods in most objects on both EMD and CD distances. Though for some objects like banana and cracker box, the evaluation metrics of Earth Mover's Distance and Chamfer Distance from our both models are bigger than other baselines. However, for other objects in our datatset, our flat/spatial models both achieve a better performance than other baselines.

Table 1 Comparison of earth mover's distance with different sparse point cloud completion models for 2048 points and multiplied by 10^3

Model	Cracker box	Banana	Pitcher base	Bleach cleanser	Bowl	Mug	Power drill	Scissors	Average
Oracle	3.4	1.7	4.6	2.9	1.9	2.0	3.8	1.5	2.7
AtlasNet [9]	9.7	4.9	10.5	10.0	8.8	5.3	15.0	5.2	8.7
MSN (fusion) [16]	10.7	4.6	12.4	14.0	11.5	12.9	23.4	5.3	11.8
MSN (vanilla) [16]	11.0	**3.8**	9.3	8.3	10.2	3.9	5.9	**3.4**	7.0
GRNet (sparse) [34]	**8.4**	4.3	8.8	6.0	6.0	4.3	5.8	4.5	5.3
Our (flat)	8.5	3.9	9.4	6.7	6.0	**3.7**	**5.2**	4.1	**4.9**
Our (spatial)	10.1	4.4	**8.4**	**5.8**	**5.6**	**3.7**	7.0	3.9	6.1

Bold values indicate the best performance

Table 2 Comparison of chamfer distance in different sparse point cloud completion models for 2048 points and multiplied by 10^3

Model	Cracker box	Banana	Pitcher base	Bleach cleanser	Bowl	Mug	Power drill	Scissors	Average
Oracle	0.24	0.52	0.28	0.12	0.10	0.09	0.13	0.38	0.23
AtlasNet [9]	4.51	0.87	4.97	5.61	4.21	1.37	6.18	0.92	3.58
MSN (fusion) [16]	5.59	1.25	5.71	2.77	10.81	1.77	8.34	1.58	4.73
MSN (vanilla) [16]	6.01	**0.71**	4.01	4.68	7.51	0.76	1.28	**0.38**	3.17
GRNet (sparse) [34]	**2.28**	0.97	3.78	1.67	2.85	0.76	1.48	0.88	1.90
Our (flat)	3.28	0.92	4.09	1.50	**2.55**	**0.66**	**1.25**	0.82	**1.88**
Our (spatial)	5.81	0.87	**3.19**	**1.20**	2.79	0.69	2.54	0.66	2.22

Bold values indicate the best performance

More importantly, the final average evaluation metrics of Earth Mover's Distance and Chamfer Distance of Our(flat) model are both the best evaluation results. For the same completion loss function, our (flat) model achieves an average of about 9% improvement in terms of the EMD distance to the latest GRNet model. Since our dataset contains much noise from the camera and the environment, we found that fusing the output completion result with the original point cloud makes the performance significantly worse, which can be seen from the comparison of MSN (fusion) and MSN (vanilla). It also implies that our model is robust enough, which is conducive to rapid deployment in real robot experiments. Furthermore, compared with ideal results from the Oracle method, it demonstrates that point cloud completion remains an arduous task to solve.

To understand the computational complexity of the proposed transformer-based model, we analyse the floating-point operations(FLOPs) and the number of network parameters and summarize in Table 3. It can be seen that the self-attention module introduced in our transformer-based encoder is lighter than traditional convolution layer, reducing the computational complexity. Moreover, after removing a large number of redundant convolution layers existing in traditional dense shape completion, our FLOPs value is also decreased significantly.

5.1.2 Ablation Studies

This section provides a series of ablation studies on our YCB-based dataset to evaluate our proposed shape completion model comprehensively. Accordingly, the effectiveness of each particular module in our model is analyzed as follows: We first evaluate our transformer-based encoder module with other representative encoder modules under the same setting of convolutional/transformer layer number and object inputs. As shown in Table 4, our encoder has a better result overall, though CMLP gets a great result on Mug's completion. When the point seed in the decoder is flat, we further analyze the influence of different point seed distributions and surface numbers in Tables 5 and 6. We can see that both Uniform and Gaussian sample methods can achieve a better result at (0, 1). We choose $Uniform(0, 1)$ in our model to achieve the best results. Like the weight parameters in the neural network, the initialization value of points cannot be close to zero, which predicts the worst result. As illustrated in Table 6, when the sub-surface number increases, the overall model performance improves. However, the improvement of completion results is limited when the number is above 16.

Table 3 Number of FLOPs and network parameters

Method	AtlasNet	MSN	MSN(fusion)	GRNet	Ours
# Params (M)	29.46	30.32	33.65	76.71	30.02
# FLOPs (GMac)	14.36	21.46	——	25.90	9.87

5.1.3 Visualization Analysis

Figure 5 shows the visualized shape completion results using our TransSC. In the visual analysis, each object's input partial point cloud is first preprocessed to remove noisy data from the camera and the environment. It can be seen that the geometric loss of the input point cloud in our dataset comes from the change of the camera viewpoint and the occlusion by other objects, which causes a big challenge for our model. The output results of the canonical pose show that our model works well on all simple and complex objects. Moreover, our model can generate realistic structures and details like the mug handle, bowl edge and bottle mouth. In robotic grasping, as the target object pose is randomly put on the support surface, another shape completion model based on the arbitrary ground-truth pose is retrained. This is done by transforming the ground truth pose to the original pose of the input partial point cloud. The completion results are also shown in Fig. 5. Arbitrary output is not as good as the canonical output while it still restores the overall shape of each object well. It also demonstrates that achieving object completion of arbitrary poses in a real environment is still a formidable task.

Table 4 Comparison of EMD and CD from different encoder structures

Earth Mover's distance (EMD)	MLP	CMLP	MSF	TMLP	Chamfer distance (CD)	MLP	CMLP	MSF	TMLP
Mug	6.01	3.69	9.45	**3.69**	Mug	2.15	**0.65**	13.80	0.66
Bleach cleanser	10.51	8.10	11.70	**6.70**	Bleach cleanser	6.88	2.63	13.89	**1.50**

Bold values indicate the best performance

Table 5 Comparison of average EMD and CD from different point generators

Similarity metrics	Uniform distribution:			Gaussian distribution:			ZERO
	0:1	− 0.5:0.5	− 1:1	0.5,0.5/3	0,0.5	0,1	
Avg EMD	**5.94**	7.09	6.50	6.34	6.15	**6.14**	9.88
Avg CD	**1.89**	3.25	2.42	2.39	2.38	**2.12**	6.17

Bold values indicate the best performance

Table 6 Influence of different surface numbers in the decoder

Earth Mover's distance (EMD)	n=4	n=8	n=16	n=32	Chamfer distance (CD)	n=4	n=8	n=16	n=32
Mug	4.71	3.94	3.70	**3.61**	Mug	9.01	6.70	**6.61**	6.69
Bleach cleanser	10.10	7.82	6.69	**5.94**	Bleach cleanser	3.69	1.70	**1.51**	1.53

Bold values indicate the best performance

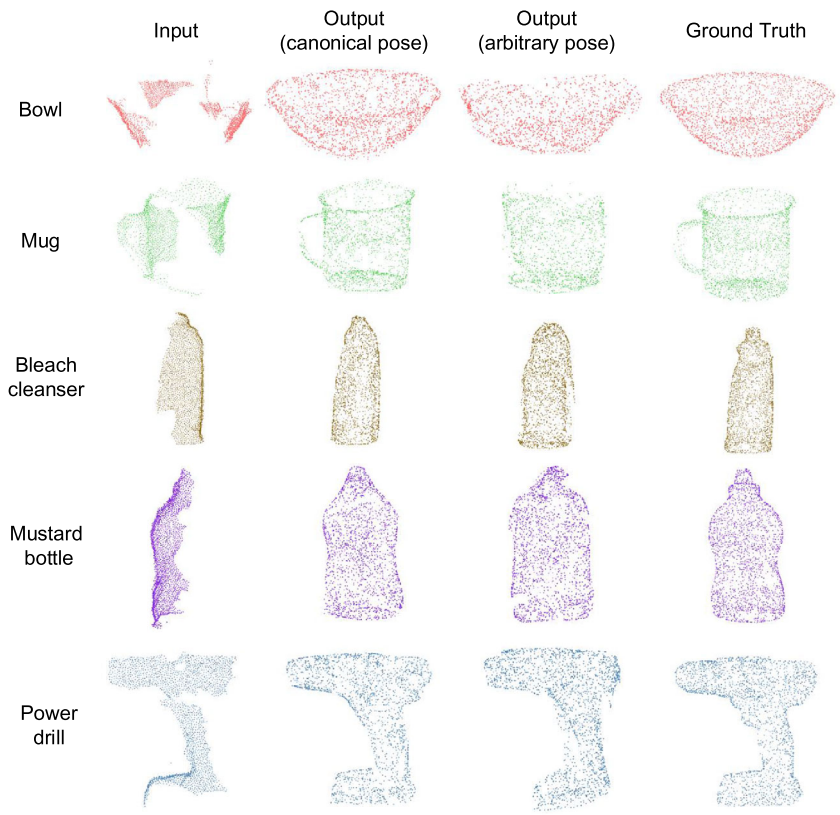

Fig. 5 Shape completion result using TransSC. The canonical pose result is trained under a fixed point cloud coordinate system while the arbitrary pose result is trained under the camera perspective. In the robot experiment, the arbitrary pose training result is used to generate grasps

5.2 Simulation Grasp Experiments with Complete Shape

Experimental Setup of Simulation Experiments We use Graspit! [20] to evaluate the quality of shape completion similar to [27]. First, the Alpha shapes algorithm [6] is used to implement surface reconstruction of the completion object. The output 3D mesh is then imported into GraspIt! Simulator to calculate grasps. To have a fair comparison, we also use a Barrett Hand to generate grasps. After finishing the grasp generation, we remove the completion object and import the ground-truth object into the same place. Meanwhile, the Barrett Hand is moved back 20 cm along the approach direction and then approaches the object until the gripper detects a collision or reaches the calculated grasp pose. Furthermore, we adjust the gripper to the calculated grasp joint angles and perform the auto-grasp function in GraspIt! to ensure the gripper makes contact with the object surface or reaches the joint limit. The different values of joint angles at different positions are then recorded. We use four objects (bleach cleanser, cracker box, pitcher base and power drill) from the YCB objects set and calculate 100 grasps for each object in our experiment.

Assuming the grasp pose is the same, we compare the average difference of the joint angle from our shape completion model to that of Laplacian smoothing in Meshlab (Partial), mirroring completion [3] (Mirror) and voxel-based completion [27]. Note that we use two different models, canonical and arbitrary. The canonical model means all the training is transformed into the same object coordinate system and the arbitrary model means all the training data are transformed into the camera's coordinate system. Although we can see from Fig. 5 that the canonical model has a better shape completion result, it requires a 6D pose of the target object if we want to map the complete point cloud into the real environment. To avoid this complication of adding a 6D pose estimation module and achieve real robot experiments, the arbitrary model is also trained. The simulation result is shown in Table 7. It can be seen that Ours (canonical) gets the best simulation grasping performance, which outperforms other completion types. Ours (arbitrary) also obtains a great simulation result though its average joint angle is slightly smaller than voxel-based methods. Moreover, the average difference between the two models also demonstrates that a perfect shape completion in an arbitrary pose is much harder than in a canonical pose.

5.3 Robotic Experiments on Single Objects

Experimental Setup of Single Objects To evaluate the performance improvement using a complete point cloud for robotic grasping, we choose six YCB objects to test the grasping success rate. The robot for evaluation is a UR5 robot arm equipped with a Robotiq 3-finger gripper. The vision sensor is an Industrial 3D camera from Mechmind[1] to acquire a high-quality partial point cloud. The selected six objects are listed in Table 8. We select these objects because they are typical objects that may fail to generate good grasp candidates without shape completion. Other objects such as a banana or a marker are quite simple and small, so that improvement of shape completion on the grasping result is minor. In our robotic experiments, each YCB object is firstly placed on the center of flat table and then moves randomly as long as it can appear in the field of the vision sensor and within the executable range of our UR5 robot arm.

For the selected six objects, we perform grasp evaluation based on PointNetGPD [15] on two different methods: Without our shape completion (WOSC) and with our shape completion (WSC). We run the robot experiment by randomly putting the object on the table and grasping it ten times, then calculating the success rate. The experiment result is shown in Table 8. We can see that all six objects' grasp success rates from our grasp pipeline outperform or are even with the original method. The low success rate of the power drill for both methods is due to the contact area of the power drill head being too slippery for the robot to grasp. The failures of WOSC with the observed point cloud input are mainly due to the limit of the camera viewpoint, and GPG generates grasp candidates that sink into the object. An example of this situation is shown in Fig. 4, which is strong evidence that our shape completion model can improve the grasp success rate in some particular objects.

5.4 Robotic Experiments on Object Occlusion

Experimental Setup of Object Occlusion When there are different objects on the flat table, the occlusions from other objects will cause a lack in geometric information on the target object. To simulate this scene, we choose bleach cleanser as the target object and other YCB objects are picked as a potential occluder where occluder as foreground is placed directly in front of the target object. All objects are placed in a natural vertical position while the horizontal distance between the two types of objects is set to 8 cm. The experimental objects and segmentation result of the target object can be seen from Fig. 6. The robot arm and camera are the same as in the robotic experiment on the single object. Furthermore, in real experiments, the target object is placed near the center of table to ensure that vision camera could capture it accurately and then we randomly change the 6D pose of target object to grasp ten times.

As shown in Fig. 7, we compare the grasping performance of WOSC and WSC when five different YCB objects

[1] https://en.mech-mind.net/

Table 7 Comparison of average difference between grasp joints from different completion types

Error	Partial	Mirror	Voxel-based	Ours (canonical)	Ours (arbitrary)
Grasp joint (degree)	10.07	4.42	2.17	**1.15**	2.02

Bold values indicate the best performance

Table 8 Robotic grasping performance on a single object

Method	Cracker box	Mug	Meat can	Pitcher base	Bleach cleanser	Power drill	Average
WOSC	70%	70%	80%	80%	90%	40%	71.67%
WSC	80%	100%	100%	80%	90%	50%	83.33%

Fig. 6 The target object and segmentation result with different occlusion settings

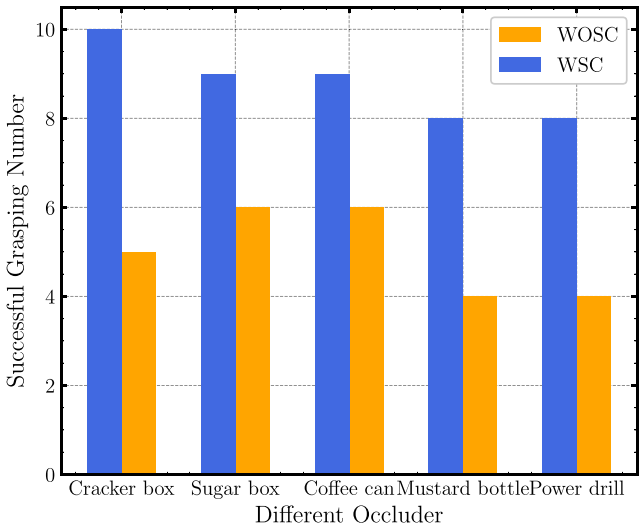

Fig. 7 Grasping performance comparison when target object is behind different occluders

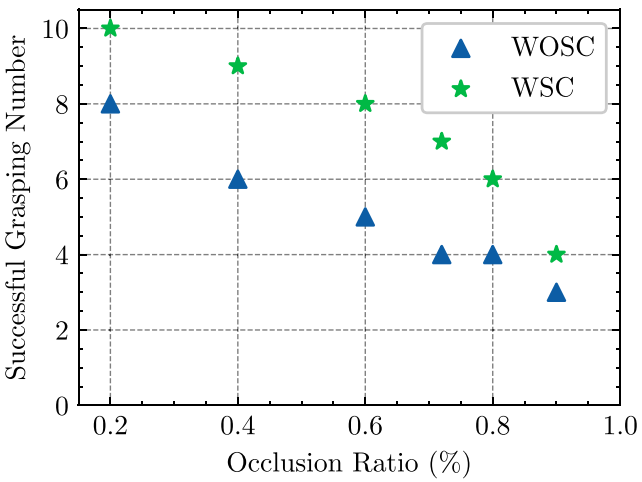

Fig. 8 Grasping performance comparison when target object is in different occlusion ratio

occlude the target object (bleach cleanser). The average successful grasping rate of WSC is 88% while WOSC is 50%. It demonstrates that our shape completion method can significantly increase the successful grasping rate up to 32% comparing original grasping strategy. However, we found that some irregularly shaped objects like the Mustard bottle and Power drill will divide the original partial point cloud of the target object into multiple surface parts. Because PointNetGPD [15] cannot understand that these separated point clouds are from the same object, WOSC generates more wrong grasp candidates without our shape completion. Furthermore, we explored the effect of the occlusion ratio on the grasp performance through stacking different blocks in front of the target object as an occlusion. Because the target object and stacking blocks are all placed on the table vertically and the horizontal length of each block is bigger than the maximum horizontal width of target object, the occlusion ratio is calculated through measuring the vertical height of stacking blocks ($\mathcal{H}_b$) and target object ($\mathcal{H}_t$). As seen from Fig. 7, we conducted six experiments with an occlusion rate between 0.2 and 0.9 to compare the two methods. When the occlusion ratio is less than 0.6, the grasping success rate of WSC is significantly improved over that of WOSC. However, because there are few high occlusion scenes in the YCB video dataset, it is still difficult for TransSC to repair the partial point cloud, especially when the occlusion ratio is higher than 0.8. Furthermore, when the occlusion ratio is between 0.8 and 1.0, it means that target object has been completely obscured. The vision information of target object is too little, so it's also much difficult to use shape completion to restore complete object information. According to our observation in daily life, we found 0.2-0.6 is also the most common object occlusion ratio and our experiments showed that our shape completion method could improve successful rate within this range (Fig. 8).

6 Conclusion and Future Work

We present a novel transformer-based sparse shape completion network (TransSC). This network includes a transformer-based encoder and manifold-based decoder that we designed, enabling our model to achieve a great completion result and outperform other representative methods. The experiments show that our network is robust to sparse and noisy point cloud input. Besides, simulation grasping experiments show our model could achieve a smaller grasp joint error than traditional robotic completion methods. Finally, when executing real robotic experiments of single objects and object occlusion, we demonstrate that our TransSC can be easily embedded into a existing grasp evaluation module and improve grasping performance significantly in both scenes.

The lack of object geometric information in our dataset is due to the change of the camera viewpoint and the occlusion by different objects. Thus, our grasp pipeline can solve both situations occurring in the grasping task successfully. However, similar to the research issue of 6 DoF pose estimation, it is still challenging to achieve shape completion of an arbitrary object at an arbitrary pose due to the limited object categories in our dataset. So the main limitation in this paper is that the object catogories in our constructed datatset are still small and they only limited in the YCB objects, which causing that our shape completion model cannot be generalized into other novel objects. In future work, our goal is to collect more objects categories to achieve a better generalization for unseen but similarly shaped objects. Furthermore, we will also consider more data augmentation strategies like adding more data representing different object 6 DoF pose and different point cloud missing ratio as our experiments shown, we think our shape completion model could achieve a better grasping performance in the real robotic experiments.

Supplementary Information The online version contains supplementary material available at https://doi.org/10.1007/s10846-022-01586-4.

Acknowledgements We thank Mech-Mind Robotics Company for providing the 3D camera.

Author Contributions List of all authors: Wenkai Chen, Hongzhuo Liang, Zhaopeng Chen, Fucun Sun and Jianwei Zhang.

All authors contributed to the conception and design of this manuscript. Technical work was conducted by Wenkai Chen and Hongzhuo Liang. The manuscript was revised by Zhaopeng Chen, Fucun Sun and Jianwei Zhang. All authors commented on the manuscript.

Funding Open Access funding enabled and organized by Projekt DEAL. This research was funded by the German Research Foundation (DFG) and the National Science Foundation of China (NSFC) in project Crossmodal Learning, DFG TRR-169/NSFC, project DEXMAN under grant 410816101 and partially supported by European projects H2020 Ultracept (778602).

Code or Data Availability Our code will be released at https://github.com/turbohiro/TransSC.

Declarations

Consent to Participate All participants consented to involve in the creation of this manuscript.

Conflict of Interests There are no conflicts of interest.

Consent for Publication All participants consented to publish this manuscript.

References

1. Achlioptas, P., Diamanti, O., Mitliagkas, I., Guibas, L.: Learning representations and generative models for 3D point clouds. In: International Conference on Machine Learning, pp. 40–49. PMLR (2018)
2. Badrinarayanan, V., Kendall, A., Cipolla, R.: Segnet: a deep convolutional encoder-decoder architecture for image segmentation. IEEE Trans. Pattern Anal. Mach. Intell. **39**(12), 2481–2495 (2017)
3. Bohg, J., Johnson-Roberson, M., León, B., Felip, J., Gratal, X., Bergström, N., Kragic, D., Morales, A.: Mind the gap-robotic grasping under incomplete observation. In: 2011 IEEE International Conference on Robotics and Automation, pp. 686–693. IEEE (2011)
4. Breyer, M., Chung, J.J., Ott, L., Siegwart, R., Nieto, J.: Volumetric grasping network: real-time 6 dof grasp detection in clutter. arXiv:2101.01132 (2021)
5. Chu, F.J., Xu, R., Vela, P.A.: Real-world multiobject, multigrasp detection. IEEE Robot. Autom. Lett. **3**(4), 3355–3362 (2018)
6. Edelsbrunner, H., Kirkpatrick, D., Seidel, R.: On the shape of a set of points in the plane. IEEE Trans. Inf. Theory **29**(4), 551–559 (1983)
7. 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)
8. Görner, M., Haschke, R., Ritter, H., Zhang, J.: Moveit! Task constructor for task-level motion planning. In: IEEE International Conference on Robotics and Automation (ICRA), pp. 190–196 (2019)
9. Groueix, T., Fisher, M., Kim, V., Russell, B., Aubry, M.: Atlasnet: a papier-mâché approach to learning 3D surface generation. arXiv:1802.05384 **11** (2018)
10. Gualtieri, M., Platt, R.: Robotic pick-and-place with uncertain object instance segmentation and shape completion. arXiv:2101.11605 (2021)
11. Guo, M.H., Cai, J.X., Liu, Z.N., Mu, T.J., Martin, R.R., Hu, S.M.: Pct: point cloud transformer. arXiv:2012.09688 (2020)
12. Huang, Z., Yu, Y., Xu, J., Ni, F., Le, X.: Pf-net: point fractal network for 3d point cloud completion. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 7662–7670 (2020)
13. Kuang, H., Wang, B., An, J., Zhang, M., Zhang, Z.: Voxel-FPN: multi-scale voxel feature aggregation for 3d object detection from lidar point clouds. Sensors **20**(3), 704 (2020)
14. Levine, S., Pastor, P., Krizhevsky, A., Ibarz, J., Quillen, D.: Learning hand-eye coordination for robotic grasping with deep learning and large-scale data collection. Int. J. Rob. Res. **37**(4-5), 421–436 (2018)
15. Liang, H., Ma, X., Li, S., Görner, M., Tang, S., Fang, B., Sun, F., Zhang, J.: PointnetGPD: detecting grasp configurations from point sets. In: 2019 International Conference on Robotics and Automation (ICRA), pp. 3629–3635. IEEE (2019)
16. Liu, M., Sheng, L., Yang, S., Shao, J., Hu, S.M.: Morphing and sampling network for dense point cloud completion. In: Proceedings of the AAAI Conference on Artificial Intelligence, pp. 11596–11603 (2020)
17. Lundell, J., Verdoja, F., Kyrki, V.: Robust grasp planning over uncertain shape completions. In: 2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 1526–1532. IEEE (2019)
18. Lundell, J., Verdoja, F., Kyrki, V.: Beyond Top-Grasps through Scene Completion. In: 2020 IEEE International Conference on Robotics and Automation (ICRA), pp. 545–551. IEEE (2020)
19. Mahler, J., Liang, J., Niyaz, S., Laskey, M., Doan, R., Liu, X., Ojea, J.A., Goldberg, K.: Dex-net 2.0: deep learning to plan robust grasps with synthetic point clouds and analytic grasp metrics. arXiv:1703.09312 (2017)
20. Miller, A.T., Allen, P.K.: Graspit! a versatile simulator for robotic grasping. IEEE Robot. Autom. Mag. **11**(4), 110–122 (2004)
21. Mousavian, A., Eppner, C., Fox, D.: 6-Dof graspnet: variational grasp generation for object manipulation. In: IEEE/CVF International Conference on Computer Vision (ICCV), pp. 2901–2910 (2019)
22. ten Pas, A., Gualtieri, M., Saenko, K., Platt, R.: Grasp pose detection in point clouds. Int. J. Rob. Res. **36**(13-14), 1455–1473 (2017)
23. ten Pas, A., Platt, R.: Using geometry to detect grasp poses in 3d point clouds. In: Robotics Research, pp. 307–324. Springer International Publishing. https://doi.org/10.1007/978-3-319-51532-8_19 (2018)
24. 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)
25. Srinivas, A., Lin, T.Y., Parmar, N., Shlens, J., Abbeel, P., Vaswani, A.: Bottleneck transformers for visual recognition. arXiv:2101.11605 (2021)
26. Tosun, T., Yang, D., Eisner, B., Isler, V., Lee, D.: Robotic grasping through combined image-based grasp proposal and 3d reconstruction. arXiv:2003.01649 (2020)
27. Varley, J., DeChant, C., Richardson, A., Ruales, J., Allen, P.: Shape completion enabled robotic grasping. In: IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 2442–2447 (2017)
28. Varley, J., Weisz, J., Weiss, J., Allen, P.: Generating multi-fingered robotic grasps via deep learning. In: 2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 4415–4420. IEEE (2015)
29. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, L., Polosukhin, I.: Attention is all you need. arXiv:1706.03762 (2017)
30. Watkins-Valls, D., Varley, J., Allen, P.: Multi-modal geometric learning for grasping and manipulation. In: 2019 International Conference on Robotics and Automation (ICRA), pp. 7339–7345. IEEE (2019)
31. Wu, C., Chen, J., Cao, Q., Zhang, J., Tai, Y., Sun, L., Jia, K.: Grasp proposal networks: an end-to-end solution for visual learning of robotic grasps. arXiv:2009.12606 (2020)
32. Wu, Z., Song, S., Khosla, A., Yu, F., Zhang, L., Tang, X., Xiao, J.: 3d shapenets: a deep representation for volumetric shapes. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 1912–1920 (2015)
33. Xiang, Y., Schmidt, T., Narayanan, V., Fox, D.: Posecnn: a convolutional neural network for 6d object pose estimation in cluttered scenes. arXiv:1711.00199 (2017)

34. Xie, H., Yao, H., Zhou, S., Mao, J., Zhang, S., Sun, W.: Grnet: gridding residual network for dense point cloud completion. In: European Conference on Computer Vision, pp. 365–381. Springer (2020)
35. Yi, L., Kim, V.G., Ceylan, D., Shen, I.C., Yan, M., Su, H., Lu, C., Huang, Q., Sheffer, A., Guibas, L.: A scalable active framework for region annotation in 3D shape collections. ACM Transactions on Graphics (ToG) **35**(6), 1–12 (2016)
36. 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)

Wenkai Chen received the B.S. degree and M.Eng in Mechanical Engineering from Wuhan University and Shanghai Jiao Tong University, China. He is currently pursuing a Ph.D. degree from the faculty of informatics, Universität Hamburg, Germany. His research interests include 3D computer vision, robotic grasping, and crossmodal learning in the robotic field.

Hongzhuo Liang received M.Eng in Mechanical Engineering from Anhui University of Technology, China. He is currently pursuing a Ph.D. degree from the faculty of informatics, Universität Hamburg, Germany. His research interests include crossmodal learning and reinforcement learning in robotic grasping and manipulation.

Zhaopeng Chen received B.S. degree in Mechanical Engineering and Automation and M.S. degree in Mechatronic Engineering from Harbin Institute of Technology, Harbin, China, in 2005 and 2007, respectively. From 2008 to 2010, he was a joint-train Ph.D. candidate in the Institute of Robotics and Mechatronics, DLR German Aerospace Center, Oberpfaffenhofen, Germany. Since 2010, he has been with DLR German Aerospace Center in Oberpfaffenhofen, Germany, as a research scientist. Now, he is the CEO of Agile Robots AG.

Fuchun Sun received the B.S. and M.S. degrees from the Naval Aeronautical Engineering Academy, Yantai, China, in 1986 and 1989, respectively, and the Ph.D. degree from Tsinghua University, Beijing, China, in 1998. He was with the Department of Automatic Control, Naval Aeronautical Engineering Academy, for over four years. From 1998 to 2000, he was a Postdoctoral Fellow with the Department of Automation, Tsinghua University. He is currently a Professor with the Department of Computer Science and Technology, Tsinghua University. His research interests include intelligent control, neural networks, fuzzy systems, variable structure control, nonlinear systems, and robotics. Dr. Sun was the recipient of the Doctoral Dissertation Prize of China in 2000.

Jianwei Zhang (Member, IEEE) received the B.Eng. (Hons.) and M.Eng. degrees from the Department of Computer Science, Tsinghua University, Beijing, China, in 1986 and 1989, respectively, the Ph.D. degree from the Department of Computer Science, Institute of Real-Time Computer Systems and Robotics, University of Karlsruhe, Karlsruhe, Germany, in 1994, and the Habilitation degree from the Faculty of Technology, Bielefeld University, Bielefeld, Germany, in 2000., He is currently a Professor and the Head of the TAMS Group, Department of Informatics, University of Hamburg, Hamburg, Germany. He has been coordinating numerous collaborative research projects of EU and the German Research Council, including the Transregio-SFB TRR 169 "Crossmodal Learning". He has published about 300 journal articles and conference papers (winning four best paper awards), technical reports, 4 book chapters, and 5 research monographs. His current research interests include cognitive robotics, sensor fusion, dexterous manipulation, and multimodal robot learning., Dr. Zhang is a Life-Long Academician of the Academy of Sciences, Hamburg. He is the General Chair of the IEEE MFI 2012, the IEEE/RSJ IROS 2015, and the IEEE Robotics and Automation Society AdCom, from 2013 to 2015.