Neurocomputing 507 (2022) 221–234

Contents lists available at ScienceDirect

Neurocomputing

journal homepage: www.elsevier.com/locate/neucom

Cosmos Propagation Network: Deep learning model for point cloud completion ☆

Fangzhou Lin [a,*,1], Yajun Xu [b,1], Ziming Zhang [c], Chenyang Gao [d], Kazunori D Yamada [a,*]

[a] Department of Applied Information Sciences, Graduate School of Information Sciences, Tohoku University, Sendai, Miyagi 980-8579, Japan
[b] Department of Systems Science and Informatics, Faculty of Information Science and Technology, Hokkaido University, Sapporo, Hokkaido 060-0814, Japan
[c] ECE Department & Data Science & Robotics Engineering, Worcester Polytechnic Institute, Worcester, MA 01609, USA
[d] Department of Computer Science and Engineering, Southern University of Science and Technology, Shenzhen 518-055, China

ARTICLE INFO

Article history:
Received 12 July 2021
Revised 24 June 2022
Accepted 3 August 2022
Available online 8 August 2022
Communicated by Zidong Wang

Keywords:
Shape completion
3D Point cloud
Deep learning
Neural networks
Point propagation

ABSTRACT

Point clouds measured by 3D scanning devices often have partially missing data due to the view positioning of the scanner. The missing data can reduce the performance of a point cloud in downstream tasks such as segmentation, location, and pose estimation. Consequently, 3D point cloud completion aims to predict the missing regions of incomplete objects for these fundamental 3D vision tasks. However, predicting the complete object can easily diminish the detail or structure of a measured region, which usually does not require repair. This study proposes a novel neural network architecture, Cosmos Propagation Network (CP-Net), for 3D point cloud completion. CP-Net extracts latent features in different scales from incomplete point clouds used as input. For point cloud generation, we propose a novel point expand method using a Mirror Expand module. Compared with existing methods, our Mirror Expand module introduces less information redundancy, which makes the distribution of points more reliable. CP-Net predicts the details of missing regions and maintains a clear general structure. The performance of CP-Net on several benchmarks was compared to that of current baseline methods. Compared to the existing methods, CP-Net showed the best performance for various metrics. Thus, CP-Net is expected to help address various problems related to 3D point cloud completion. Its source code is available at https://github.com/ark1234/CP-Net.

1. Introduction

Point clouds are one of the most popular tools for 3D representation and are becoming increasingly easy to obtain with the increased availability of 3D scanning devices [1–3]. Further, 3D point clouds are widely used in industrial [4] and commercial [5] application&s. However, occlusions, light reflections, limited viewing angles, and surface material properties often cause scanned point clouds to be incomplete [6–8]. Therefore, for further application, the missing points in incomplete point clouds must be deduced from raw input such as mesh [9], depth images [10], and other point clouds [11].

Generally, the solutions for 3D point cloud completion problems rely heavily on sophisticated mathematical methods [12–15]. However, such approaches require high-level mathematics and impose significant restrictions on the application. For example, these algorithms are often limited to filling holes [12,13]. Meanwhile, researchers are focusing on developing learning-based methods such as artificial intelligence by deep learning owing to the effectiveness of such methods. Early learning-based methods using point clouds first converted the point cloud to some regular format, such as a depth map [16] or voxels [17]. This allowed conventional 2D or 3D convolutional neural networks (CNNs) to be applied for these data types, enabling these methods [16–18] to leverage the effectiveness of CNNs in 2D image understanding [19]. Varley et al. [18] first segmented and meshed scanned point clouds, after which a fast mesh completion method was employed. However, such conversion methods not only incur high computational costs and high sparsity of volumetric data but also cause some loss of information, which severely reduces the details of

☆ Computations were partially performed on the NIG supercomputer at ROIS National Institute of Genetics. This work was supported in part by the Top Global University Project from the Ministry of Education, Culture, Sports, Science, and Technology of Japan (MEXT). Dr. Ziming Zhang was supported partially by NSF CCF-2006738.

* Corresponding authors.
E-mail addresses: lin.fangzhou.p2@dc.tohoku.ac.jp (F. Lin), y_xu@sdm.ssi.ist.hokudai.ac.jp (Y. Xu), zzhang15@wpi.edu (Z. Zhang), 11710904@mail.sustech.edu.cn (C. Gao), kyamada@ecei.tohoku.ac.jp (K.D Yamada).

[1] Equal contributions.

https://doi.org/10.1016/j.neucom.2022.08.007
0925-2312/

the local structure [20,21]. Therefore, researchers have tended to approach this task using point clouds without prior conversion.

Recently, researchers have begun using generative adversarial networks (GANs) [22] to repair incomplete point clouds. Two main reasons have driven this change in approach. First, GAN-based methods have achieved remarkable performance for 2D image inpainting [23–26]. Second, the pioneering contribution of PointNet [27] has enabled direct processing for point clouds. Several researchers have proposed full point cloud prediction networks [20,28–30] to generate the point clouds of an entire object rather than the missing points alone. However, with the continuous improvement of device performance, the acquired regions of a point cloud can be considered accurate, and missing regions of a point cloud predominantly occur due to camera positioning. Thus, as shown in Fig. 1, it is generally more reasonable to predict only the missing regions. With this consideration, Yu et al. [11], Huang et al. [31], and Alliegro et al. [6] adopted this approach and predicted only the missing regions of point clouds. However, their methods are constrained by a limited ability to augment the feature information of point clouds [31,11]. For example, the extra T-Net from PointNet [32] used in Point Encoder GAN [11] involves unnecessary computation, and the simple deconvolution layer in the generator fails to predict the details of missing regions. PF-Net [31] generates excessive redundancy in point clouds owing to the direct expansion operation, which increases the amount of duplicated information. The point cloud generation procedure also does not fully release the information obtained from the latent features. Therefore, latent feature extraction and point cloud generation still have significant room for improvement.

To overcome these problems, we propose the Cosmos Propagation Network (CP-Net), a novel network structure that can not only extract features on multiple levels but can also effectively propagate point clouds. Unlike existing methods that roughly expand the feature information through duplicated feature information [31], our expansion can introduce far less duplicated information and effectively improves the local and overall point cloud structure reconstruction results. CP-Net can predict the missing portions of a point cloud using the incomplete point cloud as input. Our contributions are as follows:

- We developed the Multi-Edge Encoder (MEE), which exploits raw input point cloud features with a low computational cost while maintaining performance.
- We propose the Point Expand Decoder (PED) with the Mirror Expand module, which improves the redundancy of point cloud generation and predicts fine-grained missing point clouds while retaining the original input point clouds.
- The results of evaluations conducted on several benchmarks show both quantitatively and qualitatively that the performance of CP-Net is comparable to that of several state-of-art baselines.

The remainder of this article is organized as follows. Section 2 discusses related work. Section 3 presents the details of the network structure and its related modules. Section 4 provides the experimental details, related discussion, quantitative evaluation of the experimental results, and performance evaluations for each component through ablation studies conducted on the benchmark ShapeNet-Part [33]. Finally, Section 5 provides concluding remarks and directions for future work.

2. Related work

The GAN architecture is currently used for popular point cloud completion methods. It contains a generator and a discriminator [22]. Researchers are currently focusing on the generator's design because improving its performance is related to improving the performance of solving point cloud completion. The generator for point cloud completion methods includes encoder and decoder components.

2.1. Encoder: Features Extraction

The encoder is used to extract the latent feature of a whole object [34]. PointNet [32] is a relatively good feature extraction encoder and has been implemented in [20,28,11]. In PointNet, the encoder attaches symmetrical operations, such as max-pooling and average pooling at the end of multilayer perceptron (MLP) layers, to aggregate the global features from unordered and sparse 3D point clouds. However, the local information from points cannot be obtained because the features are learned individually among every point through PointNet [35]. Therefore, Huang et al. [31] proposed a Multi-Resolution Encoder to extract local features better.

Meanwhile, some researchers rely on graph-based architectures to obtain a better point cloud feature. In DGCNN [36], the graph was built in the feature space and dynamically updated in each network layer sequentially. However, the memory usage is high for DGCNN with relatively high computational costs [37]. To overcome this problem and maintain the performance of the feature obtainer, we introduce MEE with Dense Edge-Conv (DEC) modules. MEE can capture the features hierarchically at a relatively low computational cost.

2.2. Decoder: Point Generation

The decoder uses the latent feature obtained from the encoder for the point cloud completion. Yang et al. [38] introduced a folding-based decoder to deform a 2D plane into a 3D surface. This folding operation essentially constructs a 2D regular domain to 3D point clouds, which can achieve low reconstruction errors. The operation repeats obtained latent features into a target point cloud of the same number, then concatenates the artificially designed 2D coordinate for each feature. Then, this intermediate 3D coordinate and the repeated latent feature are concatenated again by folding into the 3D coordinates, and the folding operation is repeated a second time to obtain the final reconstructed 3D point clouds. PCN [20] inherited this folding operation [38] as the last stage of its network's decoder, allowing it to generate a smooth and dense point cloud with fewer parameters than a fully connected decoder. However, because the latent feature is naively repeated several times, the information is duplicated for most of the 3D coordinates, making it difficult for the network to distinguish the expansion of different 3D point clouds. Subsequently, Wen et al. [39] introduced a Structure-Preserving Decoder that can progressively refine the point cloud at different resolutions with hierarchical folding operations, hierarchically preserving the complete shape structure at different resolutions.

Apart from the implemented manifold-based folding operations in the decoder design, TopNet [28] introduced a tree structure to generate point clouds. Its Rooted Tree Decoder has a rooted tree structure where the root node embeds and processes global point cloud embedding representing 3D shapes. Although it can achieve an excellent qualitative result, the fine details of the local structure may be lost [29].

Point Fractal Network (PF-Net) [40] has recently focused on predicting only the missing region and has achieved good completion results. As previously mentioned, they designed a Multi-Resolution Encoder to obtain both local and global latent features from incomplete raw input. Afterwards, their Point Pyramid Decoder (PPD) introduced a simple but effective way to generate point clouds in

Fig. 1. Example of why partial point clouds of the object cannot be obtained along with the prediction of missing regions. Here, the gray dots are the partial point clouds obtained from the 3D scanner, and the yellow dots predict missing regions.

a fractal order. It first directly generates a coarse, low-resolution point cloud by using the latent feature and then appends a different offset to each point in the point cloud to refine the point cloud and increase its resolution. However, PF-Net is copied on the original coarse point cloud before offset expansion. This operation leads to the duplication of about half of the point cloud information during the expansion process, which affects the prediction of the final point cloud. As for this reference, CP-Net provides the Mirror Expand module, which generates the corresponding potential points near each point in the point cloud for expansion. This increases the differentiation of the coarse point cloud and provides a more elegant method for missing region point cloud propagation (expansion).

3. Method

This section introduces CP-Net, which generates the missing regions of 3D point clouds from partial input. Fig. 2 shows the architecture of CP-Net, which consists of MEE and PED.

3.1. Motivation

The 3D point cloud completion task generates/predicts the missing regions of point clouds to design a good model for point cloud completion. Understanding input partial point clouds and a better point cloud generation algorithm are crucial to conquer this task. However, many existing end-to-end methods either fail to obtain rich feature information from input partial point clouds or only predict unsatisfied missing regions [20,38,11]. A literature survey [20,39,11,31] found that the mainstream methods for generating point clouds rely on simply duplicating the feature information to obtain low-resolution points or directly generating final prediction results from latent features. These methods introduce much redundant information through duplication, which could increase the difficulty of follow-up refinement in point clouds. To address these expansion problems, we first generate corresponding points in the neighborhood of every point with an offset. By so doing, new points created through expansion are related to the previous step's results without duplicated

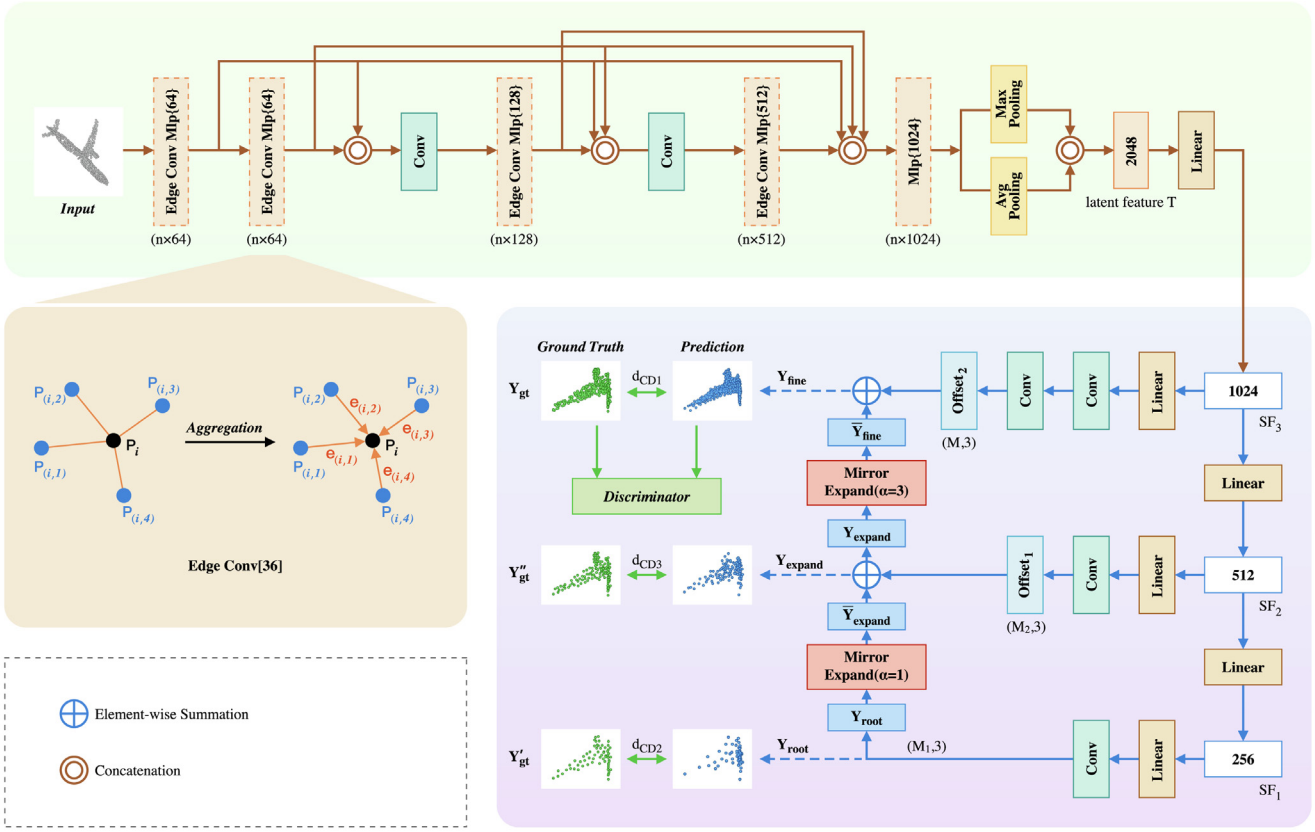

Fig. 2. Architecture of CP-Net. Partial point clouds are used as input to extract features through the MEE (Section 3.2) block. A multi-stage propagating PED (Section 3.3) predicts missing regions at three scales: two coarse predictions and one final prediction. Chamfer Distance (CD) is used to evaluate the difference between the ground truth and prediction during the training procedure. The operation of Edge-Conv is illustrated in detail in the bottom left corner.

information. We expand and refine points several times from low resolution to high resolution, ensuring the distribution of generated points is meaningful and reliable.

3.2. Multi-Edge Encoder

The Dense Edge-Conv (DEC) module constitutes the backbone of MEE. Wang et al. [36] constructed a local graph in the Edge-Conv layer with the k-nearest neighbors (kNN) algorithm in one Euclidean space and two high-dimensional spaces, which leads to high computational cost and memory usage. For the Edge-Conv [36], as illustrated in the bottom-left of Fig. 2, neighborhood points $\mathbf{P}_{(i,j)}$ aggregate related edge feature $\mathbf{e}_{(i,j)}$ through channel-wise symmetric aggregation operation; in the experimental setting, this is $j = 0, 1, \ldots, 16$. Similarly, we apply the Edge-Conv layer to build the local graph in our DEC module. We only define it in the first Euclidean space to reduce the computational cost and guarantee performance. In addition, we connect each layer using shortcuts so that the feature information of different layers can be fully released and utilized. The details of the computational cost analysis can be found in Section 4.3.

The input to MEE is $N \times 3$ unordered 3D point clouds. We directly feed the partial 3D point clouds into the first Edge-Conv layer. Starting with the output of the second layer, the output of each layer is concatenated with the previous layer's output and fed into a shared MLP, which is then fed into the next layer. This arrangement ensures that low-level feature information is properly retained to strengthen the connection between each feature at different levels. This is why we called the module DEC, as it uses extra shortcut operations (skip links) and convolutional operations to make more dense connections between each layer. As a benefit of this design, we can obtain multiple dimensional feature tensors V_i, where size $\mathrm{V}_i := N \times 64, N \times 64, N \times 128, N \times 512$, for $i = 0, 1, 2, 3$. The V_i are then concatenated, forming the combined latent tensor $\mathbf{C}$. Size $\mathbf{C} := N \times 768$, and it contains different levels of feature information. We then use MLP followed by max- and avg-pooling operations to integrate the combined tensors into latent feature $\mathbf{T}$ with size $2,048$.

3.3. Point Expand Decoder

The goal of the decoder is to generate partial point clouds from latent feature $\mathbf{T}$, which represents the shape of the missing region. PED is based on the point pyramid decoder [31]. A point pyramid decoder provides a multi-scale generating architecture, leading to a new perspective to propagate information. Nevertheless, the decoder implements duplicated expansion in the original points, creating redundant information that is difficult to refine. Moreover, a limited number of points requires more ways to expand. To overcome those limitations, we design our PED with the Mirror Expand module to make the procedure of point cloud propagation smoother and more meaningful. The details of the PED can be found in the bottom-right of Fig. 2.

As shown in the bottom-right block of Fig. 2, in the first step, latent feature $\mathbf{T}$ is converted into three smaller feature tensors (size $\mathbf{SF}_i := 256, 512, 1024$, for $i = 1, 2, 3$) through fully-connected layers. $\mathbf{SF}_1$ is the seed of feature information expansion, whereas $\mathbf{SF}_2$ and $\mathbf{SF}_3$ function as the refined information. Following with the pipeline of Fig. 2, in the second step, $\mathbf{SF}_1$ is converted to $Y_{\mathrm{root}} \in \mathbb{R}^{M_1 \times 3}$ through a fully-connected layer and a convolutional layer sequentially. Meanwhile, Y_{root} feeds into stage one of the Mirror Expand module to expand once ($\alpha = 1$) as $\overline{Y}_{\mathrm{expand}} \in \mathbb{R}^{M_2 \times 3}$. In the third step, $\mathbf{SF}_2$ is converted to $\mathbf{Offset}_1 \in \mathbb{R}^{M_2 \times 3}$, which functions as a deviation parameter to fine tune the expanded point clouds $\overline{Y}_{\mathrm{expand}}$ through the Element-wise Summation operation to become fine-tuned $Y_{\mathrm{expand}} \in \mathbb{R}^{M_2 \times 3}$. In the last step, Y_{expand} repeats this procedure again in the stage two Mirror Expand module to expand thrice ($\alpha = 3$) as $\overline{Y}_{\mathrm{fine}} \in \mathbb{R}^{M \times 3}$. Meanwhile, $\mathbf{SF}_3$ is converted to $\mathbf{Offset}_2 \in \mathbb{R}^{M \times 3}$. Similarly, $\overline{Y}_{\mathrm{fine}}$ then fine-tunes again with $\mathbf{Offset}_2$ to become the final prediction $Y_{\mathrm{fine}} \in \mathbb{R}^{M \times 3}$.

The **Mirror Expand module** is the core of PED and provides a valuable tool to aggregate the point clouds with the connection of neighborhood feature information. In this module, the graph-based points aggregation algorithm exploits neighborhood points and feature information to further enhance the performance of missing point clouds prediction. As shown in Fig. 3, we consider the blue and red balls as the points obtained from the encoder, where each ball represents a point cloud. The kNN algorithm is implemented here for finding the k-nearest nodes in the whole tensor set. The mirroring operation generates α new centrosymmetric nodes with each centroid. Eventually, the tensor set is expanded by α. The points expanded through the Mirror Expand operation are used to refine the whole point clouds. With the second expansion iteration, the point clouds are further refined. We use two iterations of this operation to aggregate sufficiently rich feature information. The Mirror Expand operation not only expands the number of features but also introduces new information for point clouds generation. In other words, the new point clouds generated through the Mirror Expand operation become the seed for aiming the prediction of point clouds. The network proceeds through backpropagation in the training procedure to learn the actual position distribution of the point clouds, which can generate better point clouds. The Mirror Expand operation can provide the point clouds with better initial distribution, making it easier to obtain further information from deeper layers. Hence, the PED can predict point clouds with finer details.

3.4. Training Losses

During training, the multi-stage completion loss L_{com} and adversarial loss L_{adv} are adopted as the optimization losses. Completion loss tries to match the prediction to the ground truth of

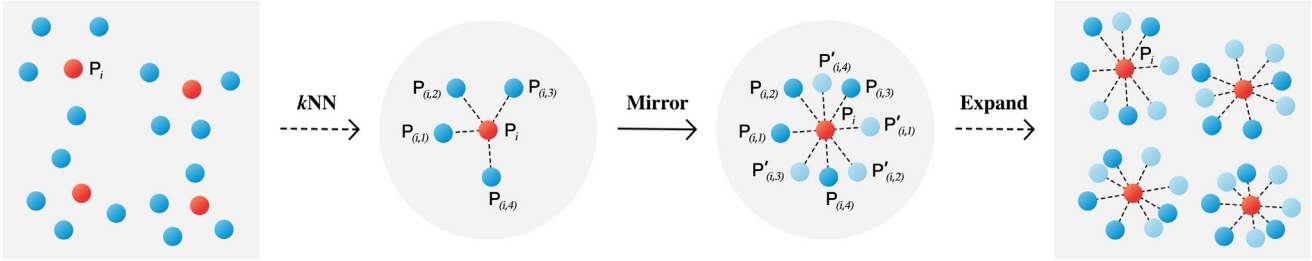

Fig. 3. Mirror Expand module pipeline. The balls in the left box represent the point clouds in the Euclidean space. With the implementation of kNN, central point cloud $\mathbf{P}_i$ (red) finds k (here, k is four) nearest neighborhood points $\mathbf{P}_{(i,j)}$ (blue). Subsequently, the Mirror Expand operation expands the number of points k times. New points $\mathbf{P}\prime_{(i,j)}$ in light blue are symmetrical about $\mathbf{P}_i$. Here, the right-side box only shows four points' expansion, and we implement this operation in the whole point clouds.

missing point cloud Y_{gt}. Adversarial loss tries to make the prediction approach the ground truth by optimizing the MEE and PED.

Similar to PF-Net [31], the total loss for training is the weighted sum of L_{com} and L_{adv} is defined as follows:

$$L = \lambda_{com} L_{com} + \lambda_{adv} L_{adv}, \tag{1}$$

where λ_{com} and λ_{adv} are the weight of completion loss and adversarial loss, which satisfy the following condition: $\lambda_{com} + \lambda_{adv} = 1$.

L_{com} is combined with different resolutions of the output of PED: $Y_{root}, Y_{expand}, Y_{fine}$. As shown below, they are used to compute the CD (the details of d_{CD} can be found in Section 4.2) with the ground- truth of the missing region Y_{gt} and the sampled ground- truth of $Y\prime_{gt}, Y\prime\prime_{gt}$ as d_{CD_1}, d_{CD_2}, and d_{CD_3}, weighted by hyper-parameter α:

$$\begin{aligned} L_{com} = {} & d_{CD1}(Y_{fine}, Y_{gt}) + \alpha\, d_{CD2}(Y_{root}, Y\prime_{gt}) \\ & + 2\alpha\, d_{CD3}(Y_{expand}, Y\prime\prime_{gt}), \end{aligned} \tag{2}$$

The adversarial loss L_{adv} is inherited from GAN [22]. We define $F() := PED(MEE())$. $F : \mathscr{X} \rightarrow \mathscr{Y}\prime$ maps the partial input $\mathscr{X}$ into the predicted missing region $\mathscr{Y}\prime$. Then, the discriminator D tries to distinguish the predicted missing region $\mathscr{Y}\prime$ and the real missing region $\mathscr{Y}$. The discriminator is a classification network with serial MLP layers.

L_{adv} is defined as

$$L_{adv} = \sum_{i \in S} \log(D(y_i)) + \sum_{j \in S} \log(1 - D(F(x_i))), \tag{3}$$

where $x_i \in \mathscr{X}, y_i \in \mathscr{Y}, i = 1, \ldots, S$. S is the dataset's size of $\mathscr{X}, \mathscr{Y}$.

4. Experimental details

4.1. Datasets

ShapeNet-Part. For the training of our model, we used the benchmark ShapeNet-Part [33] dataset, which provides part segmentation to a subset of ShapeNetCore [41] models. It contains 16 categories of different models (shapes). The total number of shapes sums to 17,775. The ground truth point cloud data was created by sampling 2,048 points uniformly on each shape. We used the same setting in PF-Net [31] for a fair comparison. The partial point cloud data were generated by randomly selecting a viewpoint as a center among multiple viewpoints and removing points within a certain radius from the complete data. The missing number of points in the point clouds was 512. We called this the known categories dataset and used a train-test ratio of 15236 : 2539 (roughly 6 : 1).

To further explore the robustness and generality of the network, we designed a selection of novel categories. For the novel categories, we also selected 16 categories from the benchmark ShapeNet-Part [33] but with a different arrangement. Eight categories were used for training, and the other eight for testing. The training and testing objects were from different categories, and the train-test ratio was 15998 : 1777 (roughly 9 : 1).

ShapeNet. PCN [20] introduced the benchmark ShapeNet dataset [42] for point cloud completion. It consists of 30,974 3D models from eight categories. The ground truth point clouds containing 16,384 points are uniformly sampled on mesh surfaces. We used the same partial point cloud generation method as in PF-Net [31]; with the same ratio of 4,096, it can further validate the performance of our CP-Net in processing dense point clouds. For a fair comparison, we sampled the input partial point clouds to 2,048, the same as in PCN, and kept the same train/val/test splits as PCN.

Completion3D. Completion3D benchmark [28,20,41] is an online benchmark. It comprises 28,974 and 800 samples for training and validation, respectively. There are 2,048 points in the ground truth point clouds. We used their validation set to test the performance of our CP-Net.

KITTI. The benchmark KITTI dataset [43] is composed of a sequence of real-world Velodyne LiDAR scans derived from PCN [20]. For each frame, the car objects are extracted according to the 3D bounding boxes containing 2,401 objects of partial point clouds. The partial point clouds in KITTI are highly sparse and do not have complete point clouds as ground truth.

4.2. Evaluation Metrics

Let $\mathscr{T} = \{(x_i, y_i, z_i)\}_{i=1}^{n_{\mathscr{T}}}$ be the ground truth and $\mathscr{R} = \{(x_i, y_i, z_i)\}_{i=1}^{n_{\mathscr{R}}}$ be a reconstructed point set being evaluated, where $n_{\mathscr{T}}$ and $n_{\mathscr{R}}$ are the numbers of points of $\mathscr{T}$ and $\mathscr{R}$, respectively. In our experiments, we used CD and F-score as quantitative evaluation metrics.

Chamfer Distance. Following TopNet [28] and Gridding Residual Network (GRNet) [44], the distance between $\mathscr{T}$ and $\mathscr{R}$ is defined as

$$d_{CD} = \frac{1}{n_{\mathscr{T}}} \sum_{t \in \mathscr{T}} \min_{r \in \mathscr{R}} ||t - r||_2^2 + \frac{1}{n_{\mathscr{R}}} \sum_{r \in \mathscr{R}} \min_{t \in \mathscr{T}} ||t - r||_2^2, \tag{4}$$

We follow the consideration in PF-Net [31], splitting the CD into two parts: Pred → GT (prediction to ground truth) error and GT → Pred (ground truth to prediction) error.

F-Score. F-Score is actively used in the multi-view 3D community [45]. Moreover, Tatarchenko et al. [46] pointed out that the CD may sometimes be misleading. As suggested in [46], we take F-Score as an extra metric to evaluate the performance of point completion results, which can be defined as follows:

$$F - score(d) = \frac{2P(d)R(d)}{P(d) + R(d)}, \tag{5}$$

where $P(d)$ and $R(d)$ denote the precision and recall for a distance threshold d, respectively:

$$P(d) = \frac{1}{n_{\mathscr{R}}} \sum_{r \in \mathscr{R}} \left(\min_{t \in \mathscr{T}} ||t - r||_2 < d \right), \tag{6}$$

$$R(d) = \frac{1}{n_{\mathscr{T}}} \sum_{t \in \mathscr{T}} \left(\min_{r \in \mathscr{R}} ||t - r||_2 < d \right), \tag{7}$$

Earth Mover's Distance (EMD) The EMD is a solution. The minimum cost to transport one shape's point cloud to another is an example of the EMD [47]. It is defined in Eq. (8), where ϕ is a bijection. The bijection is highly indicative of the uniformity of the generated shapes. This metric can reflect more reliable results in the shape completion task, but it has a relatively high computational cost, so we only implemented it in the testing period. We chose the approximation algorithm provided by the Morphing and Sampling Network (MSN) [48].

$$d_{EMD}(\mathscr{T}, \mathscr{R}) = \min_{\phi: \mathscr{T} \rightarrow \mathscr{R}} \frac{1}{|\mathscr{T}|} \sum_{t \in \mathscr{T}} ||t - \phi(t)||_2 \tag{8}$$

4.3. Computational Cost Analysis

We present the results of a computational cost analysis of several popular networks' encoders along with ours in Fig. 4. In the first three bars, we find that, compared with traditional PointNet's encoder, Edge-Conv Slayers increase the computational cost with more forward time and GPU memory cost. For our CP-Net's PED, the number of Edge-Conv layers is set to three with only one iteration of the kNN algorithm in the first Euclidean space to search

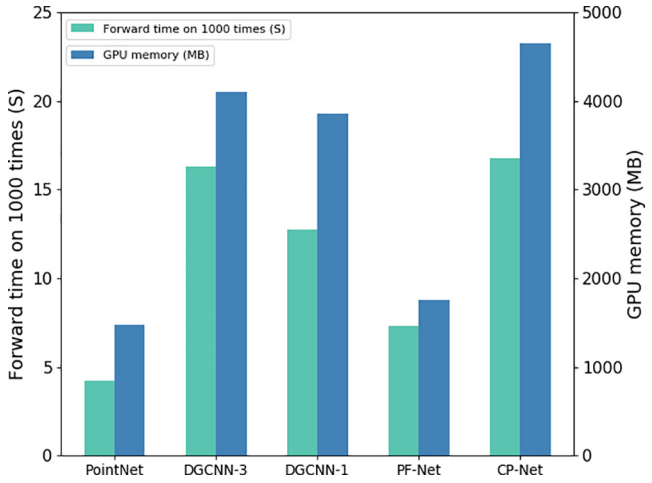

Fig. 4. The computational cost of different networks' feature extractor (encoder). For a fair comparison, the testing was conducted with the same settings.

for the edge information. This information is then shared with the other layers and represents a compromise between the performance and convolutional cost. This arrangement decreases the forward time and GPU memory cost, as shown in the second and third bars in Fig. 4. Here, DGCNN-3 represents the original Edge-Conv layers, and DGCNN-1 represents our PEDs. As shown in the last two bars in Fig. 4, the Edge-Conv layers and the related *k*NN algorithm result in an increase with an acceptable computational cost.

4.4. Implementation Details

We implemented our network using PyTorch [49] and CUDA [50]. All models were optimized using the ADAM optimizer [51] with an initial learning rate of 0.0001. For the prediction of different scales of point clouds, the settings were $\mathbf{M}_1 = 64, \mathbf{M}_2 = 128, \mathbf{M} = 512$ for the ShapeNet-Part, Completion3D, and KITTI datasets, and $\mathbf{M}_1 = 512, \mathbf{M}_2 = 1024$, and $\mathbf{M} = 4,096$ for the ShapeNet dataset. We trained the network with a batch size of 64 on four NVIDIA Tesla V100 GPUs on the ShapeNet-Part, Completion3D, and KITTI datasets and a batch size of 8 on one NVIDIA Tesla V100 GPU on the ShapeNet dataset. The optimization was set to stop after 200 epochs. Notably, we trained the network independently from scratch for all benchmarks.

4.5. Shape Completion on Known Categories

We compared our method against several current state-of-the-art point cloud completion methods: Skip-attention Network (SA-Net) [39], Morphing and Sampling Network (MSN) [48], Gridding Residual Network (GRNet) [44], and Point Fractal Network (PF-Net) [31]. For all these baselines, we ran the code provided by the respective authors to obtain quantitative and qualitative results. It is worth mentioning that SA-Net, MSN, and GRNet were designed to predict the whole point clouds, whereas PF-Net and our method were developed to predict the missing part, as mentioned previously. The previously mentioned evaluation metrics were used to evaluate the performance of each method.

Quantitative Results. Table 1 presents the completion results with the evaluation metrics of the CD, as introduced previously, and we show the two parts of the CD. The results demonstrate the superiority of CP-Net compared with all the baseline methods implemented.

As mentioned previously, one metric may not be sufficient to show the difference between each baseline and our method. Hence, we also present the quantitative results in terms of F-score@1% and EMD in Tables 2 and 3, respectively. For F-score@1%, a larger number indicates a better reconstruction result. The EMD is similar to the CD but is more dependable. The EMD can measure the completion task more closely to the actual visualization results [48]. Our method achieved a competitive score in terms of EMD.

Moreover, because CP-Net is designed for predicting the missing regions of point clouds, the quantitative results of missing regions help evaluate the network's performance. Table 4 presents the completion results with the evaluation metrics of the two parts of the CD. We also present the mean EMD and F-score@1% in Table 5. Because the results here only count the predicted point cloud region, we can use the numerical results in the above table to understand the more direct completion results. These results also fully reflect the superiority of our CP-Net.

Qualitative Results. Fig. 5 shows the point clouds reconstructed with different baseline methods and CP-Net. Our network produces a demonstrably well-reconstructed point cloud in general outline while maintaining the realistic details of the original ground truth. For example, on the cap point cloud, most of the baselines lose the curved structure of the cap. On the guitar point cloud, CP-Net is the only one that reconstructs the missing smooth edge of the object. On the lamp point cloud, CP-Net successfully predicted the general structure of the missing region while other

Table 1
Point cloud completion results of overall point cloud. The results consist of 16 categories of different objects. The numbers shown are [Pred → GT error / GT → Pred error], scaled by 1000. The mean values across all categories can be found in the final row of the table.

Category	SA-Net	GRNet	MSN	CP-Net	PF-Net
Airplane	0.671/0.895	1.002/0.607	0.680/1.039	**0.248**/**0.302**	0.278/0.305
Bag	2.618/2.758	3.005/3.208	2.851/5.948	**0.974**/**0.875**	1.027/0.894
Cap	2.223/1.853	1.122/3.294	2.841/3.121	**1.061**/**0.968**	1.526/1.077
Car	2.017/1.927	0.825/1.141	2.076/2.002	0.655/0.520	**0.630**/**0.488**
Chair	1.201/1.492	0.877/0.998	1.338/3.025	**0.480**/**0.462**	0.546/0.512
Earphone	2.836/2.580	3.118/3.916	3.052/4.761	1.350/**1.361**	**1.094**/2,219
Guitar	0.450/0.460	4.024/0.209	0.328/0.332	**0.105**/**0.121**	0.115/0.127
Knife	0.427/0.497	4.283/0.228	0.342/0.407	**0.114**/**0.143**	0.120/0.169
Lamp	1.810/1.593	5.703/1.717	2.267/5.130	**0.999**/**0.549**	1.095/0.739
Laptop	1.185/1.074	0.606/0.909	0.964/0.875	**0.313**/0.315	0.332/**0.307**
Motorbike	1.657/1.728	0.654/1.181	1.561/1.952	**0.527**/0.481	0.564/**0.444**
Mug	2.716/2.875	0.827/2.762	2.512/5.228	0.825/**0.917**	**0.763**/0.939
Pistol	1.037/1.066	0.950/0.418	0.938/0.979	**0.269**/**0.249**	0.315/0.304
Rocket	0.812/0.648	0.605/0.301	0.556/0.831	**0.246**/0.207	0.294/**0.185**
Skateboard	0.981/1.024	0.554/1.039	0.806/1.164	0.288/0.437	**0.282**/**0.339**
Table	2.025/1.688	0.618/1.148	1.533/3.016	**0.511**/**0.474**	0.575/0.479
Mean	1.480/1.456	2.023/1.055	1.386/2.637	**0.477**/**0.427**	0.517/0.463

Table 2
Point cloud completion results of overall point cloud with F-score@1%.

Category	SA-Net	GRNet	MSN	CP-Net	PF-Net
Airplane	0.798	0.691	0.830	**0.936**	0.935
Bag	0.409	0.662	0.350	**0.819**	0.818
Cap	0.344	0.689	0.458	**0.824**	0.821
Car	0.424	0.672	0.418	0.841	**0.848**
Chair	0.636	0.771	0.614	**0.874**	0.873
Earphone	0.471	0.693	0.406	0.836	**0.840**
Guitar	0.905	0.968	0.950	**0.979**	0.976
Knife	0.906	0.920	0.938	**0.972**	0.970
Lamp	0.654	0.673	0.617	**0.879**	0.870
Laptop	0.656	0.784	0.683	0.885	**0.886**
Motorbike	0.478	0.676	0.503	**0.860**	0.857
Mug	0.290	0.799	0.275	0.802	**0.803**
Pistol	0.696	0.674	0.733	**0.923**	0.919
Rocket	0.812	0.656	0.845	**0.930**	0.929
Skateboard	0.686	0.662	0.770	0.917	**0.919**
Table	0.593	0.570	0.611	0.888	**0.891**
Mean	0.642	0.684	0.653	**0.893**	0.890

Table 3
Point cloud completion results of overall point cloud with EMD (scaled by 100).

Category	SA-Net	GRNet	MSN	CP-Net	PF-Net
Airplane	3.849	3.269	3.361	**0.763**	0.913
Bag	6.498	5.456	6.650	**2.358**	2.640
Cap	5.185	4.294	5.914	**1.863**	1.830
Car	6.173	5.022	5.653	**0.897**	0.939
Chair	4.974	4.392	4.479	**1.018**	1.203
Earphone	5.989	5.519	6.212	**3.913**	5.417
Guitar	2.712	3.201	2.360	0.627	**0.615**
Knife	2.828	3.180	2.420	**0.721**	0.831
Lamp	4.858	5.364	4.817	**1.750**	2.294
Laptop	4.612	4.387	3.957	**1.174**	1.202
Motorbike	6.142	4.544	5.173	1.464	**1.393**
Mug	7.268	4.312	6.210	2.764	**2.560**
Pistol	4.508	4.145	3.901	**0.797**	0.984
Rocket	3.514	3.331	3.121	0.465	**0.461**
Skateboard	3.832	3.208	3.657	0.775	**0.774**
Table	5.115	3.653	4.617	**1.310**	1.365
Mean	4.805	4.182	4.334	**1.116**	1.348

Table 4
Point cloud completion results of missing portions of point clouds. The numbers shown are [Pred → GT error/GT → Pred error], scaled by 1000.

Category	CP-Net	PF-Net
Airplane	**1.060**/ **1.312**	1.148/ 1.339
Bag	4.282/ **4.046**	**4.278**/ 4.349
Cap	**4.515**/ **4.479**	6.092/ 5.370
Car	2.743/ 2.250	**2.650**/ **2.111**
Chair	**2.029**/ **2.079**	2.333/ 2.310
Earphone	7.253/ **6.312**	**5.227**/ 9.692
Guitar	**0.437**/ **0.579**	0.493/ 0.638
Knife	**0.493**/ **0.726**	0.505/ 0.919
Lamp	**4.326**/ **3.226**	4.940/ 4.470
Laptop	**1.283**/ **1.321**	1.410/ 1.361
Motorbike	**2.284**/ 2.203	2.347/ **2.017**
Mug	3.532/ **3.871**	**3.120**/ 4.081
Pistol	**1.161**/ **1.131**	1.361/ 1.432
Rocket	**0.864**/ **0.880**	1.221/ 0.899
Skateboard	**1.165**/ 1.917	1.187/ **1.550**
Table	**2.158**/ **2.268**	2.484/ 2.366
Mean	**2.015**/ **2.045**	2.259/ 2.312

baselines failed. In general, our results highlight the advantage of predicting only missing regions compared with the baseline of generating the whole object.

Moreover, the effectiveness of the Mirror Expand module can be intuitively found in the qualitative visualization results. The idea of Mirror Expand is to aggregate the input point cloud information to predict missing regions. With this arrangement, the mutual contact between the incomplete and predicted point clouds naturally has similar detailed structures. Thus, we can intuitively evaluate the prediction performance in the mutual contact parts of PF-Net and CP-Net's results. CP-Net successfully predicts the fine details of the missing regions in the mutual contact parts, especially for guitar, pistol, and table. The point clouds are smooth and neatly distributed in their mutual contact parts.

4.6. Shape Completion on Novel Categories

Quantitative Results. To validate the robustness of the proposed method and verify the further generality of CP-Net, we extended our experiments into novel categories. As mentioned in

Table 5
Point cloud completion results of missing portions of point clouds. The numbers shown are the mean values of EMD, scaled by 100, and F-Score@1%.

Category	CP-Net	PF-Net
Mean	**4.603**/**0.554**	5.201/ 0.540

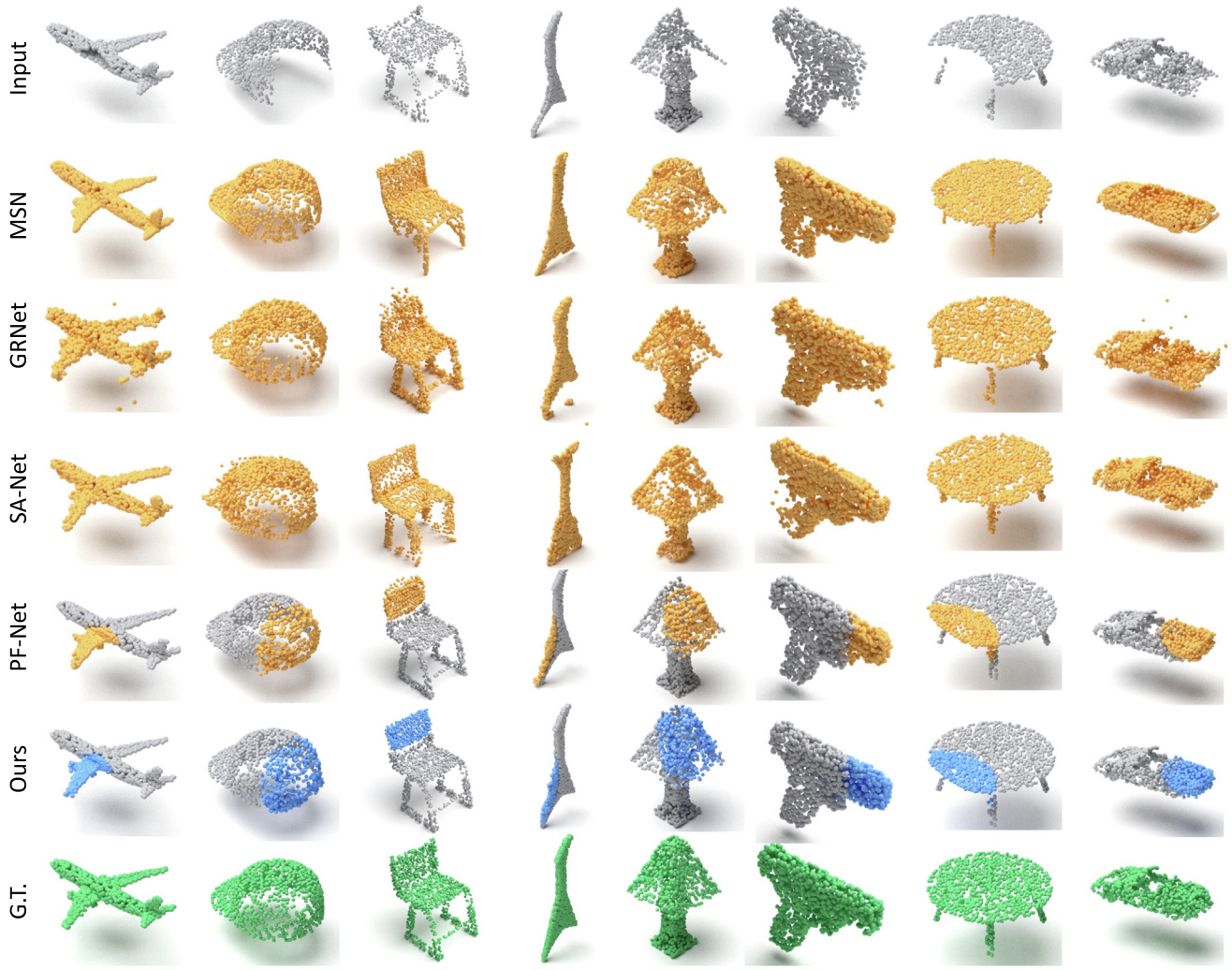

Fig. 5. Comparison of the completion results of other methods and our network on known categories. Here, gray represents the input of partial point clouds, yellow represents the prediction of other methods, and blue and green are CP-Net and the related ground truth, respectively.

Section 4.1, the training and testing categories are independent. Table 6 presents the completion results for the same experiments with known categories.

Combining these experimental results, we can see that although the results for the novel categories are slightly worse than the known categories, our CP-Net still maintains a better result and remains ahead of the baseline methods.

We also present the quantitative results in F-score@1% and EMD in Tables 7 and 8, respectively, similar to the known categories' experiments. These experimental results also strongly support the superiority of our CP-Net.

Similar to the known categories case, the quantitative results for the missing region are shown in Tables 9 and 10. Table 9 presents the completion results with the evaluation metrics for the two parts of the CD. The mean EMD and F-score@1% are shown in Table 10. These experimental results provide a better demonstration because predicting the novel categories is more complicated, and the experimental results of all baseline methods,

Table 6
Point cloud completion results of overall point cloud. The numbers shown are [Pred → GT error/ GT → Pred error], scaled by 1000.

Category	SA-Net	GRNet	MSN	CP-Net	PF-Net
Bag	2.150/ 3.530	1.762/ 2.164	3.516/ 4.585	**0.882**/ 1.179	1.179/ **1.032**
Cap	3.767/ 3.574	2.327/ 3.127	10.065/ 6.901	**2.938**/ **1.123**	3.041/ 1.478
Earphone	6.618/ 4.107	4.378/ 2.456	10.803/ 6.906	**5.450**/ **0.936**	8.100/ 2.416
Guitar	0.674/ 0.963	2.444/ 1.590	0.734/ 0.939	**0.244**/ **0.174**	0.260/ 0.187
Motorbike	1.625/ 4.725	1.558/ 3.078	3.098/ 6.842	**0.869**/ **1.218**	1.103/ 1.736
Pistol	1.133/ 3.252	0.614/ 2.054	1.774/ 5.408	**0.521**/ **1.111**	0.613/ 1.732
Rocket	1.024/ 1.045	2.275/ 0.832	1.100/ 1.330	**0.347**/ **0.383**	0.358/ 0.397
Skateboard	1.033/ 1.707	3.675/ 1.023	1.225/ 2.613	0.471/ **0.545**	**0.444**/ 0.718
Mean	1.360/ 2.261	2.407/ 1.123	2.098/ 3.173	**0.770**/ **0.614**	0.917/ 0.775

Table 7
Point cloud completion results of overall point cloud with F-score@1%.

Category	SA-Net	GRNet	MSN	CP-Net	PF-Net
Bag	0.391	0.766	0.320	**0.822**	0.819
Cap	0.309	0.741	0.203	**0.799**	0.794
Earphone	0.409	0.777	0.298	**0.815**	0.798
Guitar	0.765	0.859	0.805	**0.946**	0.943
Motorbike	0.374	0.581	0.342	**0.837**	0.822
Pistol	0.506	0.676	0.504	0.863	**0.865**
Rocket	0.745	0.717	0.746	**0.917**	0.914
Skateboard	0.676	0.684	0.653	**0.891**	0.890
Mean	0.622	0.661	0.623	**0.897**	0.894

Table 8
Point cloud completion results of overall point cloud with EMD (scaled by 100).

Category	SA-Net	GRNet	MSN	CP-Net	PF-Net
Bag	6.790	4.993	6.856	**1.982**	2.079
Cap	7.495	5.318	9.566	**2.603**	2.913
Earphone	7.601	6.329	9.549	**3.351**	4.320
Guitar	4.271	3.322	3.459	**1.022**	1.059
Motorbike	7.141	5.654	7.126	**1.935**	2.034
Pistol	5.923	3.034	5.441	**1.642**	1.719
Rocket	4.522	3.282	3.824	**1.215**	1.263
Skateboard	4.888	4.127	4.521	**1.420**	1.473
Mean	5.340	4.429	4.940	**1.481**	1.581

Table 9
Point cloud completion results of missing portions of point clouds. The numbers shown are [Pred → GT error/ GT → Pred error], scaled by 1000.

Category	CP-Net	PF-Net
Bag	5.009/ **4.088**	**4.989**/ 4.964
Cap	**11.976**/ **5.170**	14.806/ 8.329
Earphone	**22.500**/ **5.381**	46.369/ 10.800
Guitar	**1.146**/ **1.093**	1.329/ 1.334
Motorbike	**3.887**/ **5.955**	4.117/ 6.210
Pistol	**2.149**/ **6.408**	2.518/7.560
Rocket	**1.407**/ **1.857**	1.512/ 1.957
Skateboard	**1.947**/ **2.523**	2.037/ 3.134
Mean	**3.261**/ **3.176**	4.656/ 4.025

Table 10
Point cloud completion results of missing portions of point clouds. The numbers shown are the mean values of EMD, scaled by 100, and F-score@1%.

Category	CP-Net	PF-Net
Mean	**5.914**/ **0.540**	6.273/ 0.521

including our CP-Net, show a considerable degree of deterioration. However, our network still maintains a certain degree of superior performance, especially in the metrics of EMD, where we obtained sufficiently good results compared with known categories.

Qualitative Results. Fig. 6 shows the point cloud shape completion results obtained with different baseline methods and CP-Net. We can see that predicting unseen objects is a great challenge for each baseline, including our method, compared to the known categories. Many results show significant distortions. However, CP-Net's results still maintain relatively good predictions, especially in maintaining the objects' actual shape. For instance, CP-Net is the only method to predict the missing hole in the bag point cloud successfully. In the guitar point cloud, CP-Net tries its best to reconstruct the general structure without noise. CP-Net has the best overall appearance compared with other baselines. Moreover, the previously mentioned fine detail prediction of missing regions in the mutual contact parts for known categories could also be found in the case of novel categories. These intuitive visualization results further prove the effectiveness of CP-Net.

4.7. Ablation Study

Table 11 lists the results of the ablation studies for the known categories, including all our proposed modules: MEE and PED. We used PF-Net [31] for our baseline model and evaluated the completion results on missing region prediction (512 points). The effectiveness of the proposed module was validated by the fact that better completion results could be achieved by adding the proposed module.

Moreover, the visualization results with different proposed modules are presented in Fig. 7, and the effectiveness of each module can be determined with the corresponding local zoom. In the first column, we find a certain level of noise in the shape of the objects. The distortions also can be captured in the edge of the shape of the objects with the help of the related local zoom. In the second column, the significant distortions have been alleviated, proving the effectiveness of the MEE module, which can extract more useful information from the partial input to help with the upcoming point clouds completion procedures. In the third column, the introduction of the PED module helps to decrease the noise at the edge of the objects, making the details of their shapes appear smoothly and compactly. In the final column, the MEE and PED modules combined effectively predict the fine edge and precise details of the shape of the objects. Even for relatively complicated objects such as lamps and caps, our proposed two modules still maintain robustness compared with those without our modules: the object layout is completed compactly.

4.8. Supplementary Experiments on the ShapeNet Dataset

We experimented with the benchmark ShapeNet [42] dataset to validate the performance of our CP-Net in processing dense point clouds. Its ground truth point clouds contain 16,384 points. To process and generate dense point clouds and keep the merits of our CP-Net, we used the settings $\mathbf{M}_1 = 512, \mathbf{M}_2 = 1024, \mathbf{M} = 4,096$, for which we do not need to change the structure of our CP-Net.

We compared this modified CP-Net against several recent state-of-the-art point cloud completion methods that generate dense

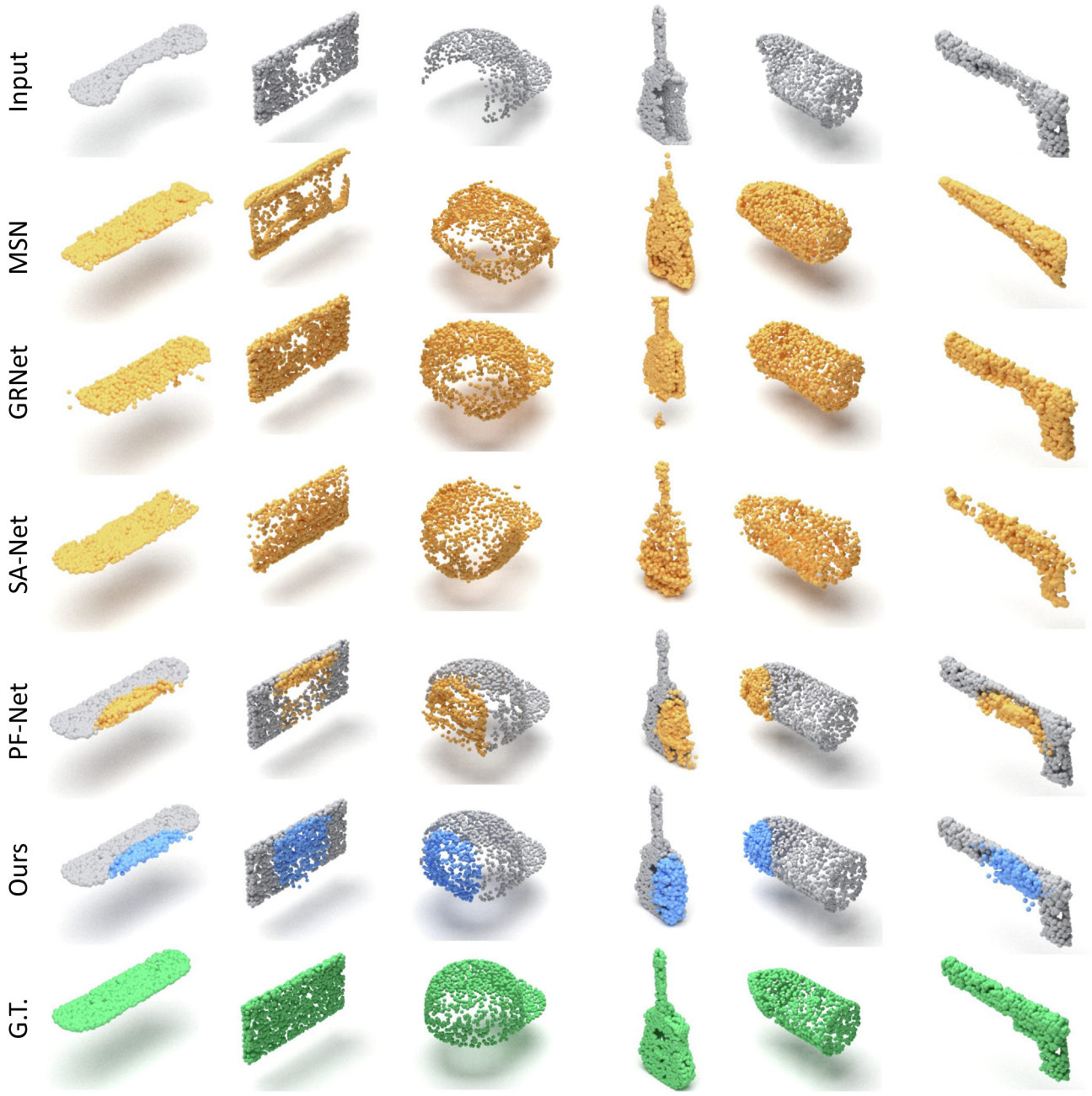

Fig. 6. Comparison of the completion results of other methods and our network on novel categories. Here, gray represents the input of partial point clouds, yellow represents the prediction of other methods, and blue and green are CP-Net and the related ground truth, respectively.

Table 11
Ablation studies (512 points) for the proposed network modules, including Multi-Edge Encoder and Point Expand Decoder.

Multi-Edge Encoder	Point Expand Decoder	CD	EMD	F1
		4.563	5.201	0.548
	✓	4.252	5.074	0.551
✓		4.176	4.908	0.558
✓	✓	**4.084**	**4.603**	**0.574**

point clouds, specifically, MSN [48] and GRNet [44]. For a relatively fair comparison, we trained CP-Net on the same number of input partial point clouds as the other baselines. The evaluation metrics CD and F-Score were used to evaluate the performance of each method. MSN and GRNet aim to predict the whole point clouds with 16,384 points, whereas our CP-Net predicts only the missing point cloud portions of 4,096 points. We present the quantitative results for the whole point clouds and missing portions in Tables 12 and 13. The results indicate that CP-Net outperforms all the other methods in terms of the CD and F-Score@1% of the whole point clouds and provides relatively competitive results for the missing portions. And here, we also provide the visualization results to intuitively validate the performance of our CP-Net in Fig. 8. As shown in Fig. 8, our CP-Net can successfully predict the missing part of point clouds with a relatively good general struc-

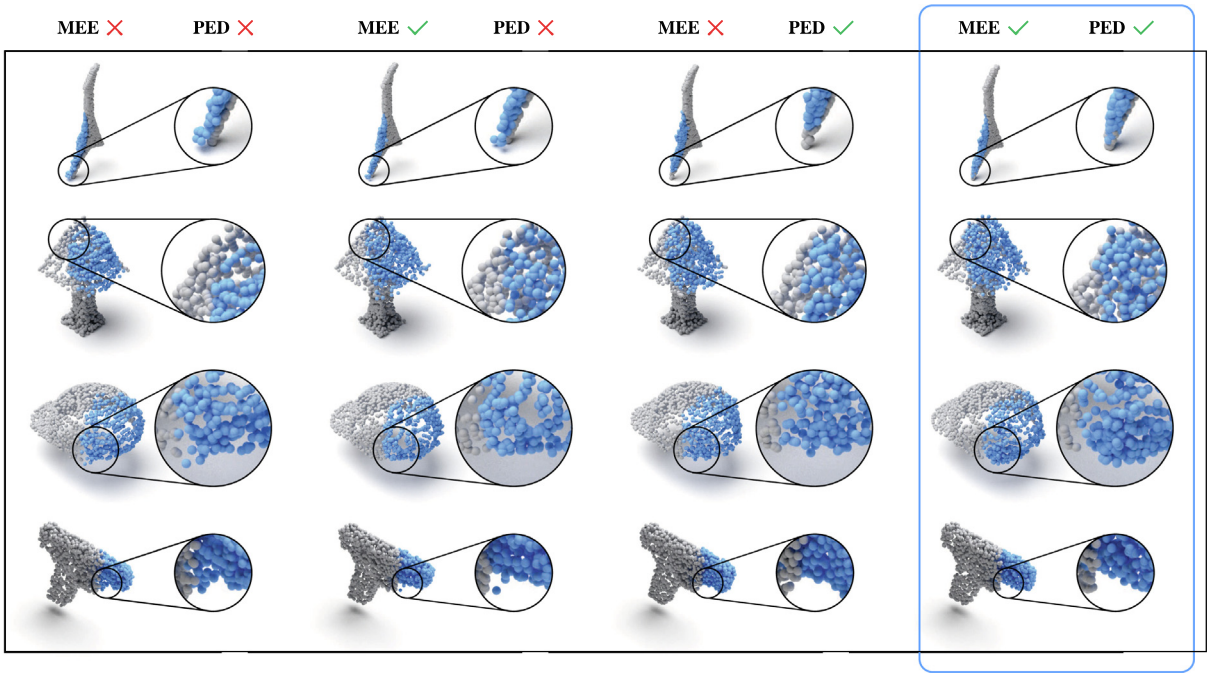

Fig. 7. Visualization of ablation studies: Qualitative results represent each proposed module's contribution. The input partial point clouds are in grey. The network's prediction of the missing part of the point clouds is in blue.

Table 12
Point completion results on ShapeNet compared using CD computed for 16,384 points and 4086 points, multiplied by 10^4. The best results are highlighted in bold.

Methods	Airplane	Cabinet	Car	Chair	Lamp	Sofa	Table	Watercraft	Overall
MSN	1.543	7.249	4.711	4.539	6.479	5.894	3.797	3.853	4.758
GRNet	1.531	3.620	2.752	2.945	2.649	3.613	2.552	2.122	2.723
CP-Net(whole)	**1.482**	**3.312**	**2.478**	**2.031**	**2.476**	**2.897**	**2.091**	**2.101**	**2.153**
CP-Net(missing)	5.089	12.571	9.173	8.014	9.296	11.651	8.478	8.515	8.153

Table 13
Point completion results on ShapeNet compared using F-Score@1%. Note that the F-Score@1% is computed on 16,384 and 4086 points. The best results are highlighted in bold.

Methods	Airplane	Cabinet	Car	Chair	Lamp	Sofa	Table	Watercraft	Overall
MSN	0.885	0.644	0.665	0.657	0.699	0.604	0.782	0.708	0.705
GRNet	0.843	0.618	0.682	0.673	0.761	0.605	0.751	0.750	0.708
CP-Net(whole)	**0.971**	**0.983**	**0.942**	**0.954**	**0.981**	**0.973**	**0.968**	**0.964**	**0.970**
CP-Net(missing)	0.872	0.883	0.854	0.884	0.892	0.874	0.891	0.875	0.879

ture of the objects. However, there is a certain level of noise and distortions in the detail of our prediction. Considering the numerical results in missing portions and the visualization results, we can find out that our CP-Net still has improvement room in processing dense point clouds; the difficulty of predicting dense point clouds needs to be tackled in the future.

4.9. Supplementary Experiments on the Completion3D Dataset

The Completion 3D benchmark [28,20,41] was designed solely for predicting the whole 2,048 points of point clouds. However, to fully demonstrate the power of our proposed method, we predict the missing part of the point clouds with 512 points.

The other methods' results are obtained from the online leaderboard [2]. As shown in Table 14, the overall CD for the proposed CP-Net is 12.32 (multiplied by 10^4), which is close to the results for the other methods, which all aim to predict the whole point clouds. There is a dataset design reason that leads to the deterioration of our results: The input partial point clouds in our ShapeNet-part dataset are generated through the camera angle positions, whereas Completion3D's input is obtained by random subsampling/oversampling of the point clouds [28]. This design in Completion3D has a considerable difference from ours, which means our designed network is unsuitable for the prediction task in the Completion3D.

4.10. Supplementary experiments on KITTI dataset

The KITTI [43] dataset comprises real-world LiDAR scans, where the ground truth is missing for quantitative evaluation. Therefore, we qualitatively evaluated the performance of our CP-Net by the visualization results. Because there are no object models in the KITTI dataset, we pre-trained our model under the car category on the ShapeNet [42] dataset. Here, we provide our visualization results to intuitively validate the performance of our CP-Net in Fig. 9. As shown in Fig. 9's input, the real-world raw data from the KITTI dataset is highly sparse, introducing extra difficulty for

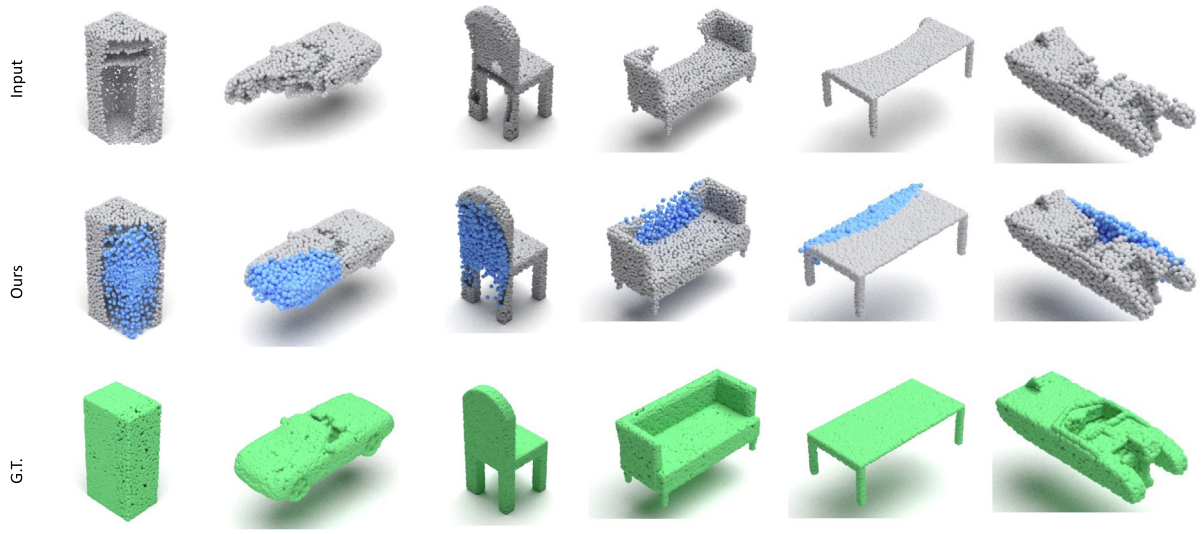

Fig. 8. Completion results of our CP-Net's in ShapeNet dataset. Here, gray represents the input of sampling partial point clouds, blue represents the prediction of CP-Net, and green is the related unsampled ground truth.

Table 14
Point completion results on Completion3D compared using CD. Note that the CD is computed on 2,048 points and multiplied by 10^4. The best results are highlighted in bold.

Methods	Airplane	Cabinet	Car	Chair	Lamp	Sofa	Table	Watercraft	Overall
SA-Net	**5.27**	**14.45**	**7.78**	13.67	13.53	**14.22**	11.75	8.84	11.22
GRNet	6.13	16.90	8.27	**12.23**	**10.22**	14.93	**10.08**	**5.86**	**10.64**
CP-Net	6.16	14.94	8.81	14.02	15.67	15.09	12.12	9.31	12.32

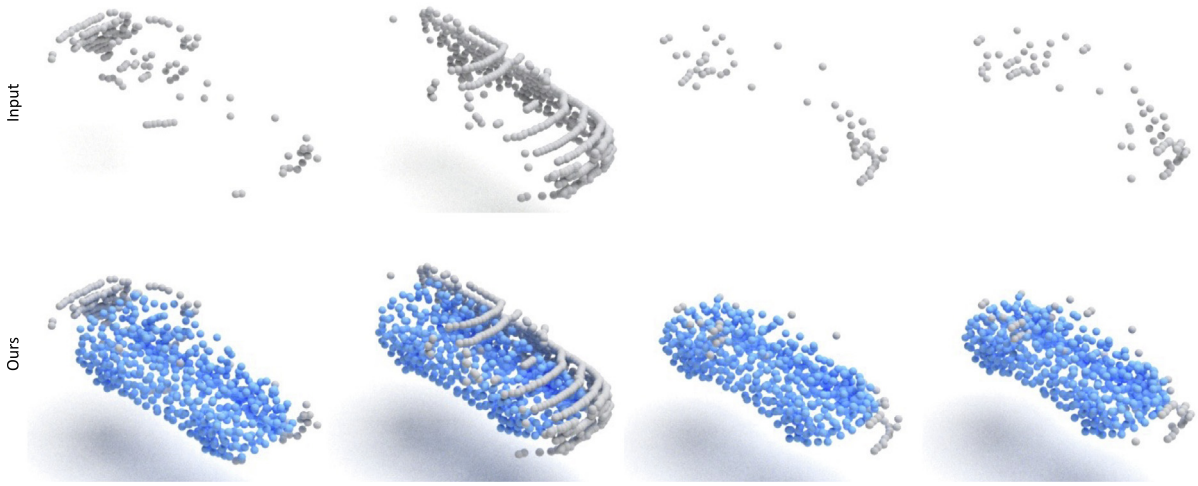

Fig. 9. Visualization of ablation KITTI dataset. Qualitative results representing the performance of our CP-Net in real-world raw data. The input partial point clouds are in grey. Our prediction of the missing part of point clouds is in blue.

the point clouds completion task. Nevertheless, our CP-Net tries its best to predict the shape of missing objects and their related details. However, we must admit that the result is not good enough, and we still have room to improve.

5. Conclusion

In this paper, we proposed CP-Net, a novel network structure for the end-to-end point cloud completion task that takes partial point clouds as the input to predict their missing part. Our proposed model can generate point clouds smoothly and effectively with local information and the whole structure of target objects. Our model outperformed the existing baseline methods on the benchmark ShapeNet-Part. Furthermore, experiments proved the effectiveness of our proposed modules. The Mirror Expand module expands the feature information with more meaningful content and less duplicated information. The proposed network structure captures latent features densely, which is key to the next step of point cloud generation. Our expansion approach is to find symmetrical neighborhood points to generate pair points. However, we obtained relatively weak results on other benchmarks; therefore, we must improve our method further. For instance, there are also many other potential ways of expansion. In the future, we will exploit the possibility of other means of expansion, such as

expanding the point clouds in high-dimensional feature space or introducing a learnable rotation angle to a new point. In this way, we can further broaden the application of our network in 3D vision.

CRediT authorship contribution statement

Fangzhou Lin: Methodology, Data curation, Writing - original draft. **Yajun Xu:** Methodology. **Ziming Zhang:** Conceptualization, Project administration. **Chenyang Gao:** Data curation. **Kazunori D Yamada:** Resources, Investigation, Supervision.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

[1] C. Boehnen, P. Flynn, Accuracy of 3d scanning technologies in a face scanning scenario, in: Fifth International Conference on 3-D Digital Imaging and Modeling (3DIM'05), 2005, pp. 310–317. doi:10.1109/3DIM.2005.13.

[2] W. Gao, S. Kim, H. Bosse, H. Haitjema, Y. Chen, X. Lu, W. Knapp, A. Weckenmann, W. Estler, H. Kunzmann, Measurement technologies for precision positioning, CIRP Annals 64 (2) (2015) 773–796, https://doi.org/10.1016/j.cirp.2015.05.009, URL:https://www.sciencedirect.com/science/article/pii/S0007850615001481.

[3] E.E. Hitomi, J.V. Silva, G.C. Ruppert, 3d scanning using rgbd imaging devices: A survey, in: Developments in Medical Image Processing and Computational Vision, Springer, 2015, pp. 379–395.

[4] Y. Xu, S. Arai, D. Liu, F. Lin, K. Kosuge, Fpcc-net: Fast point cloud clustering for instance segmentation, arXiv preprint arXiv:2012.14618.

[5] S. Kato, S. Tokunaga, Y. Maruyama, S. Maeda, M. Hirabayashi, Y. Kitsukawa, A. Monrroy, T. Ando, Y. Fujii, T. Azumi, Autoware on board: Enabling autonomous vehicles with embedded systems, in: 2018 ACM/IEEE 9th International Conference on Cyber-Physical Systems (ICCPS), 2018, pp. 287–296, https://doi.org/10.1109/ICCPS.2018.00035.

[6] A. Alliegro, D. Valsesia, G. Fracastoro, E. Magli, T. Tommasi, Denoise and contrast for category agnostic shape completion, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2021, pp. 4629–4638.

[7] H. Wu, Y. Miao, LRA-Net: local region attention network for 3D point cloud completion, in: W. Osten, D.P. Nikolaev, J. Zhou (Eds.), Thirteenth International Conference on Machine Vision, Vol. 11605, International Society for Optics and Photonics, SPIE, 2021, pp. 357–367, https://doi.org/10.1117/12.2586422, URL: https://doi.org/10.1117/12.2586422.

[8] H. Wu, Y. Miao, R. Fu, Point cloud completion using multiscale feature fusion and cross-regional attention, Image and Vision Computing 111 (2021), https://doi.org/10.1016/j.imavis.2021.104193, URL:https://www.sciencedirect.com/science/article/pii/S0262885621000986 104193.

[9] A. Dai, C.R. Qi, M. Nie?ner, Shape completion using 3d-encoder-predictor cnns and shape synthesis, in: 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017, pp. 6545–6554. doi:10.1109/CVPR.2017.693.

[10] T. Hu, Z. Han, M. Zwicker, 3d shape completion with multi-view consistent inference, in: Proceedings of the AAAI Conference on Artificial Intelligence 34 (07), 2020, pp. 10997–11004, https://doi.org/10.1609/aaai.v34i07.6734, URL: https://ojs.aaai.org/index.php/AAAI/article/view/6734.

[11] Y. Yu, Z. Huang, F. Li, H. Zhang, X. Le, Point encoder gan: A deep learning model for 3d point cloud inpainting, Neurocomputing 384 (2020) 192–199, https://doi.org/10.1016/j.neucom.2019.12.032, URL:https://www.sciencedirect.com/science/article/pii/S0925231219317357.

[12] A. Sharf, M. Alexa, D. Cohen-Or, Context-based surface completion, in: ACM SIGGRAPH 2004 Papers, SIGGRAPH '04, Association for Computing Machinery, New York, NY, USA, 2004, p. 878?887. doi:10.1145/1186562.1015814. URL: https://doi.org/10.1145/1186562.1015814.

[13] G. Harary, A. Tal, E. Grinspun, Context-based coherent surface completion, ACM Trans. Graph. 33 (1). doi:10.1145/2532548. URL:https://doi.org/10.1145/2532548.

[14] A. Nealen, T. Igarashi, O. Sorkine, M. Alexa, Laplacian mesh optimization, in: Proceedings of the 4th International Conference on Computer Graphics and Interactive Techniques in Australasia and Southeast Asia, GRAPHITE '06, Association for Computing Machinery, New York, NY, USA, 2006, p. 381?389. doi:10.1145/1174429.1174494. URL:https://doi.org/10.1145/1174429.1174494.

[15] O. Sorkine, D. Cohen-Or, Least-squares meshes, Proceedings Shape Modeling Applications 2004 (2004) 191–199, https://doi.org/10.1109/SMI.2004.1314506.

[16] H. Su, S. Maji, E. Kalogerakis, E. Learned-Miller, Multi-view convolutional neural networks for 3d shape recognition, in: 2015 IEEE International Conference on Computer Vision (ICCV), 2015, pp. 945–953, https://doi.org/10.1109/ICCV.2015.114.

[17] D. Maturana, S. Scherer, Voxnet: A 3d convolutional neural network for real-time object recognition, in: 2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2015, pp. 922–928, https://doi.org/10.1109/IROS.2015.7353481.

[18] J. Varley, C. DeChant, A. Richardson, J. Ruales, P. Allen, Shape completion enabled robotic grasping, in: 2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2017, pp. 2442–2447. doi:10.1109/IROS.2017.8206060.

[19] Y. Guo, Y. Liu, A. Oerlemans, S. Lao, S. Wu, M.S. Lew, Deep learning for visual understanding: A review, Neurocomputing 187 (2016) 27–48, recent Developments on Deep Big Vision. doi: 10.1016/j.neucom.2015.09.116. URL: https://www.sciencedirect.com/science/article/pii/S0925231215017634.

[20] W. Yuan, T. Khot, D. Held, C. Mertz, M. Hebert, Pcn: Point completion network, in: 2018 International Conference on 3D Vision (3DV), 2018, pp. 728–737, https://doi.org/10.1109/3DV.2018.00088.

[21] Z. Xie, J. Chen, B. Peng, Point clouds learning with attention-based graph convolution networks, Neurocomputing 402 (2020) 245–255, https://doi.org/10.1016/j.neucom.2020.03.086, URL:https://www.sciencedirect.com/science/article/pii/S0925231220304732.

[22] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, Y. Bengio, Generative adversarial nets, in: Z. Ghahramani, M. Welling, C. Cortes, N. Lawrence, K.Q. Weinberger (Eds.), Advances in Neural Information Processing Systems, Vol. 27, Curran Associates Inc, 2014. URL: https://proceedings.neurips.cc/paper/2014/file/5ca3e9b122f61f8f06494c97b1afccf3-Paper.pdf.

[23] R.A. Yeh, C. Chen, T.Y. Lim, A.G. Schwing, M. Hasegawa-Johnson, M.N. Do, Semantic image inpainting with deep generative models, in: 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017, pp. 6882–6890. doi:10.1109/CVPR.2017.728.

[24] J. Yu, Z. Lin, J. Yang, X. Shen, X. Lu, T.S. Huang, Generative image inpainting with contextual attention, in: 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2018, pp. 5505–5514. doi:10.1109/CVPR.2018.00577.

[25] O. Elharrouss, N. Almaadeed, S. Al-Maadeed, Y. Akbari, Image inpainting: A review, Neural Processing Letters (2019) 1–22.

[26] C. Lu, G. Dubbelman, Learning to complete partial observations from unpaired prior knowledge, Pattern Recognition 107 (2020), https://doi.org/10.1016/j.patcog.2020.107426, URL:https://www.sciencedirect.com/science/article/pii/S0031320320302296 107426.

[27] C.R. Qi, H. Su, K. Mo, L.J. Guibas, Pointnet: Deep learning on point sets for 3d classification and segmentation, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2017, pp. 652–660.

[28] L.P. Tchapmi, V. Kosaraju, H. Rezatofighi, I. Reid, S. Savarese, Topnet: Structural point cloud decoder, in: 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2019, pp. 383–392. doi:10.1109/CVPR.2019.00047.

[29] X. Wang, M.H. Ang, G.H. Lee, Cascaded refinement network for point cloud completion, in: 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2020, pp. 787–796. doi:10.1109/CVPR42600.2020.00087.

[30] Y. Chang, C. Jung, Y. Xu, Finerpcn: High fidelity point cloud completion network using pointwise convolution, Neurocomputing doi: 10.1016/j.neucom.2021.06.080. URL:https://www.sciencedirect.com/science/article/pii/S0925231221010109.

[31] Z. Huang, Y. Yu, J. Xu, F. Ni, X. Le, Pf-net: Point fractal network for 3d point cloud completion, in: 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2020, pp. 7659–7667. doi:10.1109/CVPR42600.2020.00768.

[32] R.Q. Charles, H. Su, M. Kaichun, L.J. Guibas, Pointnet: Deep learning on point sets for 3d classification and segmentation, in: 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017, pp. 77–85. doi:10.1109/CVPR.2017.16.

[33] L. Yi, V.G. Kim, D. Ceylan, I.-C. Shen, M. Yan, H. Su, C. Lu, Q. Huang, A. Sheffer, L. Guibas, A scalable active framework for region annotation in 3d shape collections, ACM Trans. Graph. 35 (6). doi:10.1145/2980179.2980238. URL: https://doi.org/10.1145/2980179.2980238.

[34] W. Zhang, S. Su, B. Wang, Q. Hong, L. Sun, Local k-nns pattern in omni-direction graph convolution neural network for 3d point clouds, Neurocomputing 413 (2020) 487–498, https://doi.org/10.1016/j.neucom.2020.06.095, URL:https://www.sciencedirect.com/science/article/pii/S0925231220310845.

[35] C.R. Qi, L. Yi, H. Su, L.J. Guibas, Pointnet++: Deep hierarchical feature learning on point sets in a metric space, in: Proceedings of the 31st International Conference on Neural Information Processing Systems, Curran Associates Inc., Red Hook, NY, USA, 2017, pp. 5105–5114.

[36] Y. Wang, Y. Sun, Z. Liu, S.E. Sarma, M.M. Bronstein, J.M. Solomon, Dynamic graph cnn for learning on point clouds, ACM Trans. Graph. 38 (5). doi:10.1145/3326362. URL:https://doi.org/10.1145/3326362.

[37] K. Zhang, M. Hao, J. Wang, C.W. de Silva, C. Fu, Linked dynamic graph cnn: Learning on point cloud via linking hierarchical features, arXiv preprint arXiv:1904.10014.

[38] Y. Yang, C. Feng, Y. Shen, D. Tian, Foldingnet: Point cloud auto-encoder via deep grid deformation, in: 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2018, pp. 206–215. doi:10.1109/CVPR.2018.00029.

[39] X. Wen, T. Li, Z. Han, Y.-S. Liu, Point cloud completion by skip-attention network with hierarchical folding, in: 2020 IEEE/CVF Conference on Computer

Vision and Pattern Recognition (CVPR), 2020, pp. 1936–1945. doi:10.1109/CVPR42600.2020.00201.
[40] Z. Huang, Y. Yu, J. Xu, F. Ni, X. Le, Pf-net: Point fractal network for 3d point cloud completion, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2020, pp. 7662–7670.
[41] A.X. Chang, T. Funkhouser, L. Guibas, P. Hanrahan, Q. Huang, Z. Li, S. Savarese, M. Savva, S. Song, H. Su, J. Xiao, L. Yi, F. Yu, ShapeNet: An Information-Rich 3D Model Repository, Tech. Rep. arXiv:1512.03012 [cs.GR], Stanford University — Princeton University — Toyota Technological Institute at Chicago (2015).
[42] Z. Wu, S. Song, A. Khosla, F. Yu, L. Zhang, X. Tang, J. Xiao, 3d shapenets: A deep representation for volumetric shapes, in: 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2015, pp. 1912–1920. doi:10.1109/CVPR.2015.7298801.
[43] A. Geiger, P. Lenz, C. Stiller, R. Urtasun, Vision meets robotics: The kitti dataset, The International Journal of Robotics Research 32 (11) (2013) 1231–1237, https://doi.org/10.1177/0278364913491297, arXiv:https://doi.org/10.1177/0278364913491297, URL:https://doi.org/10.1177/0278364913491297.
[44] H. Xie, H. Yao, S. Zhou, J. Mao, S. Zhang, W. Sun, Grnet: Gridding residual network for dense point cloud completion, in: A. Vedaldi, H. Bischof, T. Brox, J.-M. Frahm (Eds.), Computer Vision – ECCV 2020, Springer International Publishing, Cham, 2020, pp. 365–381.
[45] A. Knapitsch, J. Park, Q.-Y. Zhou, V. Koltun, Tanks and temples: Benchmarking large-scale scene reconstruction, ACM Trans. Graph. 36 (4). doi:10.1145/3072959.3073599. URL:https://doi.org/10.1145/3072959.3073599.
[46] M. Tatarchenko, S.R. Richter, R. Ranftl, Z. Li, V. Koltun, T. Brox, What do single-view 3d reconstruction networks learn?, in: 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2019, pp. 3400–3409. doi:10.1109/CVPR.2019.00352.
[47] Y. Rubner, C. Tomasi, L.J. Guibas, The earth mover's distance as a metric for image retrieval, Int. J. Comput. Vis. 40 (2) (2000) 99–121, https://doi.org/10.1023/A:1026543900054, URL:https://doi.org/10.1023/A:1026543900054.
[48] M. Liu, L. Sheng, S. Yang, J. Shao, S.-M. Hu, Morphing and sampling network for dense point cloud completion, Proceedings of the AAAI Conference on Artificial Intelligence 34 (07) (2020) 11596–11603. doi:10.1609/aaai.v34i07.6827. URL: https://ojs.aaai.org/index.php/AAAI/article/view/6827
[49] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Kopf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, S. Chintala, Pytorch: An imperative style, high-performance deep learning library, in: H. Wallach, H. Larochelle, A. Beygelzimer, F. d' Alché-Buc, E. Fox, R. Garnett (Eds.), Advances in Neural Information Processing Systems, Vol. 32, Curran Associates Inc, 2019. URL:https://proceedings.neurips.cc/paper/2019/file/bdbca288fee7f92f2bfa9f7012727740-Paper.pdf.
[50] J. Sanders, E. Kandrot, CUDA by example: an introduction to general-purpose GPU programming, Addison-Wesley Professional, 2010.
[51] D.P. Kingma, J. Ba, Adam: A method for stochastic optimization, in: ICLR (Poster), 2015. URL:http://arxiv.org/abs/1412.6980

Fangzhou Lin received an M.S. degree in information sciences from Tohoku University, Japan, Sendai, in 2021. He is pursuing a PhD from Tohoku University, Sendai, Japan. His research interests include deep learning, satellite image processing, visual question answering, recurrent neural network architecture, and point cloud completion.

Yajun Xu received a B.S. in mechanical engineering from Central South University, Changsha, China, in 2015. and the M.E. from the graduate school of Engineering, Tohoku University, Japan, in 2020. He is pursuing a PhD from Hokkaido University, Sapporo, Japan. His research interests include deep learning, 2D/3D segmentation, and 6D pose estimation.

Ziming Zhang is an assistant professor at Worcester Polytechnic Institute (WPI). Before joining WPI, he was a research scientist at Mitsubishi Electric Research Laboratories (MERL) from 2017–2019. Before that, he was a research assistant professor at Boston University from 2016–2017. Dr. Zhang received his PhD in 2013 from Oxford Brookes University, UK, under the supervision of Prof. Philip H. S. Torr. His research areas include object recognition and detection, zero-shot learning, deep learning, optimization, large-scale information retrieval, visual surveillance, and medical imaging analysis. His works have appeared in TPAMI, CVPR, ICCV, ECCV, ACM Multimedia and NIPS. He won the R&D 100 Award 2018.

Chenyang Gao received a B.S. in computer science from the Department of computer science and engineering, Southern University of Science and Technology, China, in 2021. He is pursuing a master's degree with the School of Electronic Information and Communications, Huazhong University of Science and Technology (HUST), China. His main research interests include text-based person search and cross-modal retrieval.

Kazunori Yamada is a Professor at Tohoku University, Sendai, Japan. His research interest is neural network algorithms for processing context information, machine consciousness.