CSDN: Cross-modal Shape-transfer Dual-refinement Network for Point Cloud Completion

Zhe Zhu, Liangliang Nan, Haoran Xie, *Senior Member, IEEE*, Honghua Chen, Jun Wang, Mingqiang Wei, *Senior Member, IEEE*, and Jing Qin

Abstract—How will you repair a physical object with some missings? You may imagine its original shape from previously captured images, recover its overall (global) but coarse shape first, and then refine its local details. We are motivated to imitate the physical repair procedure to address point cloud completion. To this end, we propose a cross-modal shape-transfer dual-refinement network (termed CSDN), a coarse-to-fine paradigm with images of full-cycle participation, for quality point cloud completion. CSDN mainly consists of "shape fusion" and "dual-refinement" modules to tackle the cross-modal challenge. The first module transfers the intrinsic shape characteristics from single images to guide the geometry generation of the missing regions of point clouds, in which we propose IPAdaIN to embed the global features of both the image and the partial point cloud into completion. The second module refines the coarse output by adjusting the positions of the generated points, where the local refinement unit exploits the geometric relation between the novel and the input points by graph convolution, and the global constraint unit utilizes the input image to fine-tune the generated offset. Different from most existing approaches, CSDN not only explores the complementary information from images but also effectively exploits cross-modal data in the *whole* coarse-to-fine completion procedure. Experimental results indicate that CSDN performs favorably against twelve competitors on the cross-modal benchmark.

Index Terms—CSDN, point cloud completion, cross modality, multi-feature fusion

✦

1 INTRODUCTION

As a commonly used representation, point clouds express the geometry of objects in 3D space with a simple and flexible data structure [1]. The raw point clouds, which are often captured by scanning devices (e.g., laser scanners), unavoidably have missing regions. Some of main reasons include the inter-object occlusion, self-occlusion, surface reflectivity, and limited scanning range [2]. It is possible to complete a partial point cloud by carefully positioning 3D scanners to carry out multi-pass scanning. However, such remedial measures are time-consuming and often hopeless due to the limited scene accessibility. Also, it is promising to learn a completion mapping with single-modal point clouds. Two challenges of using the single-modal point clouds will be encountered. First, the low-scanning resolution of 3D sensors (e.g., Kinect) often leads to the sparseness of captured data. Determining whether an arbitrary missing region originates from incompleteness or intrinsic spareness is very difficult in such sparse point clouds. Second, the inference uncertainty of missing regions will happen due to the limited geometry cues available [3].

Cross-sensors (e.g., depth cameras and camera-LiDAR scanners) are now accessible and affordable to continuously and simultaneously capture color images and point clouds of 3D
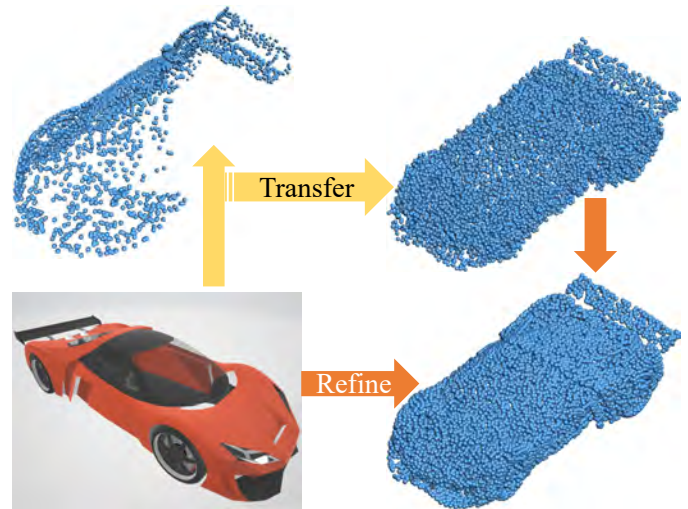

Fig. 1. Image is an important modality to provide complementary information (both structure and detail information) for 3D (sparse) data. Inspired by how a man/woman repairs physical objects, we consider that the image should participate in the whole procedure of coarse-to-fine repair. Differently from existing efforts in this topic, CSDN is a new point cloud completion paradigm that exploits cross modalities in the *whole* coarse-to-fine completion procedure (Yellow arrows: the shape fusion stage; Orange arrows: the dual-refinement stage).

scenes. Scene images have two main merits: (1) compared to 3D data acquisition using static/mobile laser scanners, images of the scene or in particular the concerned missing regions are easier to acquire [4]; (2) images have high resolutions and rich textures, potentially facilitating various geometric processing tasks by pro-

---

Z. Zhu, H. Chen, J. Wang and M. Wei are with the School of Computer Science and Technology, Nanjing University of Aeronautics and Astronautics, Nanjing, China (e-mail: zhuzhe0619@nuaa.edu.cn; chenhonghuacn@gmail.com; wjun@nuaa.edu.cn; mingqiang.wei@gmail.com).
L. Nan is with the Urban Data Science Section, Delft University of Technology, Delft, Netherlands (e-mail: liangliang.nan@tudelft.nl).
H. Xie is with the Department of Computing and Decision Sciences, Lingnan University, Hong Kong, China (e-mail: hrxie2@gmail.com).
J. Qin is with the School of Nursing, The Hong Kong Polytechnic University, Hong Kong, China (e-mail: harry.qin@polyu.edu.hk).

viding complementary information to point clouds [5]. Although recent years have witnessed considerable efforts of using images to help understand point clouds, it is still challenging to fuse data from the cross sensors, making them 'really' complementary for quality geometric processing tasks.

To address the cross-modal challenge in point cloud completion, one can absorb the wisdom of physical object repair. Imagine how a professional restorer repairs a physical object with some missings. She/he will perceive the object's original shape from previously captured images and first recover its global yet the coarse shape and then refine its local details. We attempt to imitate the physical repair procedure to design a novel cross-modal point cloud completion paradigm. To this end, we propose a cross-modal shape-transfer dual-refinement network (termed CSDN), which exploits the complementary information of images and input partial geometry for quality point cloud completion.

CSDN is designed to fuse the features of both the image and point cloud based on their role in conveying a complete shape. For effective learning of both global and local features at different completion stages, we exploit different strategies to tackle the cross-modal challenge. To fully integrate image information for shape completion in a coarse-to-fine manner, CSDN consists of a "shape fusion" module to generate a coarse yet complete shape and a "dual-refinement" module to obtain the completion result.

First, rather than generating the complete model with only single modality data, we develop a novel solution called IPAdaIN (a variant of Adaptive Instance Normalization [6], [7]) to embed the global features of both the images and the partial point cloud. To fuse the information from the two modalities, IPAdaIN adaptively transfers the "shape style" (i.e., the global shape information) obtained from the image to the point cloud domain by feature normalization, which facilitates obtaining an initial complete 3D shape with the aid of images.

Then, the dual-refinement module takes the generated point cloud as input and further refines the shape details. Specifically, we design this module with a pair of refinement units, called local refinement and global constraint, respectively. To recover geometric details, we apply graph convolution in the *local refinement* unit. The *global constraint* unit leverages image features to constrain the offset generated by *local refinement*. This dual-path refinement strategy enables CSDN to precisely capture both global shape structures and local details.

CSDN explores the complementary structure information from images, and it synergizes both point and image features in the whole coarse-to-fine completion procedure, as shown in Figure 1. This makes CSDN distinct from the recently proposed ViPC [3]. ViPC also exploits an additional single-view image to provide the global structural priors in its coarse-completion stage. However, it cannot recover high-frequency details and local topology for complex shapes in the refinement stage since the features learned from different modalities are simply concatenated. As a result, the completed shapes from ViPC are often noisy and lack finer geometric details. In contrast, our CSDN enables the recovery of finer details and topology by disentangling features for *local refinement* and features for *global constraint*.

Extensive experiments clearly show the superiority of CSDN over its ten competitors. For example, CSDN reduces the Mean Chamfer Distance by 0.281 compared to PoinTr [8], and it outperforms ViPC [3] by a large margin of 22.3% on the benchmark dataset ShapeNet-ViPC [3]. The main contributions can be summarized as follows.

- By imitating the physical repair procedure, we propose a novel cross-modal shape-transfer dual-refinement network for point cloud completion. Compared with previous cross-modal methods, CSDN applies the disentangled feature fusion strategy, which significantly improves the completion performance.
- We propose a shape fusion module for generating a coarse yet complete shape, in which a novel solution called IPAdaIN is designed to adaptively transfer the global shape information obtained from the image to the point cloud domain by feature normalization.
- We propose a dual-refinement module, enabling CSDN to capture both global shape structures and local details.

2 RELATED WORK

This section reviews recent advances in learning-based point cloud analysis, point cloud generation, single-view reconstruction, and 3D shape completion, followed by cross-modal feature fusion.

2.1 Point Cloud Analysis

Most of recent neural networks for point cloud analysis are designed to process unordered point clouds directly, rather than converting them into 2D grids or 3D voxels in advance. As the pioneer, PointNet [9] proposes to directly process points through MLPs and uses a symmetric function to aggregate features. Then, PointNet++ [10] utilizes a hierarchical architecture to capture local information. To generalize typical CNN to point clouds, PointCNN [11] learns a transformation matrix for local points permutation. PointConv [12] and KPConv [13] build convolution weights with point coordinates. By expressing a point cloud using a graph structure, DGCNN [14] proposes EdgeConv for efficient local geometry learning. Several follow-up works [15]–[18] further demonstrate the efficiency of graph representations for local point clouds. More recently, Point Transformer [19] and PCT [20] extend the Transformer architecture to analyze point clouds. These works greatly facilitate the development of point cloud processing in downstream tasks.

2.2 Point Cloud Generation

Point cloud generation aims to learn generative models on 3D shapes. rGAN [21] first proposes to use GAN to generate point clouds, followed by some variants [22]–[24] that aim at generating higher quality point clouds by exploiting local structure information. As a representative flow-based method, PointFlow [25] treats shapes and shape points as a two-level hierarchy of distributions. To better characterize the data distribution, ShapeGF [26] generates points by learning the stochastic gradient of its log density field. Generative methods are also widely adopted in other 3D vision tasks. For example, GSPN [27] generates proposals with an analysis-by-synthesis strategy. More recently, PointGrow [28] and AutoSDF [29] use auto-regressive models to generate 3D shapes. PointGrow learns point distribution in a point-by-point manner. AutoSDF employs Truncated-Signed Distance Field (T-SDF) as the 3D representation and utilizes the transformer architecture to learn non-sequential prior over 3D shapes. Note that the goal of most of the above methods is to generate high-fidelity 3D shapes by learning and mimicking existing data distributions. Different from them, Point cloud completion methods [3], [8], [30] and point cloud upsampling methods [31]–[33] are conditioned by the
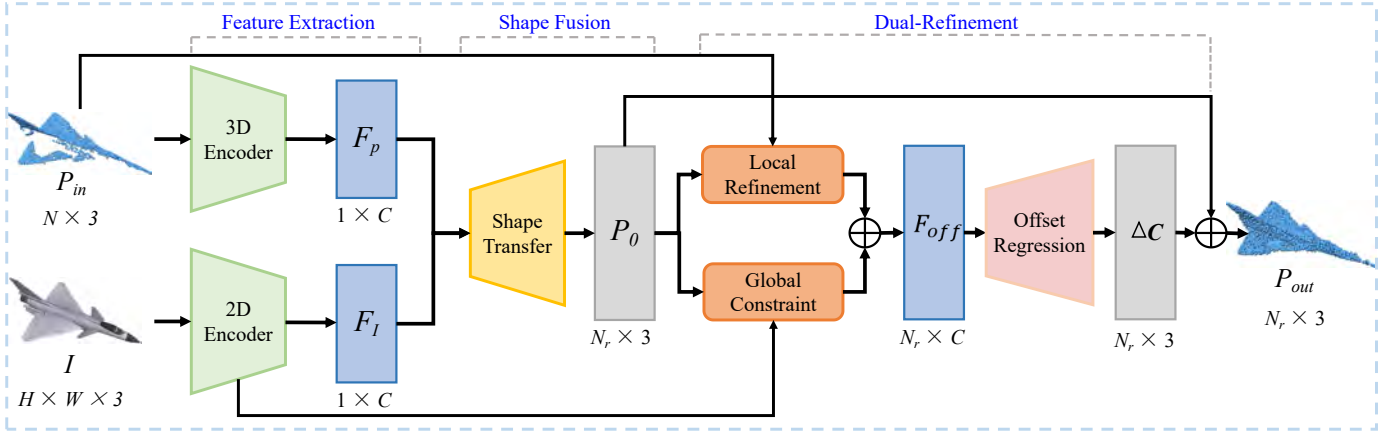
Fig. 2. The architecture of Cross-modal Shape-transfer Dual-refinement Network (CSDN). CSDN is designed as a coarse-to-fine structure, and complementary images participate in the whole process. CSDN consists of three modules: feature extraction, shape fusion, and dual refinement. Different from the existing wisdom, cross-modal data are involved in the whole coarse-to-fine completion procedure of CSDN, ensuring the completion result with both rich structures and details.

input partial/sparse point clouds. These efforts focus on mapping and completing input partial shapes to their complete versions.

2.3 Shape Completion

Traditional methods are often geometry- or template-based. The geometry-based methods either fill small holes in 3D models by local interpolations [34]–[36], or infer the complete shapes by exploiting structure regularities, such as symmetry and repetitive patterns [37]–[40]. The template-based methods [41]–[44] complete the geometry by matching the partial shape with template models from a database. However, for this conventional wisdom of shape completion, users have to tweak parameters multiple times to obtain satisfied results in practical scenarios. This inconvenience heavily discounts the efficiency and user experience.

Early learning-based methods [45]–[47] utilize voxel-based representations for 3D convolutional neural completion networks, but suffering from expensive computation cost. Later, the PointNet-based encoder-decoder architecture [9] inspires many new works. For example, the pioneering one, PCN [30], generates points in a coarse-to-fine manner and folds the 2D grids to reconstruct a complex complete shape. This coarse-to-fine completion framework has also been employed by [47], [48]. TopNet [49] introduces a hierarchical rooted tree structure decoder to generate points of different levels of details. The following work MSN [50] tackles the completion problem by generating different parts of shape with a residual sub-network to refine the coarse points. PF-Net [51] extracts multi-scale features to encode both local and global information and then fuses them to reconstruct complete point clouds. Besides, the adversarial training strategy is used to further enhance the point quality [51]. Also, Wang et al. [52] propose to synthesize the dense and complete object shapes in a cascaded refinement manner, and jointly optimize the reconstruction loss and an adversarial loss. RL-GAN-Net [53] first tries to use an RL agent to drive a generative adversarial network to predict a complete point cloud. PMP-Net [54] formulates shape prediction as a point cloud deformation problem, and generates complete point clouds by moving input points to appropriate positions iteratively with a minimum moving distance. Researchers have also introduced Transformer [55] for point cloud completion, like [8], [56]. PoinTr [8] reformulates point cloud completion as a

set-to-set translation problem, so that a transformer-based encoder-decoder architecture is naturally adopted. SnowflakeNet [56] utilizes transformer and point-wise feature deconvolutional modules to refine the initial point cloud iteratively.

Apart from the above supervised approaches, there are also a few studies concerning unpaired point cloud completion. AML [57] introduces a weakly-supervised approach by learning the maximum likelihood. Pcl2Pcl [58] proposes a GAN framework to bridge the semantic gap between incomplete and complete shapes. However, these methods only consider the single-side correspondence from the incomplete shapes to their complete shapes. In contrast, Cycle4Completion [59] enhances the completion performance by establishing the two-way geometric correspondence between complete shapes and incomplete shapes. Although achieving remarkable performance on point cloud completion, recent efforts generally use the mono-modality input. It is difficult to infer an accurate mapping from an incomplete point cloud with a large-scale incompleteness to its complete point cloud.

2.4 Single-view Reconstruction

Single-view reconstruction methods can be divided into point-based, voxel-based, mesh-based, and implicit field-based categories according to the representation of 3D data. Point-based methods [60]–[62] infer the point coordinates directly from images. PSGN [60] can reconstruct better shapes by connecting a 2D encoder and a 3D decoder. GAL [62] proposes a geometric adversarial loss to constrain the predicted point cloud. Voxel-based methods [63]–[65] solve the reconstruction problem by using 2D and 3D CNNs. As for mesh-based methods, the pioneering work Pixel2Mesh [66] and Pixel2Mesh++ [67] fuse 2D features into mesh deformation. The follow-up methods [68], [69] further extend this framework by improving deformation with topology modification and self-supervised learning. However, due to the intersection of meshes, the connection pattern hinders the generation of details. Recently, the community has focused heavily on deep implicit representations [29], [70]–[74], where 3D shapes are represented as implicit functions implemented by neural networks with possible image inputs. This representation has also been used to support other tasks, such as shape completion [29], pairwise shape mating [75], and point cloud upsampling [76]. Instead

of directly inferring 3D shapes from single-view images, our approach assists the completion by combining different modalities at the feature level, where image features are used for both coarse completion and detailed refinement.

2.5 Cross-modal Feature Fusion

When it comes to data of "cross-modality" (i.e., 3D point cloud and 2D image), the main difficulty is to fuse information provided by different modalities, which is a well-recognized challenge in various tasks such as 3D object detection, point cloud registration, and shape completion. ImVoteNet [77] fuses 2D detection into a 3D pipeline with camera parameters and pixel depths. DeepI2P [78] utilizes the attention mechanism to weight image features by point clouds. Image2Point [79] investigates the potential for transferability between single images and points. In the same problem setting, ViPC [3] also infers the missing points guided by an additional image. It directly maps a color image to the point cloud domain, explicitly aligns the generated rough geometry, and finally fine-tunes the coarse geometry. Although exploiting multi-modal information, ViPC heavily relies on the 3D reconstruction from a single image as well as requires an accurate alignment. Different from ViPC, our CSDN formulates multi-modal fusion as a shape style transfer problem, and it disentangles different types of features with a dual-refinement module.

3 METHOD

3.1 Overview

We assume in point cloud completion with a cross-modal manner that 1) images supply both the global structure and local-detail information to facilitate the inference of a complete and geometry-rich shape, and 2) images should participate in the whole coarse-to-fine completion procedure. To this end, we design a Cross-modal Shape-transfer Dual-refinement Network (CSDN). Following the encoder-decoder architecture to implement the cross-modal coarse-to-fine effort, CSDN is designed to have three modules: feature extraction, shape fusion, and dual-refinement, as illustrated in Figure 2. Denote $P_{in} = \{p_1, ..., p_N\} \subseteq \mathbb{R}^3$ with the size of N to be an input partial point cloud, and $\mathcal{I}$ with the size of $H \times W$ to be the corresponding single-view color image. Our goal is to predict a point cloud P_{out} from its dual-domain partial observations P_{in} and $\mathcal{I}$, and P_{out} represents the complete underlying shape of the object.

3.2 Feature Extraction

CSDN embeds both the pair of a partial point cloud and a single-view image into their individual feature space. Since we adopt a coarse-to-fine strategy (complete its overall shape first, and then complete its detail), we intend to extract global features from both P_{in} and I in this stage: 1) for P_{in}, the simple yet effective PointNet [9] is leveraged to yield its global feature F_p with the size of $1 \times C$; 2) for I, a sub-network with seven convolutional layers is utilized to extract $7 \times 7 \times C$ feature maps and then obtain its global feature F_I through average pooling.

3.3 Shape Fusion

We design the shape fusion module to fuse the features of different modalities, which outputs a coarse but complete set of shape points. By proposing an image and point cloud co-supported

AdaIN [6] (IPAdaIN for short), we transfer the intrinsic geometric style from the image to the point cloud.

This module takes F_p and F_I as input and outputs a coarse point cloud P_0 (see Figure 3). Specifically, similar to FoldingNet [80], we first sample N'_r 2D points from a unit square $[0,1]^2$. Each of them is appended with the global point cloud feature F_p. Then, the concatenated feature is combined with the global image feature F_I to fold the 2D grids to a point surface, by the IPAdaIN layer. Note that we repeat (but without sharing the network parameters) the above operations M times to reconstruct M surfaces, similar to [50], [61]. Finally, all of them are concatenated to synthesize a complete complex point cloud P_0 with N_r points, i.e., $N_r = M \times N'_r$. M is a hyperparameter fixed to 4 in our experiments. Also, we can change K and N'_r to adapt the proposed method to the point clouds of different resolutions.

IPAdaIN. Inspired by StyleGAN [7] that employs AdaIN for the style transfer task and SpareNet [81] that generates points with a similar batch normalization operation, we reformulate the "cross-modality" fusion as a shape style transfer problem. The core idea is that the affine parameters in Instance Normalization generated by different image features can change the point-wise feature statistics to varying values and thus normalize the output point cloud to different shapes. Meanwhile, the global features learned by convolutional neural networks can describe the overall shape characteristics. Accordingly, we propose IPAdaIN, a new solution that is built based on the instance normalization layer to better capture the image style. Specifically, we feed the concatenated feature to multiple MLPs, where each MLP is followed by an IPAdaIN layer and generates a new point-wise feature F_{in}. By denoting $F_{in} \subseteq \mathbb{R}^{B \times N \times C}$ (B stands for the batch size during training), the normalization process is formulated as

$$
\begin{aligned}
IPAdaIN(F_{in}, F_I) &= \gamma\left(\frac{F_{in} - \mu(F_{in})}{\sigma(F_{in})}\right) + \beta \\
\gamma &= L_a(F_I) \\
\beta &= L_b(F_I)
\end{aligned}, \quad (1)
$$

where $\gamma \subseteq \mathbb{R}^{B \times 1 \times C}$ and $\beta \subseteq \mathbb{R}^{B \times 1 \times C}$ are the affine parameters computed by non-linear mappings implemented by two MLPs. $\mu(F_{in}) \subseteq \mathbb{R}^{B \times 1 \times C}$ and $\sigma(F_{in}) \subseteq \mathbb{R}^{B \times 1 \times C}$ are the means and standard deviations of the channel-wise activation of F_{in}, respectively

$$
\begin{aligned}
\mu_{bc}(F_{in}) &= \frac{1}{N} \sum_{n=1}^{N} f_{bcn} \\
\sigma_{bc}(F_{in}) &= \sqrt{\frac{1}{N} \sum_{n=1}^{N} (f_{bcn} - \mu_{bc}(F_{in}))^2 + \epsilon}
\end{aligned}. \quad (2)
$$

Unlike AdaIN, IPAdaIN guides shape folding with the learned shape characteristics, which can be interpreted as a way to re-
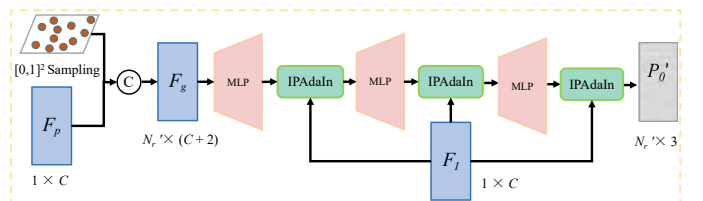

Fig. 3. Illustration of the IPAdaIN layer. IPAdaIN normalizes features of the folding operation by global image features.

construct a 3D shape from its style contained in the image. Considering that different feature channels of an image can identify certain shape characteristics, the generated affine parameters γ and β will have high activation in varying channels according to the 3D shapes. Then the point-wise features change in every channel according to Eq. 1. The fused features are then decoded back to the spatial space by the 3D decoder of the folding operation.
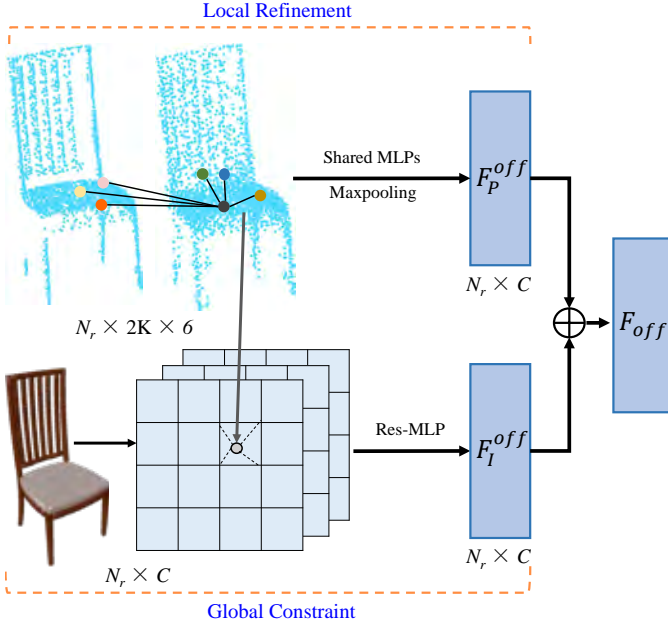

Fig. 4. Illustration of the dual-refinement module. The upper path denotes Local Refinement and the lower one denotes Global Constraint. Each sub-network generates an offset feature, and they are added to yield the feature F_{off}.

3.4 Dual-Refinement

The dual-refinement module aims to generate a set of coordinate offsets ΔC with the dimension of $N_r \times 3$ to fine-tune the coarse point cloud P_0. The whole module contains two units named Local Refinement and Global Constraint, respectively. The local refinement finds the relation between points in P_0 and P_{in} and generates one set of offset features, as shown in Figure 4. To refine P_0 without being corrupted by missing parts, the global constraint exploits image features to generate another set of offset features. Finally, the combined feature is fed into the Offset Regression unit that is built with an MLP to obtain the set of offsets ΔC. Thus, the two units of local refinement and global constraint complement each other to capture both the global structures and local details of the completed object.

3.4.1 Local Refinement

It is a very common situation that some missing high-frequency details already appear in the partial point cloud. Thus an intuition is to exploit local structure information with the aid of the partial point cloud. The local refinement unit is to integrate the structure information of the input partial shape with the coarse point cloud itself. Since a point cloud can be naturally regarded as a graph-like structure, we build this unit with graph convolution, given its effectiveness in point cloud analysis [17].

A directed graph $\mathcal{G}(\mathcal{V}, \mathcal{E})$ is established from the partial point cloud P_{in} and the coarse point cloud P_0, where $\mathcal{V} =$

$\{1, ..., N + N_r\}$ represents the set of nodes, and $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$ is the set of edges. Observing that certain areas in P_{in} have much higher quality than those in P_0, like the chair back and the three complete legs in Figure 4, we build $\mathcal{E}$ encoding the structure information from both P_0 and P_{in}. Hence, for each point in P_0, its k-nearest neighbours are respectively searched in P_{in} and P_0 to build a local graph which has 2k vertices (see Figure 4). By denoting the central point in the graph convolution as p_i and the neighborhood set as $\mathcal{N}(i) = \{j|(i, j) \in \mathcal{E}\}$, we can formulate *edge features* as $e_{ij} = \sigma([p_i, p_j - p_i])$, where σ is a non-linear function implemented by Shared-MLP and $[\cdot, \cdot]$ is a concatenation operation. Finally, each point's feature is aggregated by

$$f_i = \max_{j \in \mathcal{N}(i)} e_{ij}, \tag{3}$$

and

$$F_P^{off} = \{f_i\}_{i=1}^{N_r}, \tag{4}$$

where max denotes a channel-wise max-pooling function and f_i is the feature of the i-th point.

3.4.2 Global Constraint

We query k-nearest neighbors from the partial point cloud P_{in} for each point in P_0. Inevitably, the points in the generated missing areas cannot contain a reliable local graph similar to the corresponding local shape in the ground-truth point cloud. Meanwhile, RGB images usually contain underlying 3D-aware shape properties with high confidence (e.g., boundaries, textures, and local connectivity) [4]. To this end, we propose the global constraint unit to fix F_P^{off} with learned image features. The unit first projects 3D points into the last four feature maps (from 2D encoders) by the camera parameters and pooling them from four nearby pixels using bilinear interpolation (see the bottom of Figure 4). The obtained point-wise feature is then fed into a residual MLP block to obtain the offset feature F_I^{off}. Finally, we add F_I^{off} and F_P^{off} to obtain F_{off}.

3.5 Loss Function

The loss function measures the difference between P_{out} and the ground truth P_{gt}. We utilize the Chamfer distance (CD) as the loss function. To achieve the coarse-to-fine generation process, we regularize the training by calculating the loss function as

$$\mathcal{L} = \mathcal{L}_{CD}(P_0, P_{gt}) + \alpha\mathcal{L}_{CD}(P_{out}, P_{gt}), \tag{5}$$

where $\mathcal{L}_{CD}$ is defined as

$$\mathcal{L}_{CD}(X, Y) = \frac{1}{|X|}\sum_{x \in X}\min_{y \in Y}||x - y|| + \frac{1}{|Y|}\sum_{y \in Y}\min_{x \in X}||y - x||. \tag{6}$$

In the implementation, we increase α from 0.01 to 2.0 during the first 30k iterations since the initial point set P_0 is less accurate.

4 EXPERIMENT

4.1 Datasets

4.1.1 Training Dataset

We use the benchmark dataset ShapeNet-ViPC [3], which is derived from ShapeNet [85]. It contains 13 categories and 38,328 objects, including airplane, bench, cabinet, car, chair, monitor, lamp, speaker, firearm, sofa, table, cellphone, and watercraft. The complete ground-truth point cloud is generated by uniformly

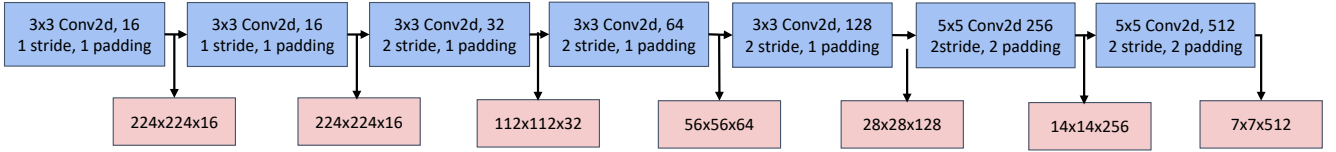| 3x3 Conv2d, 16 1 stride, 1 padding	3x3 Conv2d, 16 1 stride, 1 padding	3x3 Conv2d, 32 2 stride, 1 padding	3x3 Conv2d, 64 2 stride, 1 padding	3x3 Conv2d, 128 2 stride, 1 padding	5x5 Conv2d 256 2stride, 2 padding	5x5 Conv2d, 512 2 stride, 2 padding
224x224x16	224x224x16	112x112x32	56x56x64	28x28x128	14x14x256	7x7x512

Fig. 5. The architecture of our 2D Encoder.

TABLE 1
Quantitative results on known categories of ShapNet-ViPC using CD with 2,048 points. The best is highlighted in bold.

Methods	Mean CD per point $\times 10^{-3}$ (lower is better)								
	Avg	Airplane	Cabinet	Car	Chair	Lamp	Sofa	Table	Watercraft
AtlasNet [61]	6.062	5.032	6.414	4.868	8.161	7.182	6.023	6.561	4.261
FoldingNet [80]	6.271	5.242	6.958	5.307	8.823	6.504	6.368	7.080	3.882
PCN [30]	5.619	4.246	6.409	4.840	7.441	6.331	5.668	6.508	3.510
TopNet [49]	4.976	3.710	5.629	4.530	6.391	5.547	5.281	5.381	3.35
PF-Net [51]	3.873	2.515	4.453	3.602	4.478	5.185	4.113	3.838	2.871
MSN [50]	3.793	2.038	5.06	4.322	4.135	4.247	4.183	3.976	2.379
GRNet [47]	3.171	1.916	4.468	3.915	3.402	3.034	3.872	3.071	2.160
PoinTr [8]	2.851	1.686	4.001	3.203	3.111	2.928	3.507	2.845	1.737
ViPC [3]	3.308	1.760	4.558	3.138	**2.476**	2.867	4.481	4.99	2.197
PointAttN [82]	2.853	1.613	3.969	3.257	3.157	3.058	3.406	2.787	1.872
SDT [83]	4.246	3.166	4.807	3.607	5.056	6.101	4.525	3.995	2.856
Seedformer [84]	2.902	1.716	4.049	3.392	3.151	3.226	3.603	2.803	**1.679**
Ours	**2.570**	**1.251**	**3.670**	**2.977**	2.835	**2.554**	**3.240**	**2.575**	1.742

TABLE 2
Quantitative results on known categories of ShapNet-ViPC using F-Score with 2,048 points. The best is highlighted in bold.

Methods	F-Score@0.001 (higher is better)								
	Avg	Airplane	Cabinet	Car	Chair	Lamp	Sofa	Table	Watercraft
AtlasNet [61]	0.410	0.509	0.304	0.379	0.326	0.426	0.318	0.469	0.551
FoldingNet [80]	0.331	0.432	0.237	0.300	0.204	0.360	0.249	0.351	0.518
PCN [30]	0.407	0.578	0.27	0.331	0.323	0.456	0.293	0.431	0.577
TopNet [49]	0.467	0.593	0.358	0.405	0.388	0.491	0.361	0.528	0.615
PF-Net [51]	0.551	0.718	0.399	0.453	0.489	0.559	0.409	0.614	0.656
MSN [50]	0.578	0.798	0.378	0.380	0.562	0.652	0.410	0.615	0.708
GRNet [47]	0.601	0.767	0.426	0.446	0.575	0.694	0.450	0.639	0.704
PoinTr [8]	0.683	0.842	0.516	0.545	0.662	0.742	0.547	0.723	0.780
ViPC [3]	0.591	0.803	0.451	0.5118	0.529	0.706	0.434	0.594	0.73
PointAttN [82]	0.662	0.841	0.483	0.515	0.638	0.729	0.512	0.699	0.774
SDT [83]	0.473	0.636	0.291	0.363	0.398	0.442	0.307	0.574	0.602
Seedformer [84]	0.688	0.835	**0.551**	0.544	0.668	**0.777**	0.555	0.716	**0.786**
Ours	**0.695**	**0.862**	0.548	**0.560**	**0.669**	0.761	**0.557**	**0.729**	0.782

sampling 2048 points on the mesh surface from ShapeNet [85]. Each object has a complete point cloud, 24 rendered images under 24 viewpoints, and 24 corresponding incomplete point clouds. The incomplete point cloud is generated from the corresponding viewpoint (with occlusion) and also contains 2048 points. The image data is rendered from 24 viewpoints as ShapeNetRendering in 3D-R2N2 [63]. During training, the image viewpoint is randomly chosen for each training data pair and the point cloud will be aligned with the chosen image. For a fair comparison, we use the same training setting as in ViPC [3], i.e., 80% of the eight categories for training.

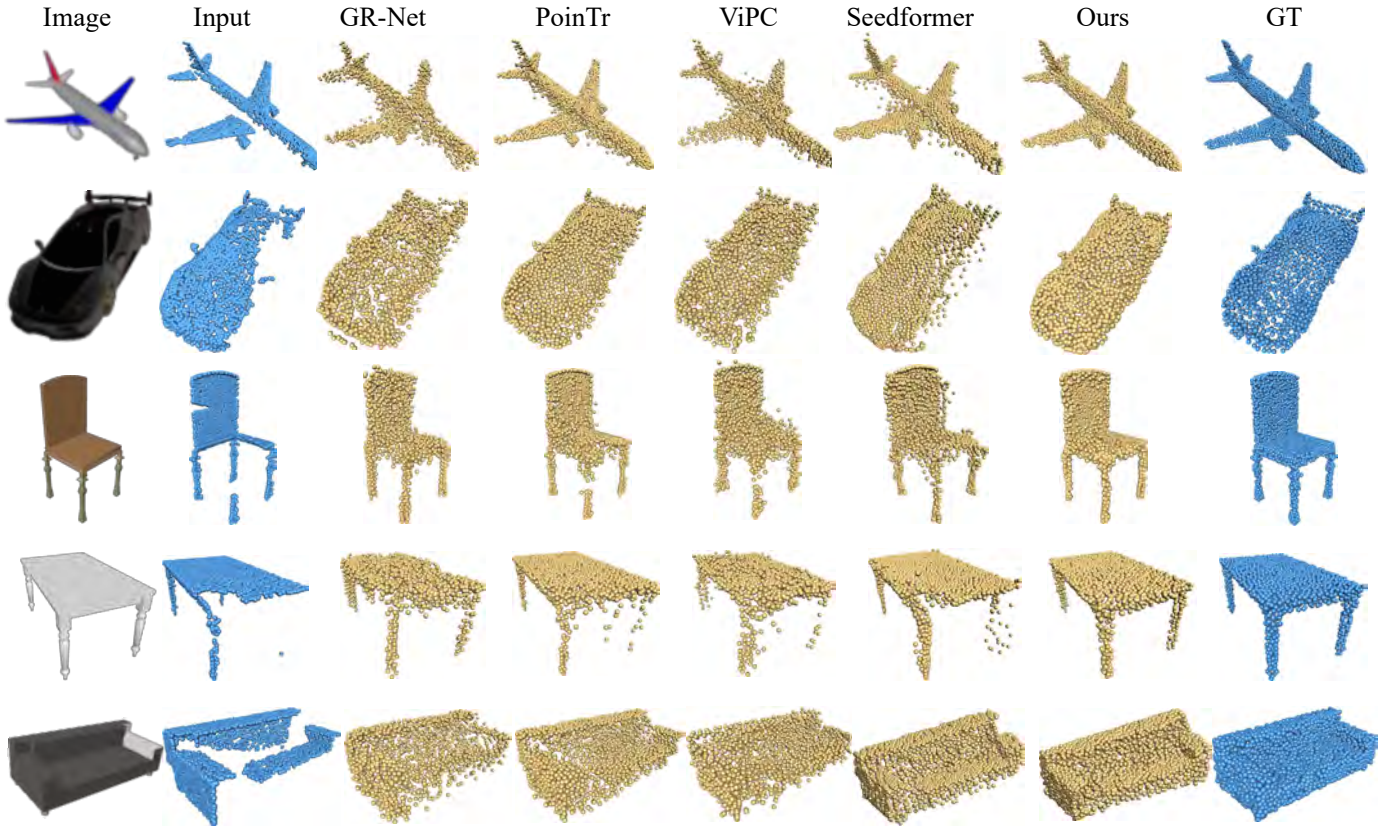| Image	Input	GR-Net	PoinTr	ViPC	Seedformer	Ours	GT

Fig. 6. Visual comparisons of recent point cloud completion methods [3], [8], [47], [84] on ShapeNet-ViPC [3]. CSDN produces the most complete and detailed structures compared to its competitors.

4.1.2 Test Dataset

The test dataset from ShapeNet-ViPC [3] consists of two parts: one contains the remaining 20% of the 8 categories of objects for training; another one contains 4 categories that are not used during training. For each of the new categories, we randomly choose 200 pairs of point clouds and images. For a fair comparison, we follow the same setting as [3], where images are randomly selected from the 24 viewpoints. Notably, the ShapeNet-ViPC dataset produces missing shapes by various kinds of occlusions (not limited to self-occlusion). Therefore, images from different viewpoints do not necessarily depict the missing parts. We test our approach when images and partial point clouds are produced under the same viewpoints. The results are reported in Section 4.7.

Besides, we test some methods on partial point clouds from real-world LiDAR scans, the KITTI dataset [86] and RGB-D scans, the Scannet dataset [87]. Following the setting of [30], we extract point clouds within the corresponding bounding boxes and then select point clouds having more than 2048 points.

4.2 Implementation Details

In our implementation, the number of point surfaces in Shape Fusion is $k = 4$ and each surface contains 512 points. The feature dimension is set to 1024, i.e., C = 1024. Thus, the generated coarse point cloud has 2048 points. For IPAdaIN, we set three layers for every surface. In the Local Refinement unit, the number of nearest neighbors is 16. In the Global Constraint unit, the size of feature maps used for projection are 56×56, 28×28, 14×14, and 7×7.

For the 3D encoder, we set a five-layer MLP. The feature dimensions are 64, 64, 64, 128, and 1024, respectively. For the

2D encoder, its architecture is shown in Figure 5. Note that we reuse the last four feature maps in the global constraint unit and their dimensions are 56×56, 28×28, 14×14, and 7×7, respectively. In the local refinement unit, the feature F_P^{off} is produced by a three-layer MLP whose output dimensions are 32, 128, and 512, respectively. Res-MLP in the global constraint unit is implemented by a Con1d ResNet block [88]. The final offset regression is implemented by four Conv1d layers whose output dimensions are 256, 128, 32, and 3, respectively. The coordinate offset is calculated through the *tanh* activation function.

For quantitative evaluation, we use both the Chamfer Distance (CD) and F-Score as evaluation metrics. The whole network is trained end-to-end with a learning rate of 5×10^{-5} for 50 epochs and the Adam optimizer. The learning rate also decayed by 0.1 for every 10 epochs. All the networks are implemented using PyTorch and trained on an NVIDIA RTX 3090 GPU.

4.3 Result on ShapeNet-ViPC

We compare our CSDN with ten state-of-the-art point cloud completion methods. They are Point-based, i.e., AtlasNet [61], PCN [30], MSN [50], Transformer-based, i.e., PoinTr [8], Seedformer [84], SDT [83], Folding-based, i.e., FoldingNet [80], TopNet [49], GAN-based, i.e., PF-Net [51], Convolution-based, i.e., GRNet [47], and Cross-modality-based, i.e., ViPC [3]. For AtlasNet, FoldingNet, PCN, and TopNet, we directly take the quantitative statistics from [3]. For the other methods, we retrain their models using the released codes on ShapeNet-ViPC. For a fair comparison, we uniformly downsample the results of PoinTr to 2048 points, the same as the output of the other methods.
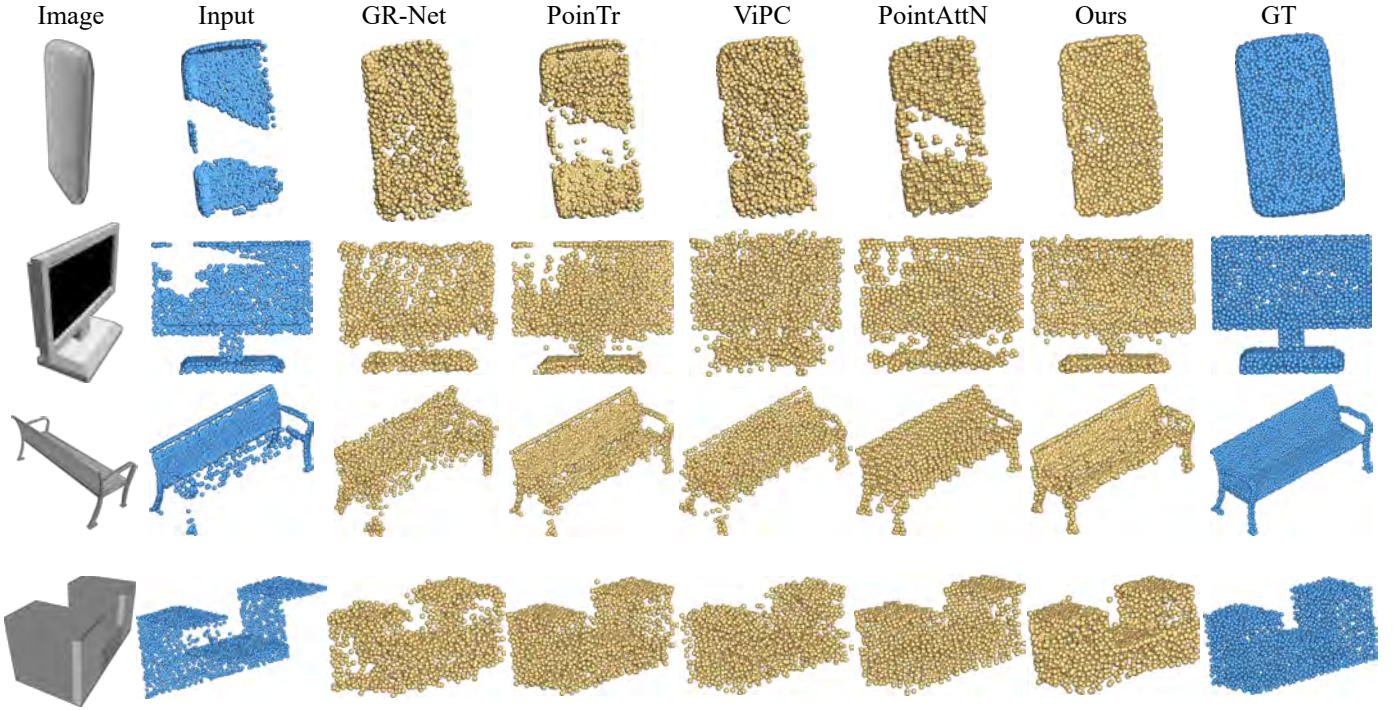
Fig. 7. Visual comparisons of recent point cloud completion methods [3], [8], [47], [82] on unseen categories of ShapeNet-ViPC [3]. CSDN produces the most complete and detailed structures compared to its competitors.

Among these methods, ViPC also exploits an additional image, while the others take only a partial point cloud as input.

4.3.1 Qualitative Results on Known Categories

Figure 6 shows the point clouds completed by the proposed CSDN and its competitors. For some cases of symmetric structures and small missing regions (e.g., plane and car), most methods behave well and generate complete shapes similar to the ground truths. Our CSDN recovers more clear overall shapes and more detailed structures than Seedformer, PoinTr, and ViPC, while GR-Net generates noisy points. For those challenging cases of complex shapes and large missing regions, like the table case, our CSDN generates the complete desktop and leg while the other methods only predict a coarse result. Compared to ViPC, our CSDN is better to recover latent geometric structures (e.g., holes in the desktop and the back of the chair).

4.3.2 Quantitative Results on Known Categories

We calculate both CD and F-Score on the 2,048 points of each object, as reported in Table 1 and Table 2. It can be observed that our CSDN achieves almost the best results among all the competitors. It is worth noting that CSDN reduces the average CD value by 0.281 compared to the result of the second-ranked PoinTr [8]. Also, CSDN outperforms ViPC [3] by a large margin of 22.3%. Meanwhile, CSDN achieves the best results in all categories in terms of F-Score.

4.3.3 Results on Novel Categories

We test different methods on four novel categories which are not used for training, to evaluate the category-agnostic ability. The quantitative results are reported in Table 3 and Table 4. PoinTr [8] and PointAttN [82] achieve competitive results similar to their results on known categories. ViPC [3] struggles to generalize to

TABLE 3
Quantitative results on the Novel categories of ShapNet-ViPC using CD with 2,048 points. The best is highlighted in bold.

Methods	Mean Chamfer Distance per point $\times 10^{-3}$				
	Avg	Bench	Monitor	Speaker	Phone
PF-Net [51]	5.011	3.684	5.304	7.663	3.392
MSN [50]	4.684	2.613	4.818	8.259	3.047
GRNet [47]	4.096	2.367	4.102	6.493	3.422
PoinTr [8]	3.755	1.976	4.084	5.913	3.049
ViPC [3]	4.601	3.091	4.419	7.674	3.219
PointAttN [82]	3.674	2.135	**3.741**	5.973	**2.848**
SDT [83]	6.001	4.096	6.222	9.499	4.189
Ours	**3.656**	**1.834**	4.115	**5.690**	2.985

new shapes, while our CSDN generalizes well to novel shapes that have not been seen during training.

We also present a visual comparison with other methods, including two Transformer-based methods PoinTr [8] and PointAttN [82], one Convolution-based method GR-Net [47], and one Cross-modality-based method ViPC [3], as shown in Figure 7. We can observe that the two Transformer-based methods cannot well recover the missing shapes for previously unseen categories, although they achieve competitive performance on quantitative results. Under the guidance of images, CSDN and ViPC both succeed in inferring the missing parts, and CSDN produces better details and fewer noisy points. This comparison further demonstrates that our method successfully exploits the complementary information provided by the images.

TABLE 4
Quantitative results on the Novel categories of ShapNet-ViPC us[...]
F-Score with 2,048 points. The best is highlighted in bold.

Methods	F-Score@0.001			
	Avg	Bench	Monitor	Speaker
PF-Net [51]	0.468	0.584	0.433	0.319
MSN [50]	0.533	0.706	0.527	0.291
GRNet [50]	0.548	0.711	0.537	0.376
PoinTr [8]	0.619	0.797	**0.599**	0.454
ViPC [3]	0.498	0.654	0.491	0.313
PointAttN [82]	0.605	0.764	0.591	0.428
SDT [83]	0.327	0.479	0.268	0.197
Ours	**0.631**	**0.798**	0.598	**0.485**

4.4 Evaluation on Real-world Scans

We train our CSDN and two other methods on the[...]
ShapeNet-ViPC to evaluate their performance on r[...]
Ideally, a point cloud completion network trained c[...]
dataset should produce satisfactory results, since t[...]
corresponding real-world images cropped with 2D[...]
are provided. As reported in [3], the real-to-synth[...]
led by rendered images in the training dataset r[...]
applicable to real-world point cloud completion. '[...]
the input image with the rendered car image in[...]
for ViPC [3] and our method. We also evaluate[...]
the chairs and tables from the Scannet dataset[...]
objects in Scannet are usually more complete and[...]
in the training datasets, we directly use the tr[...]
Section 4.3 without any fine-tuning. Figure 8 s[...]
still cannot handle the domain gap problem and F[...]
coarse results due to the low resolution of the tra[...]
Figure 9, PoinTr fails to complete the missing [...]
cannot preserve the original inputs. In contrast, C[...]
of generating more plausible results on real-worl[...]

4.5 Ablation Study

We first remove and change the main component[...]
The ablation variants can be classified as ablation on Shape Fusion
and Dual Refinement. Then, we ablate on the input modality and
analyze the contribution of each modality.

4.5.1 Ablation on Shape Fusion

To analyze the contribution of IPAdaIN, the comparison between
three variants on the Shape Fusion module is shown in Table 5.
We replace IPAdaIN in the variant A with a regular instance nor-
malization. Without the global information provided by images, A
cannot guarantee the completeness of the final shape and typically
generates point clouds with higher CD and lower F-Score. The
variant B, switching the positions of F_P and F_I, also produces
points with higher CD and lower F-Score. In the variant C, we only
utilize the image feature for the folding operation. Reconstructing
shapes directly from images like [3] leads to more performance
degradation. It further reveals the superiority of IPAdaIN over
the single-view reconstruction method in terms of feature fusion.
Moreover, from the similar performance of A and C, as well as
the coarse result of CSDN, we can conclude that IPAdaIN plays
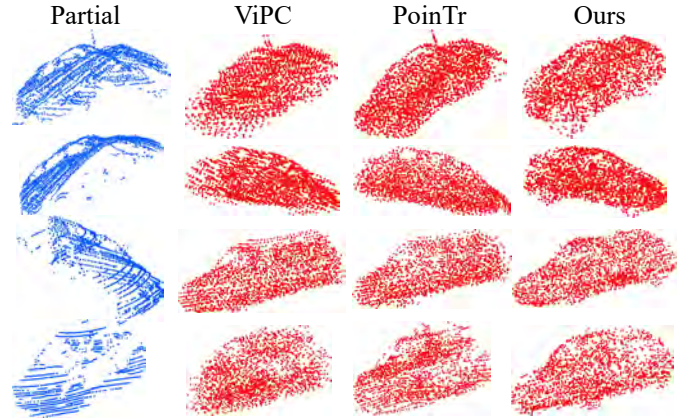the core role in the proposed cross-modal fusion strategy.

Fig. 8. Visual comparisons with two point cloud completion methods [3],
[8] on the real-scanned point clouds (from KITTI [86]). CSDN produces
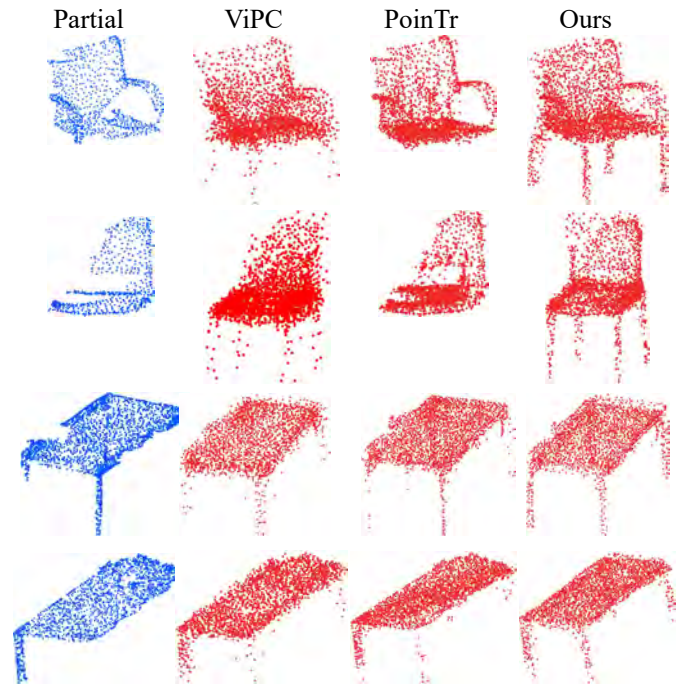the most complete and detailed structures.

Fig. 9. Visual comparisons with two point cloud completion methods [3],
[8] on the real-scanned point clouds from Scannet [87]. CSDN produces
the most complete and detailed structures.

4.5.2 Ablation on Dual-Refinement

Table 6 shows the comparison between the variants of CSDN on
Dual-Refinement. The variant D, without the Local Refinement
unit, leads to a large performance drop and coarser results. The
variant E, without the Global Constraint unit, typically has lower
performance. It means both the point cloud and image contribute
to the refinement step, but the point cloud plays a major role.
To understand the contribution of the double-refinement and dual-
branch design, in the variant F we replace the parallel structure
of dual-refinement with a serial structure (i.e., P_0 is sent to Local
Refinement for the first refinement, and the refined result is sent
to Global Constraint for final results). In contrast to the Dual-
Refinement module, F only uses information from both modalities
for separate refinement without exploiting complementary infor-
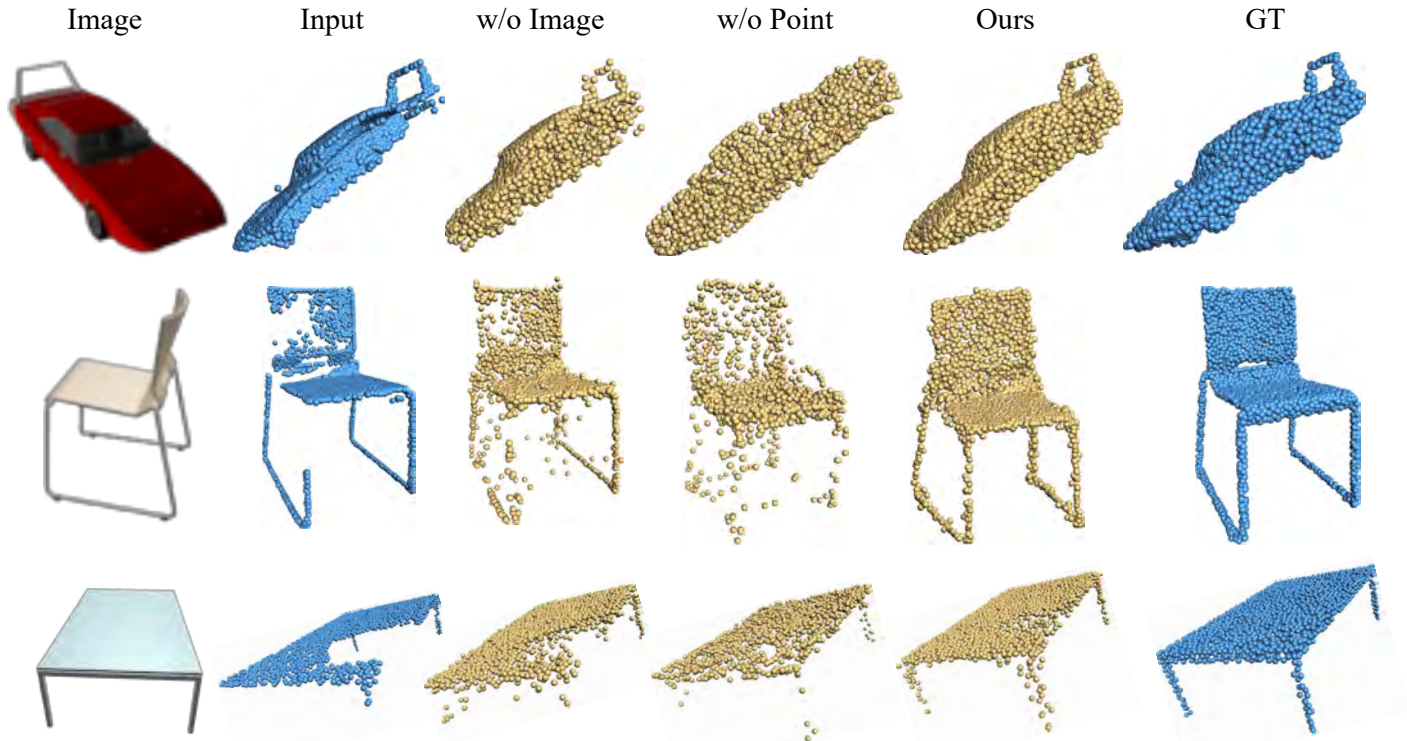mation from each modality. The performance drop of F demon-

Fig. 10. Visual comparisons between CSDN and its variants when the input images cannot depict the missing parts.

TABLE 5
Comparisons between CSDN and its variants on Shape Fusion.

Methods	Known		Novel	
	CD $\times 10^{-3}$	F-Score	CD $\times 10^{-3}$	F-Score
w/o IPAdaIN (A)	3.658	0.648	4.528	0.602
SW (B)	2.647	0.681	3.765	0.625
ImgFolding (C)	4.174	0.631	5.142	0.584
Coarse	3.752	0.632	6.757	0.548
Ours (CSDN)	**2.570**	**0.695**	**3.656**	**0.631**

TABLE 6
Comparisons between CSDN and its variants on Dual-Refinement.

Methods	Known		Novel	
	CD $\times 10^{-3}$	F-Score	CD $\times 10^{-3}$	F-Score
w/o LG (D)	3.428	0.648	6.284	0.561
w/o GC (E)	2.700	0.687	4.015	0.624
serial (F)	3.036	0.659	4.227	0.611
Coarse	3.752	0.632	6.757	0.548
Ours (CSDN)	**2.570**	**0.695**	**3.656**	**0.631**

strates the superiority of the proposed two-branch architecture.

4.5.3 Ablation of Input Modality

This set of ablative experiments aims at exploring the role of point clouds and images in our method. First, we change our network into single-modal versions and analyze the contribution of different input modalities. Specifically, we ablate on the single-view images in the variant G, where both IPAdaIN and Global Constraint are removed. The variant H is a single-view reconstruction network based on CSDN. We replace the shape fusion module with a vanilla foldingnet that uses only image features to reconstruct a coarse point cloud and remove the Local Refinement unit in the dual-refinement module. Then, we analyze their performance from two aspects: (1) missing geometry completion, and (2) original structure preservation.

Missing Geometry Completion Table 7 shows that the variants G and H have varying degrees of performance loss. While G still has competitive results on both known and novel categories, H has a large-margin performance drop (156% on known categories and 231% on novel categories). Besides, these results further demonstrate that our method is an image-guided point cloud completion network, where point clouds play the major role.

We demonstrate some cases when images cannot depict the missing parts in point clouds and visually compare the completion results of CSDN and its variants G and H in Figure 10. More results under such a condition can be found in Figure 18. We can see that G struggles to infer the missing parts. Meanwhile, the single-view reconstruction H can only reconstruct coarse shapes with a considerable number of noisy points. Both the variants cannot recover the structures that are unseen in images (e.g., the wheels of the car, the upper part of the chair legs and the table legs). In contrast, CSDN can generate smoother detailed structures that are missing in both the input point clouds and images, rather than just reconstructing geometric details from images. We credit such superiority to our feature fusion strategy.

In Figure 11, we show one more visual comparison between the final and coarse results of CSDN and its variant G. It can be observed that without the input images our method struggles to infer the missing regions and the coarse point clouds generated by Shape Fusion are already reasonably complete compared to the
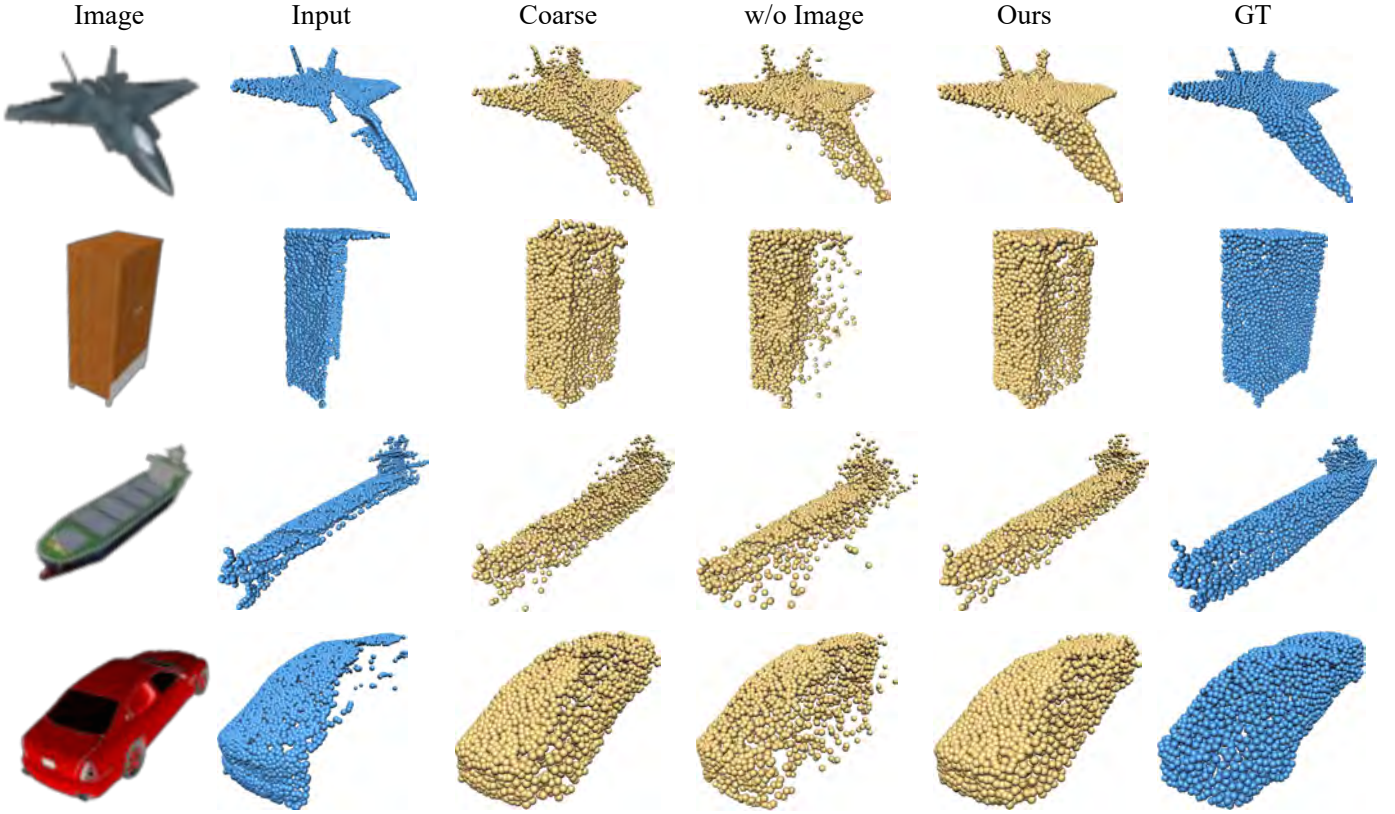
Fig. 11. Visual comparisons between CSDN and its variants.

corresponding partial inputs.

Original Structure Preservation To evaluate how our network preserves the initial structure of the objects, we also compare partial matching [59] value between P_{in} and the generated result P_{out}, which is single-side Chamfer Distance defined as

$$\mathcal{PM}(P_{in}, P_{out}) = \frac{1}{|P_{in}|} \sum_{x \in P_{in}} \min_{y \in P_{out}} ||x - y||. \quad (7)$$

Table 8 shows that the variant H struggles to preserve the partial shape since it uses only input images. However, our cross-modal network still has a better shape-preserving ability than the variant G that inputs only the partial point cloud. The comparison demonstrates that our method fuses cross-modal features to assist completion without losing the original shape information contained in point clouds. In other words, our method can efficiently extract and fuse the complementary information from the two modalities, rather than just reconstructing shapes from images. It is noteworthy that CSDN preserves initial points better than ViPC [3], which could also be concluded from Figures 6 and 18.

It is also worth noting that CSDN can preserve slender structures (e.g., the rear wing of the car) better than the two variants using a single modality input in Figure 10. To better reveal its capability in structure preservation, thin cross-sections of some models are demonstrated in Figure 12. We can see that CSDN preserves the original surfaces well while the other two competitors [3], [8] generate more noisy points inside the object.

4.5.4 Ablation on Numbers of KNN

We also conduct an ablation experiment on the number of nearest neighbors in the local refinement unit. k is set to be 4, 8, 16, and

TABLE 7
Comparisons between CSDN and its variants on the input modality.

Methods	Known		Novel	
	CD $\times 10^{-3}$	F-Score	CD $\times 10^{-3}$	F-Score
w/o Image (G)	4.179	0.629	4.974	0.588
w/o Point (H)	6.583	0.411	12.103	0.268
Coarse	3.752	0.632	6.757	0.548
Ours (CSDN)	**2.570**	**0.695**	**3.656**	**0.631**

TABLE 8
Comparisons of Partial Matching value. Lower value means a better preserving ability. G and H refer to the variants in Table 7.

Methods	Ours	G	H	ViPC [3]	PointAttN [82]
PM $\times 10^{-3}$	0.277	0.358	3.038	0.854	0.424

24, respectively. It is observed from Table 1 that CSDN has the best performance when $k = 16$. However, we believe that this parameter needs to be fine-tuned for different datasets.

4.6 Impact of Point Cloud Resolution

This section studies the stability of CSDN and two other competitors concerning point cloud resolution. To this end, we downsample the input point cloud to 1024 and 256 points respectively, and test the performance of the methods. From the results reported in Table 9, we observe that the performance drop of all three methods is small when the input is downsampled to 1024 points.
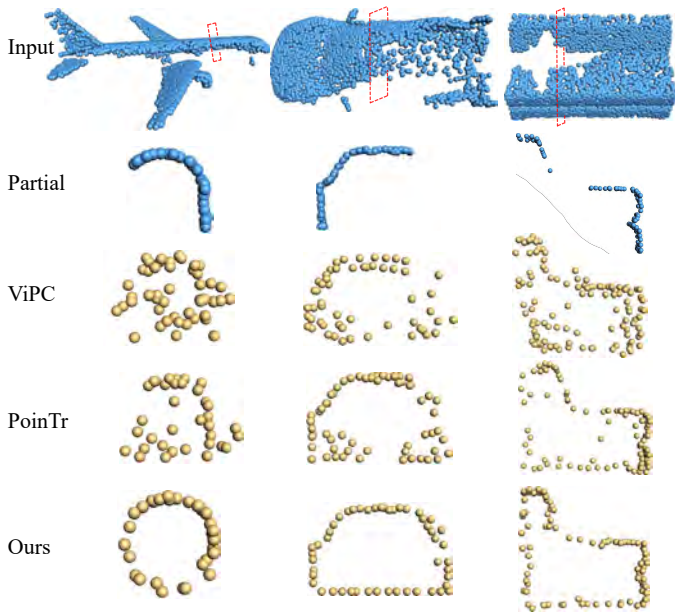
Fig. 12. Visual comparisons of CSDN with [3], [8] on structure preservation. Cross-sections of the completed point clouds are visualized. For each model, the 3D red rectangle in the first row indicates the position and orientation of the cross-section.

TABLE 9
Mean Chamfer distance with different point cloud resolutions. The first row denotes the number of input points.

Methods	Number of Input Points		
	2048	1024	256
Ours	2.570	2.597	3.358
PoinTr [8]	2.851	2.882	4.255
ViPC [3]	3.308	3.341	3.687

This is because the point cloud can still convey the geometry of the partial shape. When the input is downsampled to 265 points, the performance of all methods degrades significantly. It is worth noting that the decline of ViPC (11.5%) is smaller than that of CSDN (30.7%) and PoinTr (49.2%), which is due to that ViPC reconstructs points directly from images, while the input points play the major role in our method.

4.7 Effect of Different-view Images

This section studies the stability of CSDN when images can be collected from different views. Ideally, any good method of cross-modal point cloud completion should be view-independent. However, it is observed that images captured from different views may slightly affect the completion results. A "good" view is considered to possess complementary information for the input partial point clouds. The images with a good view tend to produce the completion results with a lower CD value.

To validate the aforementioned statement, we randomly select some partial objects and test them with the individual images from all 24 views as input. As observed in Figure 13, CSDN performs better on images that contain more information about the missing parts. For example, the image containing only the front or the end of a truck generates the results with the highest CD losses.

TABLE 10
Quantitative comparisons with different KNN numbers.

Methods	Known		Novel	
	CD $\times 10^{-3}$	F-Score	CD $\times 10^{-3}$	F-Score
K = 4	2.643	0.691	4.061	0.625
K = 8	2.636	0.688	4.053	0.622
K = 16	**2.570**	**0.695**	**3.656**	**0.631**
K = 24	2.656	0.685	4.116	0.624

To further demonstrate the stability of CSDN over the other cross-modal method, i.e. ViPC [3], we select 100 objects from each known category to evaluate how CSDN and ViPC perform with the images captured from different viewpoints. Specifically, we obtain 24 results for each incomplete object with the help of each image of the 24 different-view images. We calculate the standard deviations for the 24 CD values to compare the stability of model performance, as shown in Table 11. The quantitative results clearly show that our CSDN is more view-independent than ViPC [3] with a 42.9% drop in the standard deviation.

In addition, we also test our method when images and the corresponding partial point clouds are acquired under the same viewpoints, with the results shown in Table 12. The negligible performance drop (2.6% on the CD metric) indicates that CSDN is capable of extracting meaningful geometric information from images even when the images may not depict the missing parts of the objects. More visual comparisons can be found in Figure 18.

4.8 Effect of Different Similar-images

There may exist objects that have similar structures. Hence, we randomly select some models and replace the corresponding input images with similar images in the same category. Figure 14 shows the results of one example. When inputting the pair of the original image and the partial point cloud, CSDN generates the completion result with a CD loss of 0.707×10^{-3}. Then, we replace the original image with seven images of the other models in ShapeNet-ViPC [3]. It is found that our method still works with similar images as input but produces results with a higher CD loss. Figure 14 shows that the model performs better when the input image has similar structures and colors. However, the last two images have nearly the same shape as the original image, but our method generates results with higher CD losses. We believe that it is caused by the remarkable differences in colors. Please note that we do not use such unpaired data during the training stage.

4.9 Comparisons about Model Sizes

We compare and report the model size of ViPC, PoinTr, and our CSDN in Table 13. The comparison indicates that our CSDN has a slightly larger model size than ViPC but a much better performance, while PoinTr has the largest model size owing to its transformer architecture.

4.10 Feature Visualization

To better understand the role of IPAdaIN, this section provides visualizations of the features before and after IPAdaIN in the Shape Fusion module in two different ways. Specifically, we define the point-wise features before the last layer of IPAdaIN
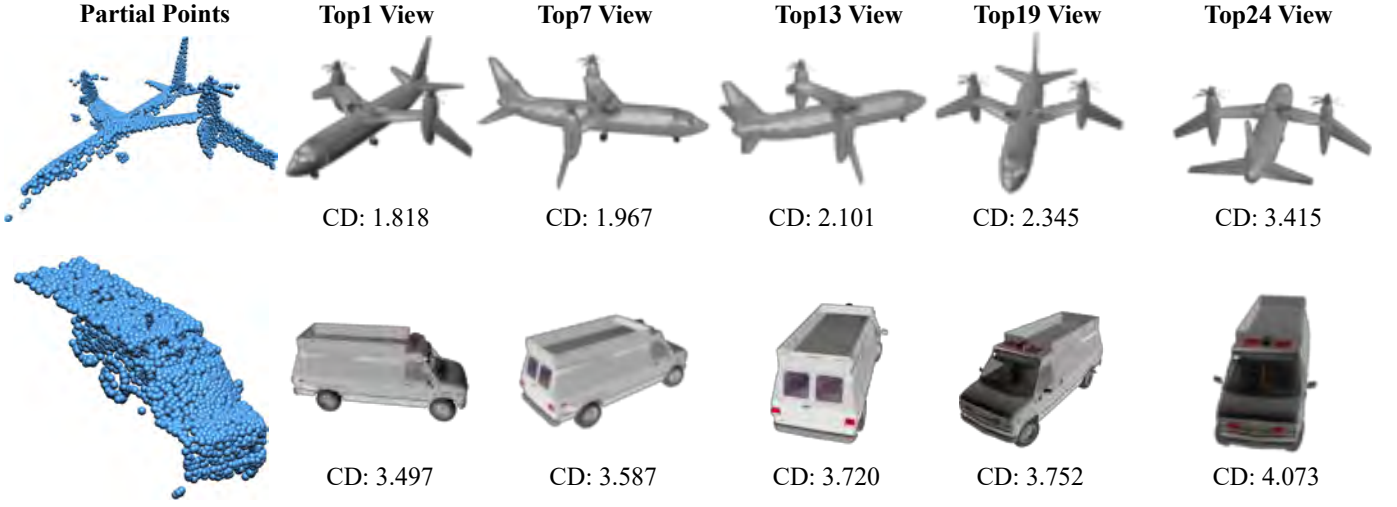
Fig. 13. Images captured from different views often lead to slightly different completion results. A "good" view that contains more complementary information for the input partial point clouds tends to produce the completion results with a lower CD value ($\times 10^{-3}$).
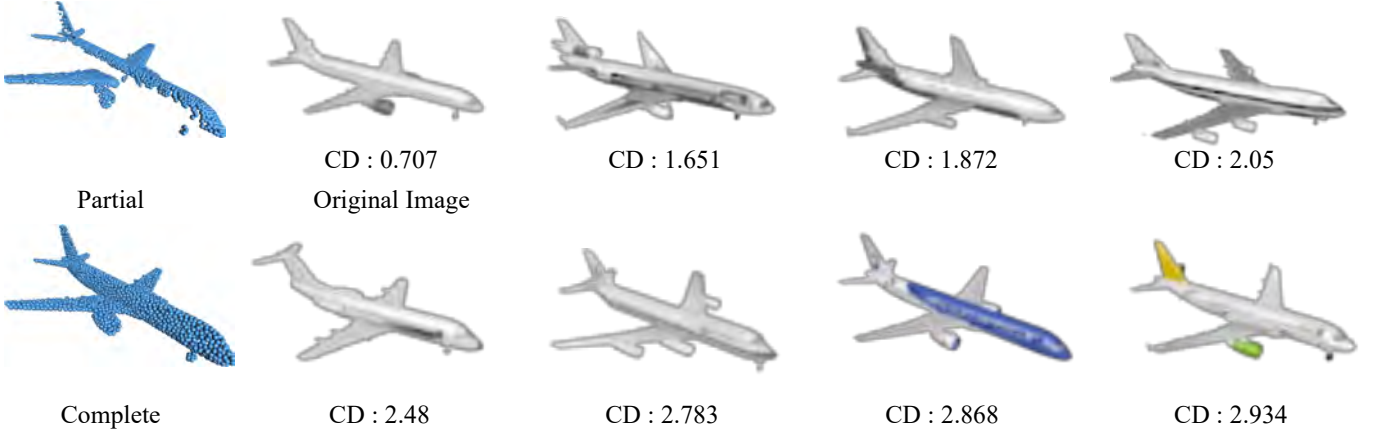

Fig. 14. Results of using similar but intrinsically different images to bind the partial point cloud for completion. The object's structures and colors
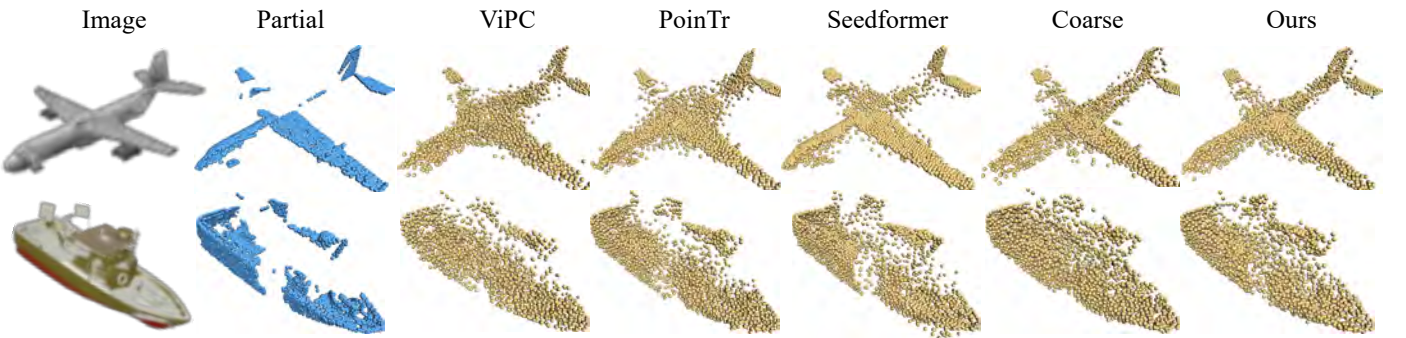

Fig. 15. Failure cases. Similar to its competitors [3], [8], [84], our CSDN may generate poor completion results (the small structures cannot be reconstructed) when the partial point cloud lacks the main body and the assembled image has very low resolution. Please note that in such challenging cases, our CSDN can still recover the overall shape, which its competitors are not.

as $F_{before} = \{f_1, ..., f_N\} \subseteq \mathbb{R}^C$, the corresponding point-wise features after IPAdaIN as F_{after}, and the generated coarse points as $P_{coarse} = \{p_1, ..., p_N\} \subseteq \mathbb{R}^3$. Note that the affine transformation in IPAdaIN is the only step where image features are involved for point generation in the Shape Fusion module.

We first explore how features change before and after

IPAdaIN. During the folding operation, points that are close in the 3D space should also be close in the feature space (i.e., f_a should have a relatively small Euclidean distance to f_b if p_a and p_b are spatially close). Figure 16 visualizes the coarse shapes P_{coarse} generated by the Shape Fusion module, and the target points were randomly selected in the originally missing parts.

TABLE 11
Standard deviations of CD on different-view images. The lower the value is, the more view-independent the method is.

Methods	Standard deviations of CD calculated on different images								
	Avg	Airplane	Cabinet	Car	Chair	Lamp	Sofa	Table	Watercraft
ViPC [3]	0.445	0.185	0.406	0.236	0.595	0.921	0.373	0.599	0.243
Ours	**0.254**	**0.119**	**0.272**	**0.164**	**0.349**	**0.277**	**0.305**	**0.316**	**0.184**

near [ ] far
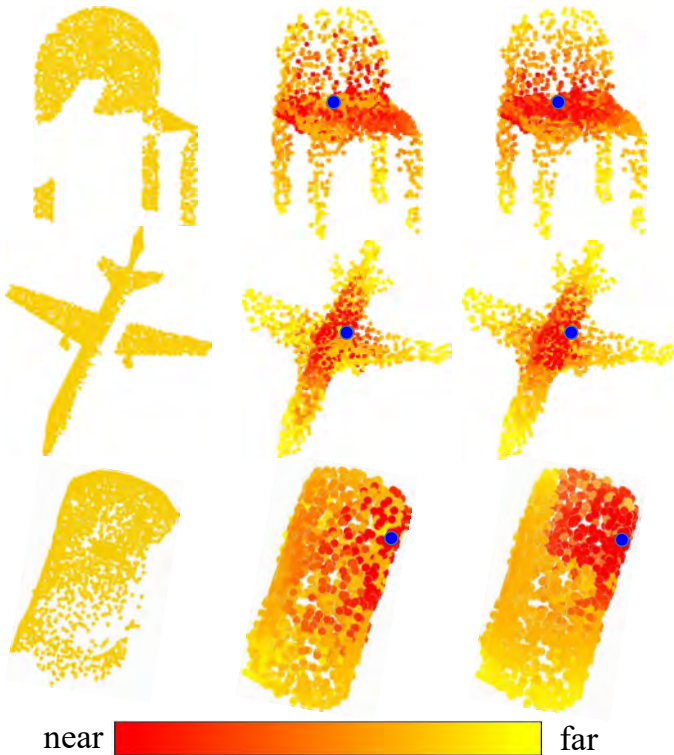
Fig. 16. Visualization of Euclidean distances between the target points (in blue) and other points in the feature space. For each set, Left: Input partial point cloud; Middle: Distance before IPAdaIN; Right: Distance after IPAdaIN. The target points are randomly selected in missing parts.

TABLE 12
Results on different experiment settings. "Different" means images are randomly selected from 24 viewpoints. "Same" means images and partial point clouds are produced from the same viewpoints.

Methods	Known		Novel	
	CD $\times 10^{-3}$	F-Score	CD $\times 10^{-3}$	F-Score
Different	2.570	0.695	3.656	0.631
Same	2.637	0.689	3.778	0.624

Then, we compute the Euclidean distances between the target point and other points in the feature space. The visualization is generated by coloring points based on the distances. As shown in Figure 16, points similar to the target point in the feature space are spread across the entire point cloud before IPAdaIN. After IPAdaIN, the similar points in the feature space become closer in the 3D space. It should be noted that IPAdaIN does not involve any learnable parameters other than the affine parameters generated by the image features. This means that the affine transformation
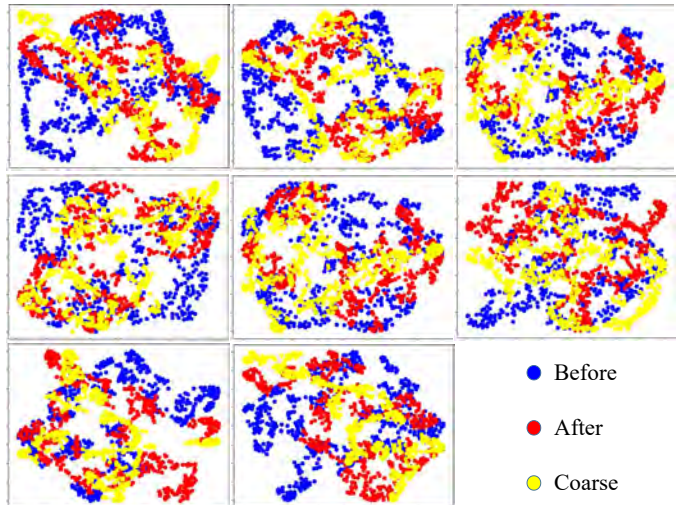

Fig. 17. Visualization of feature distributions by t-SNE: F_{before} (in blue), F_{after} (in red) and P_{coarse} (in yellow).

TABLE 13
Comparisons on model sizes and performance.

Methods	Model size (Mib)	Performance (CD)
ViPC [3]	53.79	3.308
PoinTr [8]	170.18	2.851
Ours	70.89	2.570

of IPAdaIN changes the feature distributions and normalizes the generated points to the complete shape.

In addition, we plot F_{before}, F_{after}, and P_{coarse} as 2D images by t-SNE [89] in Figure 17 to disclose the change in feature distribution after IPAdaIN. It can be concluded that IPAdaIN makes the distribution of point-wise feature F_{after} more similar to its direct result P_{coarse} than F_{before}. These results further explain that IPAdaIN fuses modalities by changing feature statistics.

5 LIMITATIONS AND FAILURE CASES

Table 14 summarizes the limitations among these compared methods to provoke further research insights. Besides, we visualize some failure cases whose CD losses are rather larger than the average error of that category, as shown in Figure 15. Although the shape fusion step succeeds in recovering the complete shape to a certain extent, and the dual-refinement module also removes noisy points, the final results are still less pleasing. This is especially the case when the partial point clouds have large missing regions both in their main bodies and detailed structures. Meanwhile, the input images are of very low resolutions, which can only provide limited

TABLE 14
Summary of the limitations of our CSDN and its competitors.

Methods	Limitation
AltasNet [61]	Rely on repeated reconstruction to represent a 3D shape as the union of several surface elements.
FoldingNet [80]	Overlook the local geometric characteristics.
PCN [30]	Only rely on a single global feature to generate points, hence incapable of synthesizing local details.
TopNet [49]	Results tend to have noisy points around the generated surface.
PF-Net [51]	Fail to recover local details to certain extent.
MSN [50]	Lack of sufficient refinement and fail to generate fine-grained details.
GRNet [47]	It is subject to the resolution of the voxel representation, which often leads to distortion in local complex structures.
PoinTr [8]	The model has relatively more parameters owing to the transformer architecture.
ViPC [3]	Rely on the 3D reconstruction from a single image as well as an accurate 3D alignment. The surface of the results is thus noisy.
Seedformer [84]	The results are not evenly distributed.
Ours	Rely on the dual-domain training dataset.

complementary information for the recovery of fine geometries (e.g., the ship cabin). This problem could be solved by training on a dataset that has real-world high-resolution images and dense point clouds. Besides, although the motivation of the proposed CSDN is to reconstruct better shapes with the help of single-view images, multi-view images and neural rendering techniques would also help point cloud completion, which is our future work.

6 CONCLUSION

We present a novel point cloud completion paradigm that mimics the physical object-repairing process. The paradigm is implemented as a cross-modal shape-transfer double-refinement network (CSDN). Experiments on standard point cloud completion benchmarks demonstrate the significant improvement of our CSDN over the pioneer multi-modal method ViPC and the other SOTAs. Our work has revealed that the fusion of features from cross-modalities facilitates the generation of reliable geometries to complete partial point clouds. This benefits from the fact that features learned from images not only constrain the generation of the overall shape of an object but also help to refine its local details, indicating the potential of using images to recover finer geometry of objects from partial point clouds.

ACKNOWLEDGEMENTS

REFERENCES

[1] D. Lu, Q. Xie, M. Wei, L. Xu, and J. Li, "Transformers in 3d point clouds: A survey," *CoRR*, vol. abs/2205.07417, 2022.

[2] B. Fei, W. Yang, W. Chen, Z. Li, Y. Li, T. Ma, X. Hu, and L. Ma, "Comprehensive review of deep learning-based 3d point cloud completion processing and analysis," *CoRR*, vol. abs/2203.03311, 2022.

[3] X. Zhang, Y. Feng, S. Li, C. Zou, H. Wan, X. Zhao, Y. Guo, and Y. Gao, "View-guided point cloud completion," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 15 890–15 899.

[4] L. Nan, A. Sharf, and B. Chen, "2d-3d lifting for shape reconstruction," *Comput. Graph. Forum*, vol. 33, no. 7, pp. 249–258, 2014.

[5] Z. Wang, Q. Xie, M. Wei, K. Long, and J. Wang, "Multi-feature fusion votenet for 3d object detection," *ACM Trans. Multim. Comput. Commun. Appl.*, vol. 18, no. 1, pp. 6:1–6:17, 2022.

[6] X. Huang and S. Belongie, "Arbitrary style transfer in real-time with adaptive instance normalization," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 1501–1510.

[7] T. Karras, S. Laine, and T. Aila, "A style-based generator architecture for generative adversarial networks," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019, pp. 4401–4410.

[8] X. Yu, Y. Rao, Z. Wang, Z. Liu, J. Lu, and J. Zhou, "Pointr: Diverse point cloud completion with geometry-aware transformers," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 12 498–12 507.

[9] C. R. Qi, H. Su, K. Mo, and 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.

[10] C. R. Qi, L. Yi, H. Su, and L. J. Guibas, "Pointnet++: Deep hierarchical feature learning on point sets in a metric space," *Advances in neural information processing systems*, vol. 30, 2017.

[11] Y. Li, R. Bu, M. Sun, W. Wu, X. Di, and B. Chen, "Pointcnn: Convolution on x-transformed points," *Advances in neural information processing systems*, vol. 31, 2018.

[12] W. Wu, Z. Qi, and L. Fuxin, "Pointconv: Deep convolutional networks on 3d point clouds," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 9621–9630.

[13] H. Thomas, C. R. Qi, J.-E. Deschaud, B. Marcotegui, F. Goulette, and L. J. Guibas, "Kpconv: Flexible and deformable convolution for point clouds," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2019, pp. 6411–6420.

[14] Y. Wang, Y. Sun, Z. Liu, S. E. Sarma, M. M. Bronstein, and J. M. Solomon, "Dynamic graph cnn for learning on point clouds," *Acm Transactions On Graphics (tog)*, vol. 38, no. 5, pp. 1–12, 2019.

[15] Z.-H. Lin, S.-Y. Huang, and Y.-C. F. Wang, "Convolution in the cloud: Learning deformable kernels in 3d graph convolution networks for point cloud analysis," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 1800–1809.

[16] F. Pistilli, G. Fracastoro, D. Valsesia, and E. Magli, "Learning graph-convolutional representations for point cloud denoising," in *European conference on computer vision*. Springer, 2020, pp. 103–118.

[17] H. Zhou, Y. Feng, M. Fang, M. Wei, J. Qin, and T. Lu, "Adaptive graph convolution for point cloud analysis," in *2021 IEEE/CVF International Conference on Computer Vision*, 2021, pp. 4945–4954.

[18] M. Wei, Z. Wei, H. Zhou, F. Hu, H. Si, Z. Chen, Z. Zhu, J. Qiu, X. Yan,

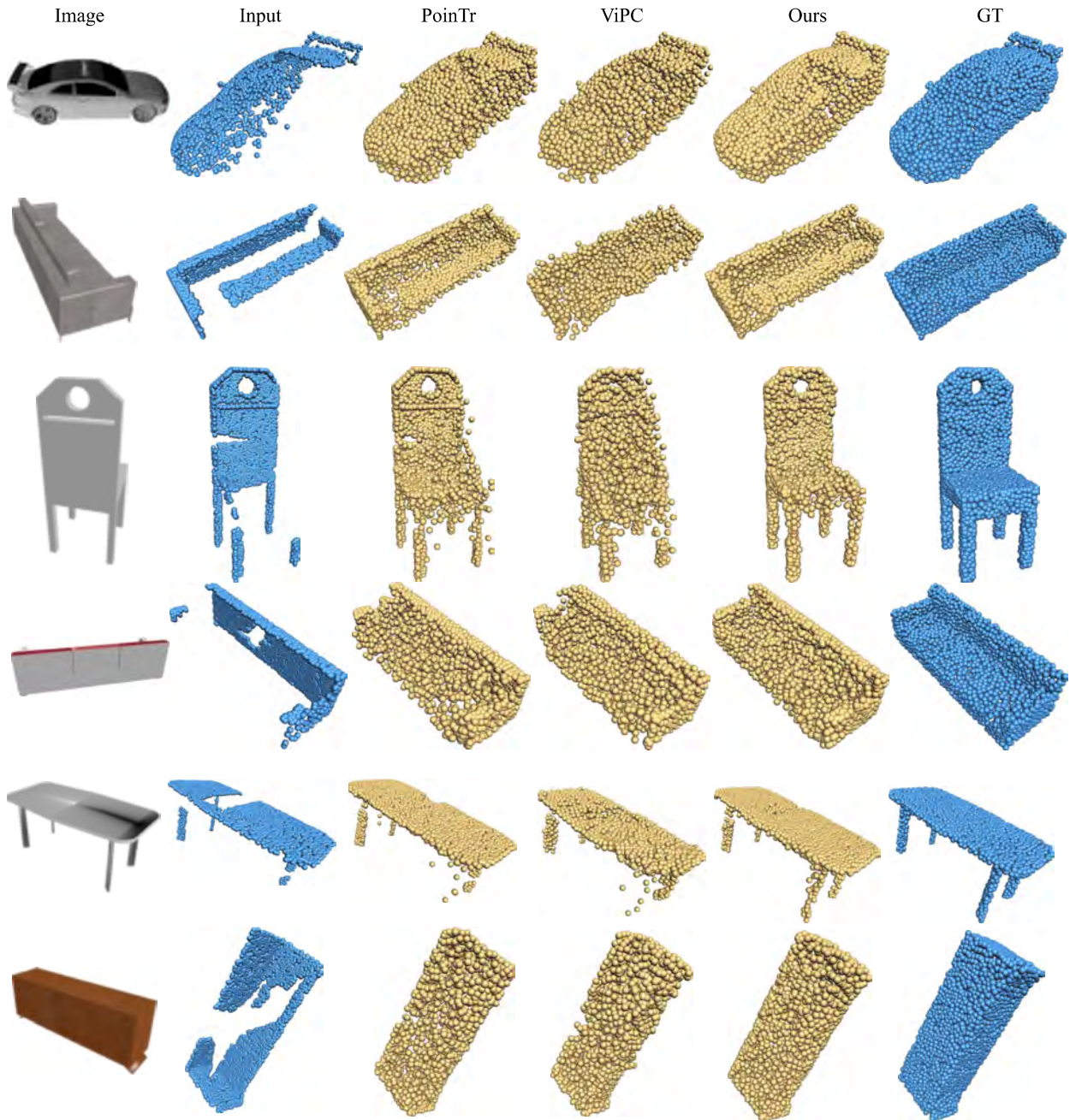Fig. 18. Comparisons between CSDN and its competitors [3], [8] when images and partial point clouds are captured under the same viewpoints.

Y. Guo *et al.*, "Agconv: Adaptive graph convolution on 3d point clouds," *arXiv preprint arXiv:2206.04665*, 2022.

[19] H. Zhao, L. Jiang, J. Jia, P. H. Torr, and V. Koltun, "Point transformer," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 16 259–16 268.

[20] M.-H. Guo, J.-X. Cai, Z.-N. Liu, T.-J. Mu, R. R. Martin, and S.-M. Hu, "Pct: Point cloud transformer," *Computational Visual Media*, vol. 7, no. 2, pp. 187–199, 2021.

[21] P. Achlioptas, O. Diamanti, I. Mitliagkas, and L. Guibas, "Learning representations and generative models for 3d point clouds," in *International conference on machine learning*. PMLR, 2018, pp. 40–49.

[22] D. W. Shu, S. W. Park, and J. Kwon, "3d point cloud generative adversarial network based on tree structured graph convolutions," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2019, pp. 3859–3868.

[23] R. Li, X. Li, K.-H. Hui, and C.-W. Fu, "Sp-gan: Sphere-guided 3d shape generation and manipulation," *ACM Transactions on Graphics (TOG)*, vol. 40, no. 4, pp. 1–12, 2021.

[24] Y. Tang, Y. Qian, Q. Zhang, Y. Zeng, J. Hou, and X. Zhe, "Warpinggan: Warping multiple uniform priors for adversarial 3d point cloud generation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 6397–6405.

[25] G. Yang, X. Huang, Z. Hao, M.-Y. Liu, S. Belongie, and B. Hariharan, "Pointflow: 3d point cloud generation with continuous normalizing flows," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 4541–4550.

[26] R. Cai, G. Yang, H. Averbuch-Elor, Z. Hao, S. Belongie, N. Snavely, and B. Hariharan, "Learning gradient fields for shape generation," in *European Conference on Computer Vision*. Springer, 2020, pp. 364–381.

[27] L. Yi, W. Zhao, H. Wang, M. Sung, and L. J. Guibas, "Gspn: Generative shape proposal network for 3d instance segmentation in point cloud," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 3947–3956.

[28] Y. Sun, Y. Wang, Z. Liu, J. Siegel, and S. Sarma, "Pointgrow: Autoregressively learned point cloud generation with self-attention," in *Proceedings*

of the IEEE/CVF Winter Conference on Applications of Computer Vision, 2020, pp. 61–70.

[29] P. Mittal, Y.-C. Cheng, M. Singh, and S. Tulsiani, "Autosdf: Shape priors for 3d completion, reconstruction and generation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 306–315.

[30] W. Yuan, T. Khot, D. Held, C. Mertz, and M. Hebert, "Pcn: Point completion network," in *2018 International Conference on 3D Vision (3DV)*. IEEE, 2018, pp. 728–737.

[31] L. Yu, X. Li, C.-W. Fu, D. Cohen-Or, and P.-A. Heng, "Pu-net: Point cloud upsampling network," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 2790–2799.

[32] G. Qian, A. Abualshour, G. Li, A. Thabet, and B. Ghanem, "Pu-gcn: Point cloud upsampling using graph convolutional networks," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 11 683–11 692.

[33] R. Li, X. Li, P.-A. Heng, and C.-W. Fu, "Point cloud upsampling via disentangled refinement," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 344–353.

[34] M. Berger, A. Tagliasacchi, L. Seversky, P. Alliez, J. Levine, A. Sharf, and C. Silva, "State of the art in surface reconstruction from point clouds," in *Eurographics 2014-State of the Art Reports*, vol. 1, no. 1, 2014, pp. 161–185.

[35] J. Davis, S. R. Marschner, M. Garr, and M. Levoy, "Filling holes in complex surfaces using volumetric diffusion," in *Proceedings. First International Symposium on 3D Data Processing Visualization and Transmission*. IEEE, 2002, pp. 428–441.

[36] A. Nealen, T. Igarashi, O. Sorkine, and M. Alexa, "Laplacian mesh optimization," in *Proceedings of the 4th international conference on Computer graphics and interactive techniques in Australasia and Southeast Asia*, 2006, pp. 381–389.

[37] N. J. Mitra, L. J. Guibas, and M. Pauly, "Partial and approximate symmetry detection for 3d geometry," *ACM Transactions on Graphics (TOG)*, vol. 25, no. 3, pp. 560–568, 2006.

[38] N. J. Mitra, M. Pauly, M. Wand, and D. Ceylan, "Symmetry in 3d geometry: Extraction and applications," in *Computer Graphics Forum*, vol. 32, no. 6. Wiley Online Library, 2013, pp. 1–23.

[39] M. Sung, V. G. Kim, R. Angst, and L. Guibas, "Data-driven structural priors for shape completion," *ACM Transactions on Graphics (TOG)*, vol. 34, no. 6, pp. 1–11, 2015.

[40] S. Thrun and B. Wegbreit, "Shape from symmetry," in *Tenth IEEE International Conference on Computer Vision (ICCV'05) Volume 1*, vol. 2. IEEE, 2005, pp. 1824–1831.

[41] V. Blanz and T. Vetter, "A morphable model for the synthesis of 3d faces," in *Proceedings of the 26th annual conference on Computer graphics and interactive techniques*, 1999, pp. 187–194.

[42] A.-L. Chauve, P. Labatut, and J.-P. Pons, "Robust piecewise-planar 3d reconstruction and completion from large-scale unstructured point data," in *2010 IEEE computer society conference on computer vision and pattern recognition*. IEEE, 2010, pp. 1261–1268.

[43] F. Han and S.-C. Zhu, "Bottom-up/top-down image parsing with attribute grammar," *IEEE transactions on pattern analysis and machine intelligence*, vol. 31, no. 1, pp. 59–73, 2008.

[44] E. Kalogerakis, S. Chaudhuri, D. Koller, and V. Koltun, "A probabilistic model for component-based shape synthesis," *Acm Transactions on Graphics (TOG)*, vol. 31, no. 4, pp. 1–11, 2012.

[45] A. Dai, C. Ruizhongtai Qi, and M. Nießner, "Shape completion using 3d-encoder-predictor cnns and shape synthesis," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 5868–5877.

[46] X. Han, Z. Li, H. Huang, E. Kalogerakis, and Y. Yu, "High-resolution shape completion using deep neural networks for global structure and local geometry inference," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 85–93.

[47] H. Xie, H. Yao, S. Zhou, J. Mao, S. Zhang, and W. Sun, "Grnet: Gridding residual network for dense point cloud completion," in *European Conference on Computer Vision*. Springer, 2020, pp. 365–381.

[48] W. Zhang, Q. Yan, and C. Xiao, "Detail preserved point cloud completion via separated feature aggregation," in *European Conference on Computer Vision*. Springer, 2020, pp. 512–528.

[49] L. P. Tchapmi, V. Kosaraju, H. Rezatofighi, I. Reid, and S. Savarese, "Topnet: Structural point cloud decoder," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 383–392.

[50] M. Liu, L. Sheng, S. Yang, J. Shao, and S.-M. Hu, "Morphing and sampling network for dense point cloud completion," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 34, no. 07, 2020, pp. 11 596–11 603.

[51] Z. Huang, Y. Yu, J. Xu, F. Ni, and 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.

[52] X. Wang, M. H. Ang Jr, and G. H. Lee, "Cascaded refinement network for point cloud completion," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 790–799.

[53] M. Sarmad, H. J. Lee, and Y. M. Kim, "Rl-gan-net: A reinforcement learning agent controlled gan network for real-time point cloud shape completion," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 5898–5907.

[54] X. Wen, P. Xiang, Z. Han, Y.-P. Cao, P. Wan, W. Zheng, and Y.-S. Liu, "Pmp-net: Point cloud completion by learning multi-step point moving paths," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 7443–7452.

[55] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," *Advances in neural information processing systems*, vol. 30, 2017.

[56] P. Xiang, X. Wen, Y.-S. Liu, Y.-P. Cao, P. Wan, W. Zheng, and Z. Han, "Snowflakenet: Point cloud completion by snowflake point deconvolution with skip-transformer," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 5499–5509.

[57] D. Stutz and A. Geiger, "Learning 3d shape completion from laser scan data with weak supervision," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 1955–1964.

[58] X. Chen, B. Chen, and N. J. Mitra, "Unpaired point cloud completion on real scans using adversarial training," *arXiv preprint arXiv:1904.00069*, 2019.

[59] X. Wen, Z. Han, Y.-P. Cao, P. Wan, W. Zheng, and Y.-S. Liu, "Cycle4completion: Unpaired point cloud completion using cycle transformation with missing region coding," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 13 080–13 089.

[60] H. Fan, H. Su, and L. J. Guibas, "A point set generation network for 3d object reconstruction from a single image," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 605–613.

[61] T. Groueix, M. Fisher, V. G. Kim, B. C. Russell, and M. Aubry, "A papier-mâché approach to learning 3d surface generation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 216–224.

[62] L. Jiang, S. Shi, X. Qi, and J. Jia, "Gal: Geometric adversarial loss for single-view 3d-object reconstruction," in *Proceedings of the European conference on computer vision (ECCV)*, 2018, pp. 802–816.

[63] C. B. Choy, D. Xu, J. Gwak, K. Chen, and S. Savarese, "3d-r2n2: A unified approach for single and multi-view 3d object reconstruction," in *European conference on computer vision*. Springer, 2016, pp. 628–644.

[64] H. Xie, H. Yao, X. Sun, S. Zhou, and S. Zhang, "Pix2vox: Context-aware 3d reconstruction from single and multi-view images," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2019, pp. 2690–2698.

[65] H. Xie, H. Yao, S. Zhang, S. Zhou, and W. Sun, "Pix2vox++: Multiscale context-aware 3d object reconstruction from single and multiple images," *International Journal of Computer Vision*, vol. 128, no. 12, pp. 2919–2935, 2020.

[66] N. Wang, Y. Zhang, Z. Li, Y. Fu, W. Liu, and Y.-G. Jiang, "Pixel2mesh: Generating 3d mesh models from single rgb images," in *Proceedings of the European conference on computer vision (ECCV)*, 2018, pp. 52–67.

[67] C. Wen, Y. Zhang, Z. Li, and Y. Fu, "Pixel2mesh++: Multi-view 3d mesh generation via deformation," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2019, pp. 1042–1051.

[68] J. Pan, X. Han, W. Chen, J. Tang, and K. Jia, "Deep mesh reconstruction from single rgb images via topology modification networks," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 9964–9973.

[69] X. Li, S. Liu, K. Kim, S. D. Mello, V. Jampani, M.-H. Yang, and J. Kautz, "Self-supervised single-view 3d reconstruction via semantic consistency," in *European Conference on Computer Vision*. Springer, 2020, pp. 677–693.

[70] J. J. Park, P. Florence, J. Straub, R. Newcombe, and S. Lovegrove, "Deepsdf: Learning continuous signed distance functions for shape representation," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019, pp. 165–174.

[71] L. Mescheder, M. Oechsle, M. Niemeyer, S. Nowozin, and A. Geiger, "Occupancy networks: Learning 3d reconstruction in function space," in

Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, 2019, pp. 4460–4470.

[72] B. Mildenhall, P. P. Srinivasan, M. Tancik, J. T. Barron, R. Ramamoorthi, and R. Ng, "Nerf: Representing scenes as neural radiance fields for view synthesis," *Communications of the ACM*, vol. 65, no. 1, pp. 99–106, 2021.

[73] C.-H. Lin, C. Wang, and S. Lucey, "Sdf-srn: Learning signed distance 3d object reconstruction from static images," *Advances in Neural Information Processing Systems*, vol. 33, pp. 11 453–11 464, 2020.

[74] S. Duggal and D. Pathak, "Topologically-aware deformation fields for single-view 3d reconstruction," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 1536–1546.

[75] Y.-C. Chen, H. Li, D. Turpin, A. Jacobson, and A. Garg, "Neural shape mating: Self-supervised object assembly with adversarial shape priors," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 12 724–12 733.

[76] W. Zhao, X. Liu, Z. Zhong, J. Jiang, W. Gao, G. Li, and X. Ji, "Self-supervised arbitrary-scale point clouds upsampling via implicit neural representation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 1999–2007.

[77] C. R. Qi, X. Chen, O. Litany, and L. J. Guibas, "Imvotenet: Boosting 3d object detection in point clouds with image votes," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 4404–4413.

[78] J. Li and G. H. Lee, "Deepi2p: Image-to-point cloud registration via deep classification," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 15 960–15 969.

[79] C. Xu, S. Yang, B. Zhai, B. Wu, X. Yue, W. Zhan, P. Vajda, K. Keutzer, and M. Tomizuka, "Image2point: 3d point-cloud understanding with pretrained 2d convnets," *arXiv preprint arXiv:2106.04180*, 2021.

[80] Y. Yang, C. Feng, Y. Shen, and D. Tian, "Foldingnet: Point cloud auto-encoder via deep grid deformation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 206–215.

[81] C. Xie, C. Wang, B. Zhang, H. Yang, D. Chen, and F. Wen, "Style-based point generator with adversarial rendering for point cloud completion," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 4619–4628.

[82] J. Wang, Y. Cui, D. Guo, J. Li, Q. Liu, and C. Shen, "Pointattn: You only need attention for point cloud completion," *arXiv preprint arXiv:2203.08485*, 2022.

[83] W. Zhang, Z. Dong, J. Liu, Q. Yan, C. Xiao *et al.*, "Point cloud completion via skeleton-detail transformer," *IEEE Transactions on Visualization and Computer Graphics*, 2022.

[84] H. Zhou, Y. Cao, W. Chu, J. Zhu, T. Lu, Y. Tai, and C. Wang, "Seedformer: Patch seeds based point cloud completion with upsample transformer," *arXiv preprint arXiv:2207.10315*, 2022.

[85] A. X. Chang, T. Funkhouser, L. Guibas, P. Hanrahan, Q. Huang, Z. Li, S. Savarese, M. Savva, S. Song, H. Su *et al.*, "Shapenet: An information-rich 3d model repository," *arXiv preprint arXiv:1512.03012*, 2015.

[86] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun, "Vision meets robotics: The kitti dataset," *The International Journal of Robotics Research*, vol. 32, no. 11, pp. 1231–1237, 2013.

[87] A. Dai, A. X. Chang, M. Savva, M. Halber, T. Funkhouser, and M. Nießner, "Scannet: Richly-annotated 3d reconstructions of indoor scenes," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 5828–5839.

[88] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[89] L. Van der Maaten and G. Hinton, "Visualizing data using t-sne." *Journal of machine learning research*, vol. 9, no. 11, 2008.

Liangliang Nan received the Ph.D. degree in mechatronics engineering from the Graduate University of the Chinese Academy of Sciences, Beijing, China, in 2009. He worked at the Visual Computing Center, King Abdullah University of Science and Technology (KAUST), as a Research Scientist. He is currently an Assistant Professor with the Delft University of Technology (TU Delft), where he is leading the AI Laboratory on 3D Urban Understanding (3DUU). His research interests include computer graphics, computer vision, 3D geoinformation, and machine learning.

Haoran Xie (Senior Member, IEEE) received the Ph.D. degree in Computer Science from City University of Hong Kong. He is currently an Associate Professor at the Department of Computing and Decision Sciences, Lingnan University. His research interests include artificial intelligence, big data, and educational technology. He has published 300 research publications, including 159 journal articles such as IEEE TPAMI, IEEE TKDE, IEEE TAFFC, IEEE TCVST. He is the Editor-in-Chief of Natural Language Processing Journal, Computers & Education: Artificial Intelligence and Computers & Education: X Reality. He has been selected as The World Top 2% Scientists by Stanford University.

Honghua Chen is a Ph.D. candidate at Nanjing University of Aeronautics and Astronautics (NUAA), China. He earned his master's degree from Nanjing Normal University in 2017. His research interest is smart geometry processing.
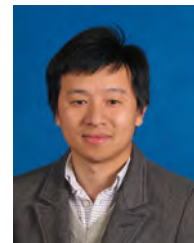
Jun Wang is currently a Professor at Nanjing University of Aeronautics and Astronautics (NUAA), China. He received his Ph.D. degrees in Computer-Aided Design from NUAA in 2007. He paid an academic visit to the Department of Mathematics at Harvard University. His research interests include geometry processing and geometric modeling, especially large-scale LiDAR point data capturing, management, processing and analysis.

Mingqiang Wei received his Ph.D degree (2014) in Computer Science and Engineering from the Chinese University of Hong Kong (CUHK). He is a Professor at the School of Computer Science and Technology, Nanjing University of Aeronautics and Astronautics (NUAA). He was a recipient of the CUHK Young Scholar Thesis Awards in 2014. He is now an Associate Editor for ACM TOMM, the Visual Computer Journal, and a Guest Editor for IEEE Transactions on Multimedia. His research interests focus on 3D vision, computer graphics, and deep learning.

Jing Qin is currently an associate professor in School of Nursing, The Hong Kong Polytechnic University and is a key member in the Centre for Smart Health. His research focuses on creatively leveraging advanced virtual reality (VR) and artificial intelligence (AI) techniques in healthcare and medicine applications. He won the Hong Kong Medical and Health Device Industries Association Student Research Award for his PhD study on VR-based simulation systems for surgical training and planning. He won 3 best paper awards for his research on AI-driven medical image analysis and computer-assisted surgery, including one of the most prestigious awards in this field: MIA-MICCAI best paper award in 2017.

Zhe Zhu is a Ph.D candidate at Nanjing University of Aeronautics and Astronautics (NUAA), China. He received his B.Sc. degree from China University of Mining and Technology. His research interests include 3D vision and learning-based geometry processing.