Point Cloud Completion via Self-Projected View Augmentation and Implicit Field Constraint

Haihong Xiao, Ying He, *Member, IEEE*, Hao Liu, Wenxiong Kang, *Member, IEEE*, and Yuqiong Li

***Abstract*— Recent advances in point cloud completion make it possible to simultaneously recover complete shapes and fine details from partial point clouds captured by professional 3D devices, such as Lidar, or consumer cameras, such as iPhones. Despite significant progress, the potential utilization of self-projected views from partial inputs and the effective reduction of noise in generated point clouds remain under-explored. In this paper, we propose a novel point cloud completion method that leverages self-projected view augmentation and implicit field constraints. Specifically, we introduce a cross-view augmentation (CVA) module and a cross-modal fusion (CMF) module to enhance information interaction and integration at the image and modality levels, respectively. We also propose a bidirection-aware refinement block to improve detail and completeness by considering both complete-to-partial detail perception and partial-to-complete structure perception paths. Additionally, we address the issue of noise reduction from the perspective of implicit field constraints. We evaluate our method on several baseline datasets, including PCN, ShapeNet55/34 and KITTI (car). Extensive experiments demonstrate that our method outperforms state-of-the-art methods, achieving improvements of 0.11 CD-ℓ_1, 0.015 DCD and 0.009 F-score on the standard PCN test set. Furthermore, our approach effectively reduces noise in the generated point clouds, showcasing its promising potential for practical applications.**

***Index Terms*— 3D point cloud, point cloud completion, 3D vision.**

I. Introduction

POINT cloud completion, which is the task of predicting the completed, detailed and noise-free shape of objects from partial input point clouds, has emerged as a hot research topic in 3D vision and graphics. Humans can effortlessly imagine the complete 3D geometry of occluded objects, but how to endow machines with this extraordinary capability is vital for many cutting-edge applications, including autonomous driving [1], virtual/augmented reality [2] and industrial design [3].

Thanks to the rapid development of deep learning and computing hardware, learning-based point cloud completion methods [4], [5], [6], [7], [8], [9], [10], [11], [12], [13], [14], [15], [16], [17], [18], [19], [20], [21], [22], [23], [24], [25], [26], [27], [28], [29], [30] have shown explosive growth. Motivated by the different mechanisms through which humans perceive 3D world and 2D images, these methods can broadly be categorized into three types: point-based [4], [5], [6], [7], [8], [9], [10], [11], [12], [13], [14], [15], [16], [17], [18], [19], [20], [21], [22], voxelization-based [23], [24], [25], [26] and view-guided [27], [28], [29], [30] point cloud completion approaches.

Point-based point cloud completion techniques, building upon the enlightening framework proposed by PCN [4], achieve significant performance improvements through enhancements in feature extraction [5], [6], [7], [8], point cloud refinement [9], [10], [11], [12], rendering optimization [13], [14] and loss function modification [15], [16]. Voxelization-based point cloud completion approaches are primarily inspired by the success of Convolutional Neural Networks (CNNs) [31] on regular data, prompting researchers to transform irregular point clouds into regular grids for processing. Essentially, the success of deep learning in the 2D vision domain can be attributed to the strong priors to data structured as tensors. View-guided point cloud completion methods enhance the understanding of the object's complete structure by incorporating corresponding 2D images and camera parameters as extra information. This is inspired by how humans utilize the multimodal information to make more comprehensive and intelligent decisions.

Although the aforementioned methods have each contributed to the development of the point cloud completion community from various perspectives, we note that there are still intractable issues impeding the progress of this field.

1) View-guided point cloud completion methods, which need additional RGB images and corresponding camera parameters, may not always be feasible in certain scenarios. Thus, a reasonable hypothesis explores the potential of using self-projected views of partial point clouds to enhance the point cloud completion task, which is orthogonal to the predecessor SVDFormer [30].

Manuscript received 26 March 2024; revised 9 June 2024; accepted 3 July 2024. Date of publication 8 July 2024; date of current version 27 November 2024. This work was supported in part by the National Natural Science Foundation of China under Grant 62376100 and in part by the Natural Science Foundation of Guangdong Province of China under Grant 2022A1515010114. This article was recommended by Associate Editor X. Guo. *(Corresponding author: Wenxiong Kang.)*

Haihong Xiao is with the School of Automation Science and Engineering, South China University of Technology, Guangzhou 510641, China (e-mail: auhhxiao@mail.scut.edu.cn).

Ying He and Hao Liu are with the College of Computing and Data Science, Nanyang Technological University, Singapore 639798 (e-mail: yhe@ntu.edu.sg; hao.liu@ntu.edu.sg).

Wenxiong Kang is with the School of Automation Science and Engineering, South China University of Technology, Guangzhou 510641, China, also with the School of Future Technology, South China University of Technology, Guangzhou 510641, China, and also with the Pazhou Laboratory, Guangzhou 510335, China (e-mail: auwxkang@scut.edu.cn).

Yuqiong Li is with the Key Laboratory for Mechanics in Fluid Solid Coupling Systems, Institute of Mechanics, Chinese Academy of Sciences, Beijing 100190, China (e-mail: liyuqiong@imech.ac.cn).

Color versions of one or more figures in this article are available at https://doi.org/10.1109/TCSVT.2024.3424776.

Digital Object Identifier 10.1109/TCSVT.2024.3424776

TABLE I

COMPARISON AMONG SOME REPRESENTATIVE METHODS. "PROJECTION-ASSISTED" REFERS TO ENHANCING POINT CLOUD COMPLETION TASKS THROUGH THE USE OF SELF-PROJECTED VIEWS. "COMPLETION TYPE" REFERS TO THE DIFFERENT STRATEGIES FOR THE COMPLETION TASK, TYPICALLY DIVIDED INTO TWO MAIN CATEGORIES: PREDICTING MISSING PARTS AND PREDICTING COMPLETE STRUCTURES

Method	projection-assisted	detail-aware	noise-aware	completion type	principal operator
TopNet [21]	✗	✗	✗	complete structure	MLP
PF-Net [22]	✗	✗	✗	missing part	MLP
GRNet [23]	✗	✗	✗	complete structure	3DCNN
PoinTr [17]	✗	✓	✗	missing part	Transformer
PDR [19]	✗	✓	✓	complete structure	Diffusion
SeedFormer [32]	✗	✓	✗	complete structure	Transformer
SVDFormer [30]	✓	✓	✗	complete structure	Transformer
Ours	✓	✓	✓	complete structure	Transformer

However, SVDFormer is plagued by the indistinct connections between self-projected views, necessitating further in-depth analysis within this track.

2) Existing point cloud completion efforts largely adhere to a coarse-to-fine fashion, wherein the generated coarse-grained complete point clouds are fused with the input point clouds for secondary coordinate or offset prediction [5], [32]. However, they have not fully explored the interaction of structural information in the coarse-grained complete point clouds and the local detail information in the partial point clouds.

3) The goal of point cloud completion is to predict complete and fine-grained point clouds. Simultaneously, we aspire for the generated point clouds to be free of noise, making them readily applicable in downstream applications, such as virtual/augmented reality and industrial design. Regrettably, most existing works still lead to a small amount of noise. We argue that this issue is primarily caused by two main reasons: *1) the absence of effective noise reduction strategies, 2) the presence of noise in even clean point clouds sampled from synthetic datasets, resulting in inaccurate supervisory signals,*[1] *as illustrated by the purple dashed box in Fig. 5.*

In this paper, we introduce a novel point cloud completion method, which builds upon view-guided techniques, enhanced with self-projected views and implicit field constraints. In Table. I, we conduct a systematic comparison of our method with several benchmark methods. Our method tackles these challenges mentioned above with the following solutions:

1) We propose the utilization of self-projected views to assist in the point cloud completion task. Experimental results demonstrate that even with incomplete input point clouds, their self-projected views can still contribute to the understanding of point cloud shape, which aligns with the findings of the recent SVDFormer. Unlike SVDFormer, our self-projected view augmentation strategy incorporates cross-view augmentation (CVA) and cross-modal fusion (CMF) modules, which promote information interaction and integration at the image and modality levels, respectively.

2) To further refine point clouds and fully exploit the complete structural information of coarse-grained point clouds along with the detail information of partial inputs, we introduce a bidirection-aware refinement block. This block improves the detail and completeness from both the complete to partial detail perception and partial to complete structure perception paths.

3) Unlike existing point cloud completion methods that utilize additional normal constraints [33] or design more complex loss functions [15], [16] for supervision, we propose reducing noise through self-learned implicit field constraints. Specifically, our method directly learns the underlying implicit surfaces from point clouds, effectively drawing outliers towards the approximate surface. This scheme, motivated by the ability of the learned implicit field function, could accurately capture the signed distance function (SDF) of the point cloud. Therefore, we can draw any point to the surface by predicting its SDF value and corresponding gradient.

To summarize, our contributions are four-fold as below.

- We propose to utilize self-projected views to augment point cloud completion tasks. Specifically, we propose the cross-view augmentation (CVA) and cross-modal fusion (CMF) modules to promote information interaction and integration at the image and modality levels, respectively.
- We introduce a bidirection-aware refinement block to improve the detail and completeness by considering both complete-to-partial detail perception and partial-to-complete structure perception paths.
- We make a new attempt to reduce noise in completed point clouds through self-learned implicit field constraints.
- We conduct comprehensive comparisons on the PCN, ShapeNet55/34 and KITTI (car) datasets to evaluate the effectiveness of our proposed method. We will open source all code to facilitate the research in this field.

II. RELATED WORK

A. Traditional Shape Completion

Traditional methods for shape completion are primarily categorized into three types: symmetry-based [34], [35], surface reconstruction-based [36], [37], [38], [39] and template alignment-based [40], [41] approaches.

[1]The data is sourced from the PCN dataset, including the object 92a4a6cf717d042ee194052f3f12cb2e under the car category (02958343) and the object 79a3bd60b48584b11ea954af295a6a98 under the table category (04379243).

Symmetry-based methods [34], [35] leverage the geometric symmetry in objects or spaces to reconstruct structures in missing regions. These methods rely heavily on the assumption of mirror symmetry, positing that unobserved geometric parts are exact reflections of observed sections, which is particularly effective for objects exhibiting simple bilateral symmetry. However, this symmetry assumption is not universally valid for all natural objects. Surface reconstruction-based methods [36], [37], [38], [39] typically fall into one of two categories: interpolation [36], [37] and fitting [38], [39]. Interpolation techniques, starting with surface points and employing various algorithms, generate dense surfaces, such as B-spline [36] and fractal interpolation [37]. Fitting techniques like Poisson surface reconstruction [38] utilize sampled point clouds for the accurate reconstruction of approximate surfaces. While effective across various scenarios, these methods are particularly suited for repairing small-area holes. Template alignment-based methods [40], [41] cover both partial and entire shape alignment techniques. The partial shape alignment technique [40] involves matching and assembling the most appropriate components from a pre-defined, extensive shape model library to get a complete object. The entire shape alignment technique [41] focuses on directly retrieving the best-fit complete shape from the model library. It is worth mentioning that how best to choose the optimization algorithm plays a crucial role. Furthermore, these methods depend on large model libraries to cover all shapes for completion, which is often impractical in real-world scenarios.

B. Learning-Based Point Cloud Completion

Benefiting from the rapid advancement of deep learning and the advantages of point clouds in terms of small storage yet strong representational capacity [42], learning-based point cloud completion methods [4], [5], [6], [7], [8], [9], [10], [11], [12], [13], [14], [15], [16], [17], [18], [19], [20], [23], [24], [25], [26], [27], [28], [29], [30], [43], [44], [45] have achieved significant progress. In the following, we will briefly revisit learning-based point cloud completion studies.

1) Point-Based Point Cloud Completion: Point-based point cloud completion methods aim to reconstruct complete shapes from partial point clouds. As a seminal work, PCN [4] utilizes PointNet [42] for feature extraction and FoldNet's [46] folding decoder for point cloud generation, complemented by heuristic loss functions such as the Chamfer Distance (CD) [47] and Earth Mover's Distance (EMD) [48] for effective supervision. This comprehensive supervised paradigm set by PCN has spurred a surge in research, particularly in efficient feature extraction for point clouds [5], [6], [7], [8], multi-stage generation [9], [10], [11], [12], [43], rendering optimization [13], [14] and loss function modification [15], [16]. Additionally, the remarkable success of Transformers [49] in Natural Language Processing (NLP) and image processing has further advanced the field of 3D vision, notably in point cloud recognition [50], [51], segmentation [52], [53] and completion [17], [18]. A notable development in this context is PointTr [17], which reformulates point cloud completion as a set-to-set transformation problem, proposing a Transformer-based encoder-decoder framework. Fueled by this, Fu et al. [54] proposed VAPCNet, a viewpoint-aware point cloud completion method. Unlike method [55], VAPCNet does not require ground truth viewpoint values. Specifically, it extracts viewpoint information implicitly through contrastive learning, providing a new perspective for point cloud completion tasks. Liu et al. [56] proposed CloudMix, an unpaired point cloud completion method that adopts source-target domain integration via dual mixup consistency to improve generalization capability from virtual to real-world domains. The recent advancements in diffusion models [57], recognized as potent generative tools, have demonstrated outstanding outcomes in both image [58], [59] and point cloud generation [60], [61]. This has led to a novel perspective where point cloud completion is defined as a generative process using conditional diffusion models [19], [20], thereby enriching the point cloud completion community. Despite the ideal representation of point clouds as spatial particles in diffusion models, one of the major limitations of diffusion models is the requirement of thousands of sequential steps to obtain high-quality completion samples. To address this bottleneck, some researchers seek to leverage acceleration strategies [62], [63] or compress the point cloud into a latent space [64] before applying diffusion models. However, these adaptations often involve a trade-off, potentially sacrificing the quality of generated outputs.

2) Voxelization-Based Point Cloud Completion: Despite years of effort in feature extraction for point clouds [50], [65], [66], [67], [68], a universally used paradigm for handling the inherent disorder and irregularity of point clouds, analogous to the role of Convolutional Neural Networks (CNNs) in 2D vision, remains elusive. Indeed, the success of deep learning in 2D vision is largely attributed to convolutional networks that deploy strong priors suited to data structured as tensors. Consequently, there has been a growing interest among researchers in exploring the voxelization of point clouds [23], [24], [25], [26] to make them suitable for regularized convolution processing in completion tasks. A noteworthy advancement is GRNet [23], which converts unordered point clouds into structured grids for intermediate representation, then employs 3D CNNs for both feature extraction and intermediate data synthesis before reverting these grid structures back to point cloud format. Similarly, Wang et al. [24] innovated the multi-scale point cloud completion network VE-PCN, which integrates edge structural information into the shape completion process. Despite the impressive progress, the escalating computational demands of 3D CNNs, which increase cubically with resolution, pose considerable challenges and necessitate substantial computational resources. To address this challenge, recent research has focused on employing the sparse convolution [69], spatial group convolution [70] and Minkowski convolution [71] to reduce the parameter load of 3D convolutions. However, the demanding computational resources needed remain a significant obstacle. Moreover, voxelization, despite its structural advantages, incurs the fine-grained information loss, thus limiting the generation of high detailed structures.

3) View-Guided Point Cloud Completion: Departing from previous methods that directly infer complete point clouds

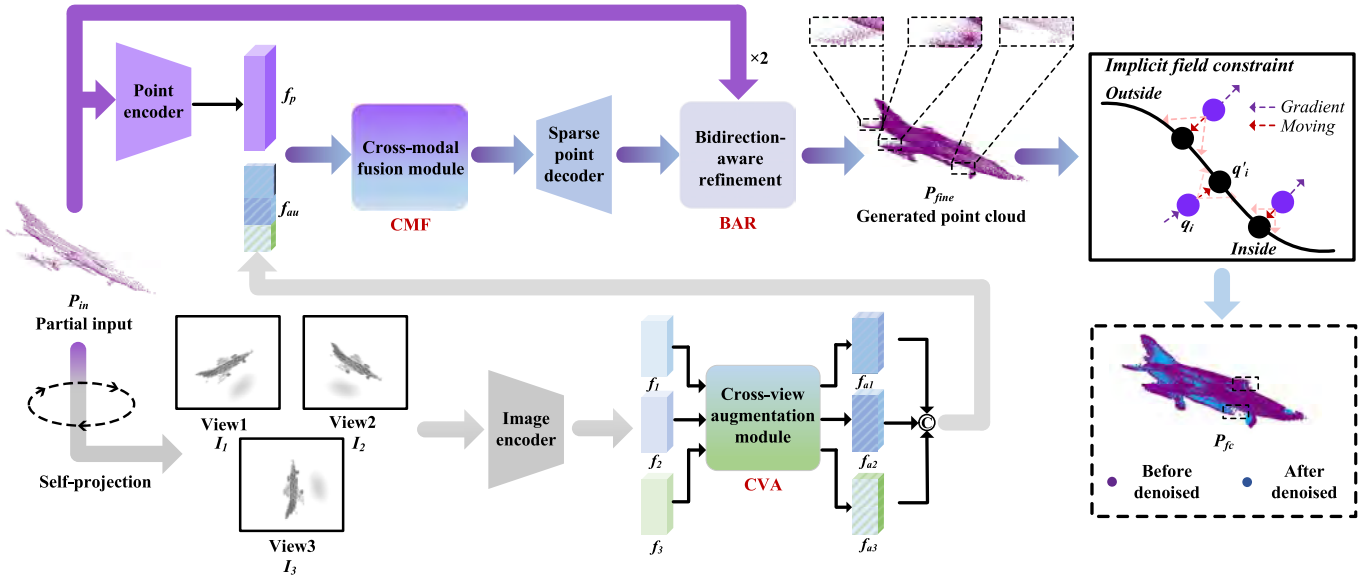

Fig. 1. Overall of our framework. Given a partial input P_{in}, our aim is to predict a completed, detailed and noise-free shape P_{fc}. Our method consists of two stages. In Stage I, we enhance the point cloud completion task using self-projected views. Specifically, we first employ cross-view augmentation (CVA) and cross-modal fusion (CMF) modules to promote information interaction and integration at the image and modality levels, respectively. Then, we introduce a bidirection-aware refinement (BAR) block to improve the detail and completeness of the coarse-grained point cloud and obtain the fine-grained point cloud P_{fine}. In Stage II, we focus on reducing noise within the generated point cloud P_{fine} through self-learned implicit field constraints, obtaining the constrained and noise-free point cloud P_{fc}.

from partial ones, a segment of the research community seeks to improve the point cloud completion performance by introducing additional RGB images [27], [28], [29]. For instance, Zhang et al. [27] introduces the ViPC network, which leverages global structural information from an extra single-view image to assist the point cloud completion task. Furthermore, they also proposed a refinement module called Dynamic Offset Predictor (DOP) to refine the points in a coarse point cloud. Fueled by this, Aiello et al. [28] proposed a weakly-supervised scheme based on differentiable rendering to ensure visual consistency and further improve point cloud completion quality. In parallel, Zhu et al. [29] further manifest the effectiveness of their proposed disentangled feature fusion strategy. However, these approaches heavily rely on cross-modal alignment, which may not always be perfectly feasible, thereby limiting practical applicability. To address this, SVDFormer [30] proposed a self-view rendering enhancement strategy, effectively easing the constraints between image and point cloud camera parameters. While this work marks significant strides in the field, they do not fully explore the synergies among these views. Contrasting with the approach of SVDFormer, our work delves deeper into exploring the synergistic effects among different views and further enhances point cloud completion performance from a bidirectional interaction perspective. Moreover, in this paper, we advocate addressing the noise issue from the perspective of implicit surface constraints, providing a novel solution to high-quality point cloud completion.

III. Method

Fig. 1 provides an overview of our pipeline, which follows a coarse-to-fine fashion. Given an incomplete point cloud P_{in}, our aim is to predict a complete and noise-free shape P_{fc}. Our method has three core points: 1) Utilizing existing 2D pre-trained models, we extract self-projected multi-view features from the partial input P_{in}, providing additional priors for predicting a coarse-grained complete point cloud P_{coarse}. 2) Considering the global structure of P_{coarse} and the fine details of P_{in}, we introduce a bidirectional perception refinement module to achieve a fine-grained point cloud P_{fine}. 3) Unlike existing methods that use extra normal constraints or sophisticated point-level loss functions to alleviate the noise problem, we focus on reducing outliers in point clouds through constraints derived from a self-learned implicit surface field.

Specifically, we initially design a cross-view augmentation module and a cross-modal fusion module to promote **information interaction and integration at the image and modality levels, respectively**. This helps in predicting a coarse-grained complete point cloud P_{coarse}. Subsequently, a bidirection-aware refinement block is employed to improve the detail and completeness of P_{coarse} from both **the complete to partial detail perception and partial to complete structure perception paths**. Lastly, we optimize outliers from the perspective of implicit field constraints, **guiding them towards the learned implicit surface** to further improve the completion quality.

A. Point Cloud Completion via Self-Projected View

Our motivation primarily stems from two sides: 1) Incomplete point cloud inputs can be enhanced through their projected view features, which is in line with the recent work [30]. Different from the approach [30] which fuses different view features via the simple concatenation operation, we enhance the features of the original views by explicitly constructing interactions between different views. 2) Considering

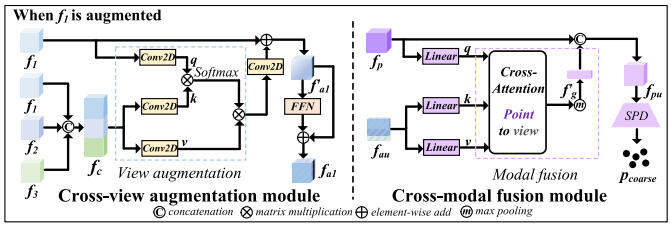

Fig. 2. Illustration of the cross-view augmentation (CVA) module and the cross-modal fusion (CMF) module. Best viewed in colors.

the inherent differences across different modalities, we recognize that simple concatenation of multi-modal data [27], [28] might lead to sub-optimal results. To address this issue, we employ a cross-attention fusion module, which enables a more effective integration of this multi-modal information. Subsequently, a sparse point decoder is utilized to generate a complete coarse-grained point cloud. In the following, we provide a detailed introduction to this part of our work. *To simplify the notation, we omit the dimensional representations associated with the symbols.*

To improve training efficiency, we first project the input incomplete point cloud P_{in} from three orthogonal perspectives respectively along with the x, y and z axes, following the method described in [72]. Each point is processed by discarding one of its three coordinates. The remaining two coordinates are used to determine their two-dimensional position on the corresponding map. The discarded coordinate value serves as the projected pixel value to represent the point's relative depth, which is repeated by three times to imitate the three-channel RGB format. Then, the depth maps projected from the incomplete point cloud are denoted as $\{I_i\}_{i=1}^{3}$. Subsequently, we employ the pretrained ResNet-18 model to extract features from the various depth maps, resulting in $\{f_i\}_{i=1}^{3}$. To explicitly construct interactions between these depth maps, we propose a cross-view augmentation module that augments each view feature, as illustrated in Fig. 2. For conciseness and clarity, we take the augmentation of f_1 as an example. First, we concatenate the features from different views to create a composite view feature, denoted as f_c. We then establish a cross-attention mechanism between the view feature f_1 and the composite view feature f_c, resulting in the enhanced view feature f_{a1}. Last, we concatenate these individually enhanced view features to form an enhanced comprehensive view feature, referred to as f_{au}.

$$\mathbf{q}=\text{Conv2D}\left(f_1\right),\mathbf{k}=\text{Conv2D}\left(f_c\right),\mathbf{v}=\text{Conv2D}\left(f_c\right), \quad (1)$$

$$f'_{a1}=f_1+\text{Conv2D}\left(\left(\mathbf{v}^{\top}\cdot\text{softmax}\left(\mathbf{k}^{\top}\cdot\mathbf{q}\right)\right)^{\top}\right), \quad (2)$$

$$f_{a1}=f'_{a1}+\text{FFN}\left(f'_{a1}\right), \quad (3)$$

$$f_{au}=\text{Concat}\left(f_{a1},f_{a2},f_{a3}\right), \quad (4)$$

where Conv2D$(\cdot)$ denotes the 2D convolutional layer. FFN$(\cdot)$ denotes the Feed-Forward Network. Concat$(\cdot)$ denotes the concatenation operation.

Next, we employ the existing 3D point cloud feature extraction network, PointNet++ [65], as the backbone to extract the point cloud feature, denoted as f_p. To effectively utilize self-projected view features to enhance the point cloud features, we do not directly concatenate them as done in [27] and [28]. Instead, at the modality level, we also adopt a cross-attention architecture [50] to fuse features from different modalities, upon which a multi-modal global feature descriptor f'_g is generated through the point-wise max pooling operation. Finally, the fused multi-modal feature f_{pu} is fed into a sparse point decoder for the prediction of a coarse-grained complete point cloud P_{coarse}. This sparse point decoder employs a multi-scale pyramid structure, as described in [5], which allows high-level features to influence the depiction of lower-level features and enables points of lower resolution to convey local geometric details to the predictions at a higher resolution.

B. Bidirection-Aware Refinement

We propose a bidirection-aware refinement block that effectively utilizes the local information of P_{in} and the structural information of P_{coarse}, as shown in Fig. 3. Unlike previous refinement networks [9], [10] that primarily focus on refining in a single direction, our method considers refinement in both directions – from complete to partial for detail perception, and from partial to complete for structure perception – thereby enhancing both the completeness and detail of completion results.

We use a two-layer Multi-Layer Perceptron (MLP) to extract features from P_{coarse}, resulting in a feature matrix f'_{coarse}, where each row represents the features $\mathbf{f}_i$ of the point $\mathbf{P}_i \subseteq P_{coarse}$. The global feature descriptor f'_g is also processed using a two-layer MLP, yielding the transformed global feature f_g. Then, f_g is concatenated with each point feature $\mathbf{f}_i$, obtaining an enhanced point feature matrix f_{coarse}. For the input partial point cloud P_{in}, we focus on the extraction of local features. To this end, we utilize the EdgeConv [73] module, which constructs a local graph by selecting the k-nearest neighbors in the feature space, thereby extracting the local features of the point cloud, denoted as f_{in}.

Next, we introduce the bidirectional perception module. Specifically, we first perform convolution operations on f_{in} and f_{coarse} using the 1D convolutional neural network to obtain the query, key and value matrixs. Then, we apply matrix multiplication and the softmax function to calculate the relative internal relationships of both the coarse-grained complete point cloud and the partial input point cloud, encompassing global and partial intra-relations.

$$\begin{cases}\mathbf{Q}_c=\text{conv1D}\left(f_{coarse}\right),\\ \mathbf{K}_c=\text{conv1D}\left(f_{coarse}\right),\\ \mathbf{V}_c=\text{conv1D}\left(f_{coarse}\right),\\ \mathbf{I}_c=\text{softmax}\left(\mathbf{Q}_c\otimes\mathbf{K}_c^{T}\right),\end{cases} \quad (5)$$

$$\begin{cases}\mathbf{Q}_{in}=\text{conv1D}\left(f_{in}\right),\\ \mathbf{K}_{in}=\text{conv1D}\left(f_{in}\right),\\ \mathbf{V}_{in}=\text{conv1D}\left(f_{in}\right),\\ \mathbf{I}_{in}=\text{softmax}\left(\mathbf{Q}_{in}\otimes\mathbf{K}_{in}^{T}\right),\end{cases} \quad (6)$$

where Conv1D$(\cdot)$ denotes the 1D convolutional layer.

Following this, we utilize the relative intra-relationships $\mathbf{I}_c$ and $\mathbf{I}_{in}$ to enhance the features of the partial point cloud

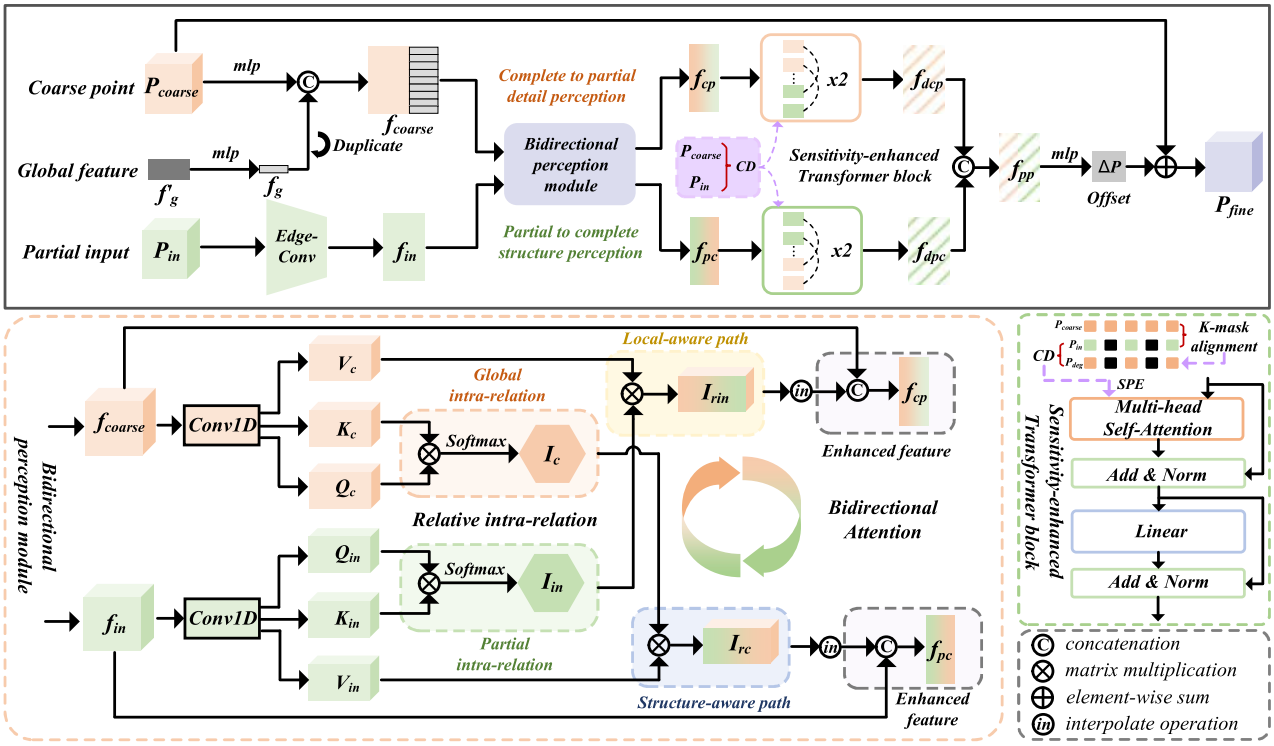

Fig. 3. Illustration of the bidirection-aware refinement (BAR) block. SPE denotes the sinusoidal positional embedding. Best viewed in colors.

and the coarse-grained complete point cloud, respectively. The improved features are represented as I_{rin} and I_{rc}.

$$\begin{cases} I_{rc} = \mathbf{I}_{in} \otimes \mathbf{V}_c, \\ I_{rin} = \mathbf{I}_c \otimes \mathbf{V}_{in}. \end{cases} \tag{7}$$

Subsequently, we utilize PyTorch's built-in interpolation function to interpolate the features and then concatenate these with their corresponding original features to obtain the enhanced features, denoted as f_{cp} and f_{pc}. For the f_{cp} and f_{pc}, we also utilize a sensitivity-enhanced Transformer block to enhance point-level geometric perception, inspired by SVDFormer [30]. However, different from it, we employ the K-mask degradation function [74] to achieve regional alignment between P_{coarse} and P_{in}, resulting in the degraded point cloud P_{deg}. We then calculate point-wise differences between P_{in} and P_{deg} using the CD function with a scaling factor of 0.2. This sensitive information is embedded into the Multi-head Self-Attention module through the sinusoidal positional embedding (SPE) function to provide spatial features. Finally, following a concatenation operation, we use a MLP to predict the coordinate offset ΔP and then add it to the coarse-grained complete point cloud to produce the final point cloud P_{fine}. To evaluate the similarity between the generated point clouds and ground truth, we adopt $CD\text{-}\ell_2$ as loss function.

$$L_{gcd} = L_{CD\text{-}\ell_2}\left(P_{coarse}, P_{gt_s}\right) + L_{CD\text{-}\ell_2}\left(P_{fine}, P_{gt}\right), \tag{8}$$

where P_{gt_s} denotes the point cloud sampled from the ground truth P_{gt}, with the same number of points of P_{coarse}.

C. Self-Learned Implicit Field Constraint

Although existing works have generated dense and complete point clouds through various refinement methods, the issue of noise within these generated point clouds has not been fully addressed. In our work, we propose a novel method that directly learns the underlying implicit surfaces from point clouds, effectively drawing outliers towards the approximate surface. Note that we do not need to explicitly extract the approximate surfaces. By doing so, we also avoid the issue of small discontinuous regions that could arise during the surface extraction process.

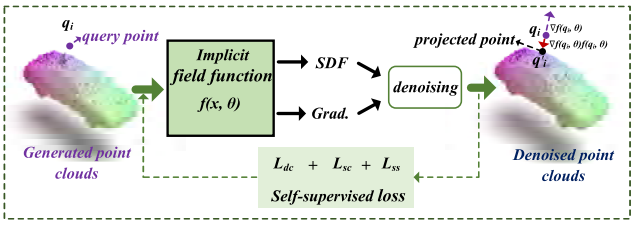

Fig. 4. Illustration of the implicit field constraint network. The core idea is to learn a signed distance field to perceive the underlying surface of the input point cloud and pull its noise points towards the approximate surface in a self-learned fashion. Best viewed in colors.

We use the simple Multi-layer Perceptron (MLP) neural network $f(x, \theta)$, parameterized by θ, to directly learn the Signed Distance Function (SDF) and gradient from the input point cloud P_{fine}, as illustrated in Fig. 4. Specifically, we sample a set of 3D query points $\boldsymbol{Q} = \{\boldsymbol{q_i}, i = 1 \ldots k\}$ around P_{fine}, following the gaussian distribution. For each query point q_i, the network predicts a step size, moving it in the direction of the gradient at q_i. Here, the step size denotes the signed distance value $f(q_i, \theta)$ and the gradient represents the partial derivative of $f(x, \theta)$ at the point q_i, which indicates the direction of the steepest increase of the signed distance value in 3D space. Therefore, the movement of the query point q_i

towards the surface point q_i' can be defined as follows.

$$q_i' = q_i - f(q_i, \theta) \times \nabla f(q_i, \theta) / \|\nabla f(q_i, \theta)\|_2, \quad (9)$$

where $\nabla f(q_i, \theta)$ denotes the gradient at q_i within the network.

The movement operation is differentiable with respect to the signed distance values and their gradients, enabling their direct optimization during training. Subsequently, we follow the Neural-pull [75] approach, using the L_2 loss to minimize the distance between the projected query point q'_i and the nearest neighbor $n_i \subseteq P_{fine}$. Thus, we define the denoising constraint loss as follows.

$$L_{dc} = \frac{1}{k} \sum_{i \in [1,k]} \left\| q_i' - n_i \right\|_2^2. \quad (10)$$

As equ. (9) does not penalize incorrect sign predictions near the surface, as discussed in [76], we introduce a periodic sign constraint loss. This loss is designed to correct inaccurate gradients and enforce directional changes in the subsequent iterations.

$$L_{sc} = \frac{1}{k} \sum_{i \in [1,k]} 1 - \cos\left(\nabla f(q_i, \theta), \frac{(q_i' - n_i)}{\|q_i' - n_i\|_2} \right), \quad (11)$$

where $\cos(\cdot)$ represents the cosine function.

Additionally, we also introduce a smoothness surface constraint term, enabling the constructed implicit field to accurately depict the underlying surface formed by the entire point cloud P_{fine}.

$$L_{ss} = \left| f\left(P_{fine}, \theta\right) \right|. \quad (12)$$

Finally, the overall denoising loss, defined through the constraints of the implicit field, can be formulated as follows.

$$L_{odc} = \alpha \cdot L_{\mathrm{dc}} + \beta \cdot L_{\mathrm{sc}} + \gamma \cdot L_{\mathrm{ss}}, \quad (13)$$

where α, β and γ represent the weight parameters, respectively.

Note that, even if the input generated point cloud P_{fine} contains some noise, our learned implicit field function could accurately capture the signed distance field around P_{fine}. Therefore, we can draw any point within P_{fine} to the surface by predicting its SDF value and corresponding gradient. Here, we represent the constrained point cloud as P_{fc}.

IV. Experiments

In this section, we first introduce the datasets employed in our experiments, along with the implementation details and evaluation metrics. Then, we present both qualitative and quantitative comparisons of our method against current state-of-the-art methods across various datasets. Finally, we conduct a series of comparative experiments and statistical analyses of model parameters to verify the effectiveness of the main modules in our proposed approach.

A. Datasets

1) The PCN Dataset: The PCN dataset is a subset of the ShapeNet [77] dataset, encompassing 8 categories. In our experiments, we follow the standard training and testing protocol proposed in [4], with the training set comprising 28974 samples and the test set comprising 1200 samples. The incomplete input point clouds are obtained by back-projecting 2.5D depth maps from eight different viewpoints and each input point cloud is processed to contain 2048 points. The complete point clouds were uniformly sampled from mesh models, yielding 16384 points per sample.

2) The ShapeNet55/34 Dataset: Compared to the 8 categories in PCN, we utilize all 55 categories of the ShapeNet dataset to assess our proposed method against others across a wider range of categories. In our experiments, we adhere to the PoinTr protocol [17] with a training-to-testing ratio of 8:2. During the training phase, we set the ratios of incomplete point clouds to be 25%, 50% and 75% of the complete point clouds, respectively. And all input point clouds are uniformly sampled to 2048 points. The complete point clouds are sampled to 8192 points. During the testing phase, we select eight fixed viewpoints and set the number of incomplete points to 2048, 4096 or 6144, representing 25%, 50% or 75% of the complete point cloud, respectively. These configurations define three levels of difficulty: simple (S), moderate (M) and hard (H). In contrast to the ShapeNet55 dataset, the ShapeNet34 dataset use 34 categories for training while reserving 21 unseen categories for testing. Specifically, we use 46765 samples for training. For testing, 3400 samples are dedicated to evaluate known categories and 2305 samples to assessing unknown categories. Additionally, the ShapeNet34 also sets three difficulty levels - S, M and H – to evaluate the performance of different methods.

3) The KITTI (car) Dataset: To further evaluate the performance of our proposed method in real-world scenarios, we perform tests on the car category of the KITTI dataset. This dataset extracts car point clouds from LiDAR scans using 3D bounding boxes, yielding a total of 2401 partial point cloud samples. Compared to the synthetic datasets, KITTI's (car) point clouds are much sparser and lack real complete point clouds as ground truth.

B. Implementation Details

Our method employs a two-stage training strategy: In the stage I, we generate the fine-grained point cloud P_{fine} from the partial input P_{in}; in the stage II, we focus on the study of moving noise from the generated point cloud P_{fine} and obtain the noise-free point cloud P_{fc}. We train our model on NVIDIA GeForce RTX 3090 GPUs using PyTorch in both stages.

Stage I: We use Adam optimizer with $\beta_1 = 0.9$ and $\beta_2 = 0.999$ and set the initial learning rate to 0.0001. We train our model for 300 epochs with a batch size of 16. The learning rate is decayed by 0.7 after around every 50 epochs. The depth map size is set as 224×224. We perform the bidirection-aware refiner twice. In the second iteration, we downsample the generated point clouds using the iterative farthest point sampling (FPS) algorithm to ensure consistency with the resolution of the coarse-grained complete point clouds, which is 512.

Stage II: For the input fine-grained point cloud P_f, we sample 25 query points from each point to form the query set $\boldsymbol{Q} = \{\boldsymbol{q_i}, i = 1 \ldots k\}$ following the Neural-Pull [75]. For

TABLE II

QUANTITATIVE RESULTS ON THE PCN DATASET (CD-ℓ_1 $\times 10^3$ (LOWER IS BETTER) AND F-SCORE@1% (HIGHER IS BETTER)). THE BOLD VALUES ARE THE BEST

Methods	airplane	cabinet	car	chair	lamp	sofa	table	watercraft	CD-ℓ_1	DCD	F-score
PCN [4]	6.39	12.00	9.62	12.30	12.70	13.43	10.23	9.92	10.82	-	0.617
TopNet [21]	7.61	13.31	10.90	13.82	14.44	14.78	11.22	11.12	12.15	-	0.528
GRNet [23]	6.45	10.37	9.45	9.41	7.96	10.51	8.44	8.04	8.83	0.622	0.708
CRN [9]	4.79	9.97	8.31	9.49	8.94	10.69	7.81	8.05	8.51	-	0.652
PoinTr [17]	4.75	10.47	8.68	9.39	7.75	10.93	7.78	7.29	8.38	0.611	0.745
SnowflakeNet [80]	4.29	9.16	8.08	7.89	6.07	9.23	6.55	6.40	7.21	0.585	0.801
PMP-Net++ [81]	4.39	9.96	8.53	8.09	6.06	9.82	7.17	6.52	7.56	0.611	0.781
ProxyFormer [82]	4.01	9.01	7.88	7.11	5.35	8.77	6.03	5.98	6.77	0.577	0.820
SeedFormer [32]	3.85	9.05	8.06	7.06	**5.21**	8.85	6.05	5.85	6.74	0.583	0.818
SVDFormer [30]	3.62	8.79	7.46	6.91	5.33	8.49	5.90	5.83	6.54	0.536	0.841
Ours	**3.60**	**8.64**	**7.33**	**6.82**	5.24	**8.37**	**5.69**	**5.77**	**6.43**	**0.521**	**0.850**

the PCN and KITTI (car) datasets, the k is 409600. For the ShapeNet55/34 dataset, the k is 204800. To learn the SDF values from P_{fine}, we employ a neural network similar to OccNet [78] to predict SDF values for given 3D queries. Our network is trained for 2500 epochs with a batch size of 5000, which means that each batch consists of 5000 query points. α, β and γ are set as 1, 0.01, 0.01, respectively.

C. Evaluation Metrics

In our experiments, we primarily adopt the following evaluation metrics: CD-ℓ_1, CD-ℓ_2, Density-aware Chamfer Distance (DCD) [15] and F-score [79] for the PCN and ShapeNet55/34 datasets. Considering the absence of ground truth in the KITTI (car) dataset, we also follow the approach used in [17], employing the Fidelity Distance (FD) and Minimal Matching Distance (MMD) for evaluation.

D. Point Cloud Completion Results

1) Results on PCN Dataset: We compare our method with ten previous approaches, encompassing both classical methods and current competitive approaches. Table. II demonstrates that our method achieves the best results across seven categories in the PCN dataset. Although it does not achieve the best results in the lamp category, it was very close, trailing the top performance by only 0.03 (in terms of CD-ℓ_1 $\times$ 10^3). Furthermore, we also conduct a comprehensive evaluation of our method against others using three widely used metrics: CD-ℓ_1, DCD and F-score, where our method consistently outperforms previous approaches. Compared with the latest method SVDFormer [30], our approach shows improvements of 0.11, 0.015 and 0.009 in the CD-ℓ_1, DCD and F-score metrics, respectively.

Fig. 5 presents the qualitative results of our method compared to other competitors: PCN [4], TopNet [21], PoinTr [30], SeedFormer [32] and SVDFormer [30]. To ensure a fair comparison, we use the pre-trained models provided by PCN and PoinTr for generating point clouds. For Seedformer, we utilize the generated point clouds provided by the authors.

TABLE III

QUANTITATIVE RESULTS ON THE SHAPENET55 DATASET (CD-ℓ_2 $\times 10^3$ (LOWER IS BETTER) AND F-SCORE@1% (HIGHER IS BETTER)). THE BOLD VALUES ARE THE BEST

Methods	CD-S	CD-M	CD-H	CD-ℓ_2	DCD	F-score
PCN [4]	1.94	1.96	4.08	2.66	0.618	0.133
GRNet [23]	1.35	1.71	2.85	1.97	0.592	0.238
PoinTr [17]	0.58	0.88	1.79	1.09	0.575	0.464
SeedFormer [32]	0.50	0.77	1.49	0.92	0.558	**0.472**
SVDFormer [30]	0.48	0.70	1.30	0.83	0.541	0.451
Ours	**0.47**	**0.68**	**1.27**	**0.81**	**0.536**	0.468

As for TopNet and SVDFormer, we rigorously reproduce their methods as described in their original papers for a comprehensive comparison. The qualitative results clearly show that transformer-based methods (PoinTr, Seedformer, SVDFormer and Ours) significantly outperform earlier methods such as PCN and TopNet. However, it is also observed that all these competitive methods, to varying extents, exhibit some degree of noise, as indicated by red dashed circles in Fig. 5. This is attributed primarily to two reasons: 1) the lack of an effective method for noise constraint, 2) the noise present in the real point cloud results in inaccurate supervision signals, as marked by purple dashed circles in Fig. 5. In contrast, our approach effectively mitigates noise by pulling it towards the approximated surface.

2) Results on ShapeNet55/34 Dataset: We first evaluate the performance of our method against five competitive approaches on the Shapenet55 dataset, which encompasses a broader range of categories. It is important to note that, except for the pre-trained model provided by PoinTr [30], the results for other methods [4], [23], [30], [32] are strictly replicated according to the original settings described in their respective papers. Table. III presents the performance of our method and other competitors across three difficulty levels (simple, moderate and hard) in terms of the CD-ℓ_2, as well as the overall CD-ℓ_2, DCD and F-score metrics. The results demonstrate a significant improvement in our method over the second-best in terms of CD-ℓ_2, particularly at higher difficulty

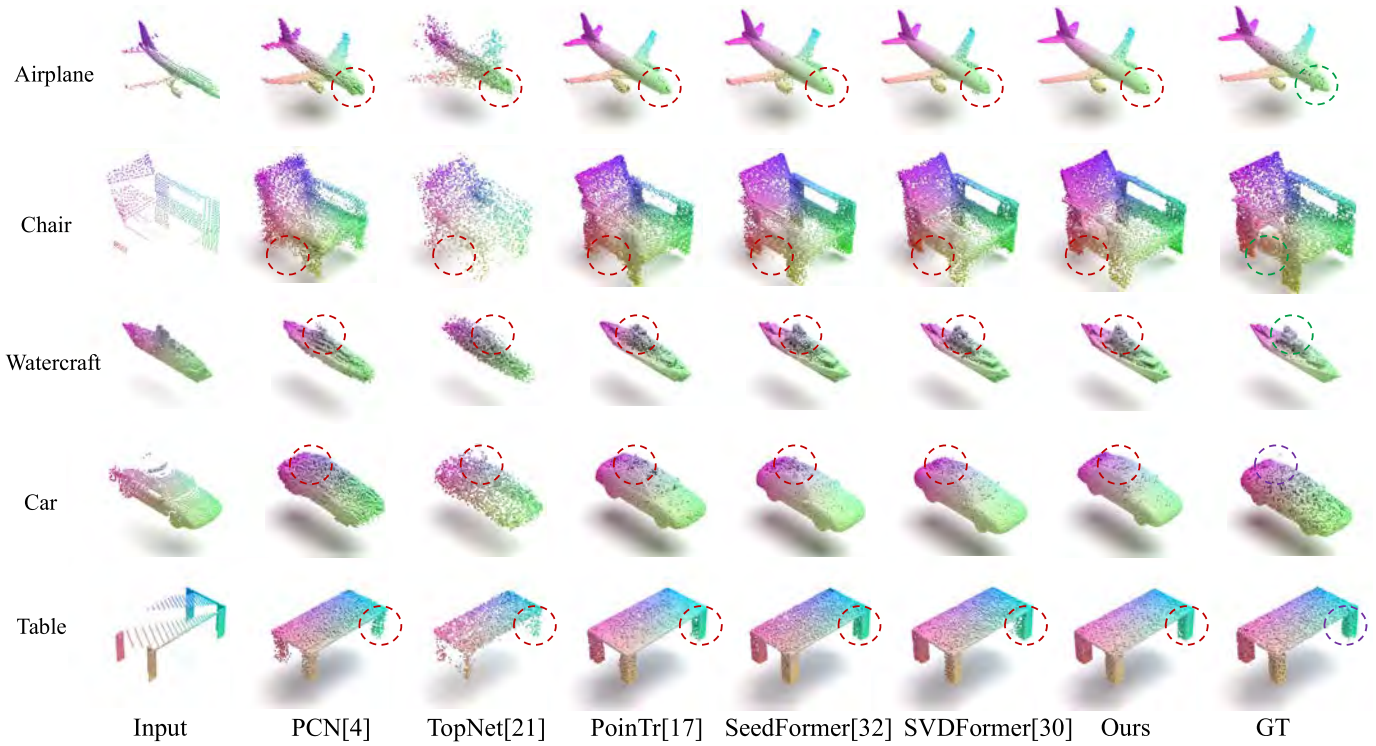

Fig. 5. Qualitative point cloud results on the PCN dataset by different methods. We demonstrate that our approach not only performs well in terms of completeness and detail but also produces fewer noise points across a wide variety of examples: airplane, chair, watercraft, car and table. Best viewed in colors.

TABLE IV

QUANTITATIVE RESULTS ON THE SHAPENET34 DATASET (CD-ℓ_2 $\times 10^3$ (LOWER IS BETTER) AND F-SCORE@1% (HIGHER IS BETTER)). THE BOLD VALUES ARE THE BEST

Methods	34 seen categories						21 unseen categories					
	CD-S	CD-M	CD-H	CD-ℓ_2	DCD	F-score	CD-S	CD-M	CD-H	CD-ℓ_2	DCD	F-score
PCN [4]	1.87	1.81	2.97	2.22	0.624	0.150	3.17	3.08	5.29	3.85	0.644	0.101
GRNet [23]	1.26	1.39	2.57	1.74	0.600	0.251	1.85	2.25	4.87	2.99	0.625	0.216
PoinTr [17]	0.76	1.05	1.88	1.23	0.575	0.421	1.04	1.67	3.44	2.05	0.604	0.384
SeedFormer [32]	0.48	0.70	1.30	0.83	0.561	0.452	0.61	1.07	2.35	1.34	0.586	0.402
SVDFormer [30]	0.46	0.65	1.13	0.75	0.538	0.457	0.61	1.05	2.19	1.28	0.554	0.427
Ours	**0.45**	**0.63**	**1.08**	**0.72**	**0.529**	**0.459**	**0.60**	**1.03**	**2.10**	**1.24**	**0.542**	**0.430**

levels. This improvement is attributed to our method's effective learning of spatial relationships in point clouds. In terms of the overall CD-ℓ_2 and DCD metrics, our method consistently surpasses others. Nevertheless, the existing gap of 0.004 in F-score compared to the best result inspires us to further enhance our approach. Additionally, complete results for all 55 categories are detailed in Table V.

Similarly, we also compare our method with aforementioned approaches on the ShapeNet34 dataset to assess its generalization capability on unknown categories. Table. IV demonstrates that our approach consistently outperforms previous methods across three metrics (CD-ℓ_2, DCD and F-score) for the 34 known categories. Notably, our method also achieves the best results in 21 unknown categories, which are not seen during training. Compared to the latest method SVDFormer, our approach shows improvements of 0.04, 0.012 and 0.003 in CD-ℓ_2, DCD and F-score, respectively, further evidencing its superior generalization capabilities.

We also present qualitative visual comparisons of our method against competitive approaches such as PCN, PoinTr and SVDFormer, across varying degrees of data incompleteness (25%, 50% and 75%) on the Shapenet55 dataset. The results demonstrate that our method consistently achieves the best point cloud completion results, even in scenarios of significant data incompleteness (75%), as indicated by the read dashed circles. Additionally, our method effectively mitigates the noise problem. *However, some undesired results are observed, as in the case of the bus (sourced from Fig. 6 of the ShapeNet55), where a minimal number of cable line points are mistakenly identified as noise, leading to their movement. This observation encourages us to consider further improvements of our approach, such as incorporating additional textual prior knowledge.*

3) Results on KITTI Dataset: To evaluate our method's performance in real-world scenarios, we perform comparisons with four classical methods – PCN [4], GRNet [23],

TABLE V

DETAILED QUANTITATIVE RESULTS ON THE SHAPENET55 DATASET (CD-ℓ_2 $\times 10^3$ (LOWER IS BETTER)). S.,M. AND H. STAND FOR THE SIMPLE, MODERATE AND HARD SETTINGS, RESPECTIVELY. THE BOLD VALUES ARE THE BEST

CD-ℓ_2($\times$ 1000)	PCN [4]			GRNet [23]			PoinTr [17]			SeedFormer [32]			Ours		
	S.	M.	H.	S.	M.	H.	S.	M.	H.	S.	M.	H.	S.	M.	H.
airplane	0.90	0.89	1.32	0.87	0.87	1.27	0.27	0.38	0.69	0.23	0.35	0.61	**0.22**	**0.33**	**0.54**
trash bin	2.16	2.18	5.15	1.69	2.01	3.48	0.80	1.15	2.15	0.73	1.08	1.94	**0.69**	**0.94**	**1.59**
bag	2.11	2.04	4.44	1.41	1.7	2.97	0.53	0.74	1.51	0.43	0.67	1.28	**0.40**	**0.62**	**1.14**
basket	2.21	2.10	4.55	1.65	1.84	3.15	0.73	0.88	1.82	**0.65**	0.83	1.54	**0.65**	**0.82**	**1.33**
bathtub	2.11	2.09	3.94	1.46	1.73	2.73	0.64	0.94	1.68	0.52	0.82	1.45	**0.51**	**0.73**	**1.21**
bed	2.86	3.07	5.54	1.64	2.03	3.7	0.76	1.1	2.26	0.63	0.91	1.89	**0.60**	**0.86**	**1.64**
bench	1.31	1.24	2.14	1.03	1.09	1.71	0.38	0.52	0.94	0.32	0.42	0.84	**0.31**	**0.37**	**0.64**
birdhouse	3.29	3.53	6.69	1.87	2.4	4.71	0.98	1.49	3.13	**0.76**	1.30	2.46	0.77	**1.18**	**2.20**
bookshelf	2.70	2.70	4.61	1.42	1.71	2.78	0.71	1.06	1.93	0.57	0.84	1.57	**0.56**	**0.82**	**1.40**
bottle	1.25	1.43	4.61	1.05	1.44	2.67	0.37	0.74	1.50	0.31	0.63	1.21	**0.30**	**0.56**	**1.08**
bowl	2.05	1.83	3.66	1.6	1.77	2.99	0.68	0.78	1.44	0.56	0.65	1.18	**0.53**	**0.58**	**0.95**
bus	1.20	1.14	2.08	1.06	1.16	1.48	0.42	0.55	0.79	0.42	0.55	0.73	**0.40**	**0.51**	**0.66**
cabinet	1.60	1.49	3.47	1.27	1.41	2.09	0.55	0.66	1.16	0.57	0.69	1.05	**0.54**	**0.62**	**0.91**
camera	4.05	4.54	8.27	2.14	3.15	6.09	1.1	2.03	4.34	0.83	1.68	3.45	**0.80**	**1.47**	**2.89**
can	2.02	2.28	6.48	1.58	2.11	3.81	0.68	1.19	2.14	0.58	1.03	1.79	**0.56**	**0.92**	**1.76**
cap	1.82	1.76	4.20	1.17	1.37	3.05	0.46	0.62	1.64	0.33	0.45	1.18	**0.32**	**0.41**	**0.82**
car	1.48	1.47	2.60	1.29	1.48	2.14	0.64	0.86	1.25	0.65	0.86	1.17	**0.58**	**0.77**	**1.00**
cellphone	0.80	0.79	1.71	0.82	0.91	1.18	0.32	0.39	0.6	0.31	0.40	0.54	**0.26**	**0.36**	**0.45**
chair	1.70	1.81	3.34	1.24	1.56	2.73	0.49	0.74	1.63	0.41	0.65	1.38	**0.38**	**0.56**	**1.14**
clock	2.10	2.01	3.98	1.46	1.66	2.67	0.62	0.84	1.65	0.53	0.74	1.35	**0.50**	**0.66**	**1.16**
keyboard	0.82	0.82	1.04	0.74	0.81	1.09	0.30	0.39	0.45	0.28	0.36	0.45	**0.27**	**0.32**	**0.40**
dishwasher	1.93	1.66	4.39	1.43	1.59	2.53	**0.55**	0.69	1.42	0.56	0.69	1.30	**0.55**	**0.60**	**1.13**
display	1.56	1.66	3.26	1.13	1.38	2.29	0.48	0.67	1.33	**0.39**	0.59	1.10	**0.39**	**0.52**	**0.93**
earphone	3.13	2.94	7.56	1.78	2.18	5.33	0.81	1.38	3.78	0.64	1.04	2.75	**0.57**	**0.86**	**2.51**
faucet	3.21	3.48	7.52	1.81	2.32	4.91	0.71	1.42	3.49	**0.55**	1.15	2.63	**0.55**	**1.05**	**2.20**
filecabinet	2.02	1.97	4.14	1.46	1.71	2.89	0.63	0.84	1.69	0.63	0.84	1.49	**0.62**	**0.76**	**1.34**
guitar	0.42	0.38	1.23	0.44	0.48	0.76	0.14	0.21	0.42	0.13	0.19	0.32	**0.12**	**0.18**	**0.29**
helmet	3.76	4.18	7.53	2.33	3.18	6.03	0.99	1.93	4.22	0.79	1.52	3.61	**0.71**	**1.28**	**2.30**
jar	2.57	2.82	6.00	1.72	2.37	4.37	0.77	1.33	2.87	0.63	1.13	2.36	**0.60**	**0.98**	**2.03**
knife	0.94	0.62	1.37	0.72	0.66	0.96	0.20	0.33	0.56	0.15	0.28	0.45	**0.15**	**0.25**	**0.42**
lamp	3.10	3.45	7.02	1.68	2.43	5.17	0.64	1.4	3.58	**0.45**	**1.06**	2.67	0.48	**1.06**	**2.66**
laptop	0.75	0.79	1.59	0.83	0.87	1.28	0.32	0.34	0.6	0.32	0.37	0.55	**0.30**	**0.33**	**0.45**
loudspeaker	2.50	2.45	5.08	1.75	2.08	3.45	0.78	1.16	2.17	0.67	1.01	1.80	**0.63**	**0.91**	**1.60**
mailbox	1.66	1.74	5.18	1.15	1.59	3.42	0.39	0.78	2.56	0.30	0.67	2.04	**0.27**	**0.53**	**1.69**
microphone	3.44	3.90	8.52	2.09	2.76	5.70	0.70	1.66	4.48	0.62	1.61	3.66	**0.60**	**1.33**	**2.81**
microwaves	2.20	2.01	4.65	1.51	1.72	2.76	0.67	0.83	1.82	0.63	0.79	1.47	**0.61**	**0.68**	**1.37**
motorbike	2.03	2.01	3.13	1.38	1.52	2.26	0.75	1.10	1.92	0.68	0.96	1.44	**0.61**	**0.85**	**1.28**
mug	2.45	2.48	5.17	1.75	2.16	3.79	0.91	1.17	2.35	0.79	1.03	2.06	**0.74**	**0.96**	**1.74**
piano	2.64	2.74	4.83	1.53	1.82	3.21	0.76	1.06	2.23	0.62	0.87	1.79	**0.56**	**0.72**	**1.35**
pillow	1.85	1.81	3.68	1.42	1.67	3.04	0.61	0.82	1.56	0.48	0.75	1.41	**0.42**	**0.57**	**1.00**
pistol	1.25	1.17	2.65	1.11	1.06	1.76	0.43	0.66	1.30	0.37	0.56	0.96	**0.36**	**0.55**	**0.86**
flowerpot	3.32	3.39	6.04	2.02	2.48	4.19	1.01	1.51	2.77	0.93	1.30	2.32	**0.88**	**1.29**	**2.00**
printer	2.90	3.19	5.84	1.56	2.38	4.24	0.73	1.21	2.47	0.58	1.11	2.13	**0.57**	**0.92**	**1.81**
remote	0.99	0.97	2.04	0.89	1.05	1.29	0.36	0.53	0.71	**0.29**	0.46	0.62	**0.29**	**0.43**	**0.58**
rifle	0.98	0.80	1.31	0.83	0.77	1.16	0.3	0.45	0.79	0.27	0.41	0.66	**0.24**	**0.37**	**0.58**
rocket	1.05	1.04	1.87	0.78	0.92	1.44	0.23	0.48	0.99	0.21	0.46	0.83	**0.18**	**0.41**	**0.76**
skateboard	1.04	0.94	1.68	0.82	0.87	1.24	0.28	0.38	0.62	0.23	0.32	0.62	**0.22**	**0.30**	**0.47**
sofa	1.65	1.61	2.92	1.35	1.45	2.32	0.56	0.67	1.14	0.50	0.62	1.02	**0.47**	**0.56**	**0.84**
stove	2.07	2.02	4.72	1.46	1.72	3.22	0.63	0.92	1.73	0.59	0.87	1.49	**0.57**	**0.79**	**1.73**
table	1.56	1.50	3.36	1.15	1.33	2.33	0.46	0.64	1.31	0.41	0.58	1.18	**0.39**	**0.51**	**1.15**
telephone	0.80	0.80	1.67	0.81	0.89	1.18	0.31	0.38	0.59	0.31	0.39	0.55	**0.30**	**0.36**	**0.47**
tower	1.91	1.97	4.47	1.26	1.69	3.06	0.55	0.9	1.95	0.47	0.84	1.65	**0.46**	**0.76**	**1.45**
train	1.50	1.41	2.37	1.09	1.14	1.61	0.50	0.70	1.12	0.51	0.66	1.01	**0.45**	**0.60**	**0.89**
watercraft	1.46	1.39	2.40	1.09	1.12	1.65	0.41	0.62	1.07	0.35	0.56	0.92	**0.32**	**0.52**	**0.83**
washer	2.42	2.31	6.08	1.72	2.05	4.19	0.75	1.06	2.44	0.64	0.91	2.04	**0.63**	**0.84**	**1.59**
mean	1.96	1.98	4.09	1.35	1.63	2.86	0.58	0.88	1.8	0.50	0.77	1.49	**0.47**	**0.68**	**1.27**

PoinTr [30] and SeedFormer [32] – on the KITTI (car) dataset. As shown in Table. VI, our method consistently outperforms other competitors in terms of the MMD metric. Notably, in the Fidelity Distance (FD) metric evaluation, the value of the method PoinTr is zero, while our method secures a second-place ranking. This outcome is due to PoinTr's strategy, which focuses on exclusively predicting missing components and merging the input into the final result.

We further offer a qualitative comparison, highlighting the distinctions between our method and PoinTr across three varying levels of completion difficulty. Fig. 7 shows that the point clouds completed by PoinTr display notable inaccuracies and a lack of coherence between the completed and existing points, as indicated by the read dashed circles. Conversely, our approach effectively mitigates these issues.

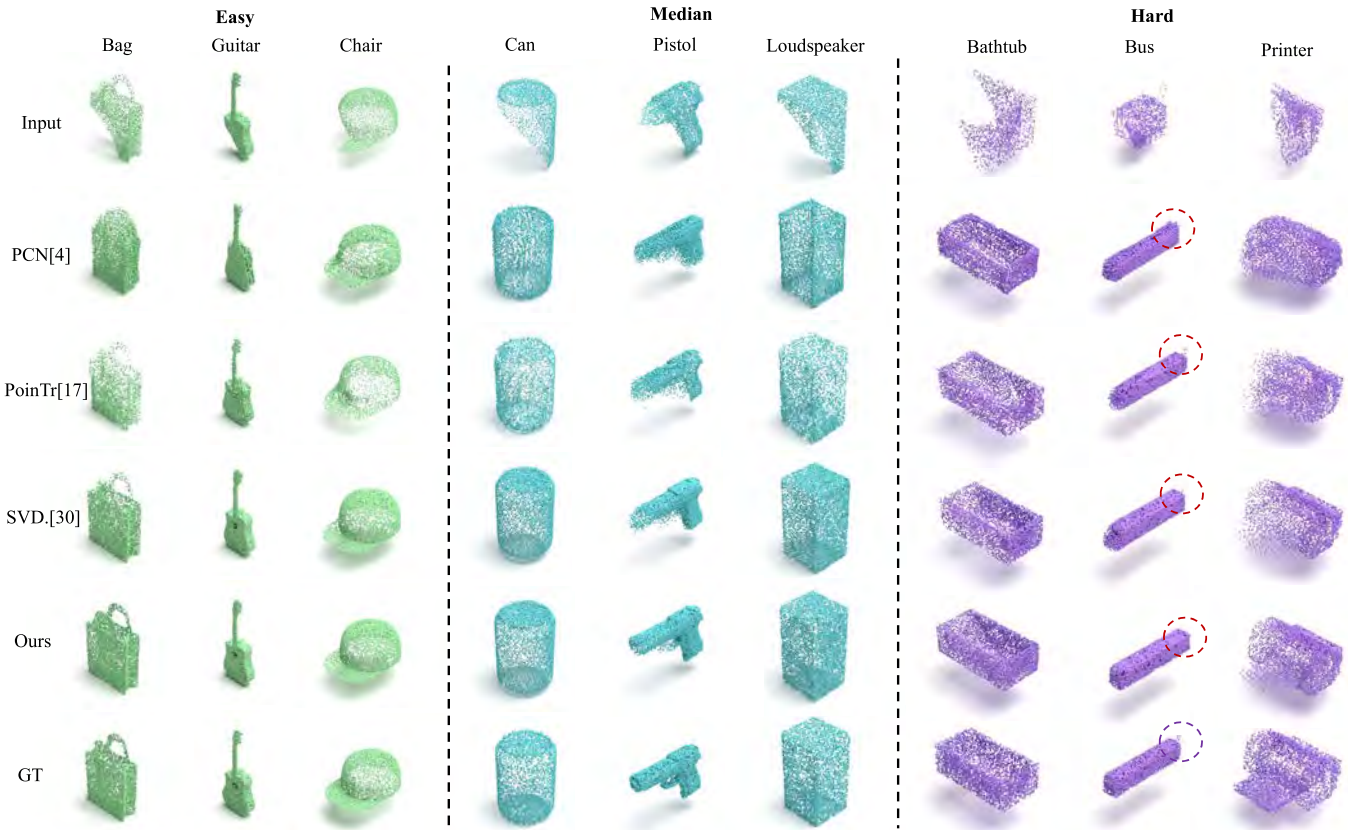

Fig. 6. Qualitative point cloud results on the ShapeNet55 dataset by different methods. We compare our method against competitive approaches across three levels of difficulty, highlighting our method's superiority. Our approach consistently demonstrates stable performance, even in challenging scenarios. Best viewed in colors.

TABLE VI

QUANTITATIVE COMPARISON ON THE KITTI (CAR) DATASET. FOR FIDELITY DISTANCE (FD) AND MINIMAL MATCHING DISTANCE (MMD), LOWER IS BETTER. THE BOLD VALUES ARE THE BEST

	PCN [4]	GRNet [23]	PoinTr [17]	Seed. [32]	Ours
FD	2.235	0.816	**0.000**	0.151	0.119
MMD	1.366	0.568	0.526	0.516	**0.494**

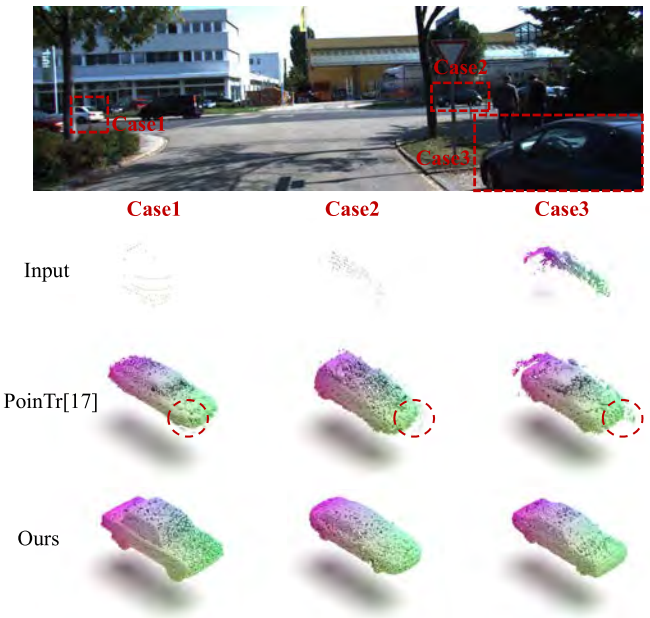

Fig. 7. Qualitative point cloud results on the KITTI (car) dataset by different methods. We compare our method with the approach of PoinTr across three varying cases of completion difficulty. Best viewed in colors.

E. Ablation Studies

In this section, we present a series of ablation studies to offer insights into the design of our modules. First, we perform an evaluation analysis on the necessity of the self-projected view. Subsequently, we assess the performance of the self-learned implicit field constraint module. Finally, we conduct the complexity analysis of different methods. Unless otherwise specified, all experiments are performed on the PCN dataset.

1) Necessity of the Self-Projected View: In this section, we first compare the results of employing the self-projected views with those not using them. We then delved into the effectiveness of the cross-view augmentation (CVA) and cross-modal fusion (CMF) modules within the self-projected views, comparing them against direct concatenation operations (baseline). To ensure a more precise evaluation of this module's effectiveness, we exclude the self-learned implicit field constraint module in this experiment to prevent its potential benefits from masking the contrast in results stemming from different designs.

Table. VII demonstrates a significant performance improvement when using self-projected views compared to directly utilizing point clouds. This implies that even when dealing

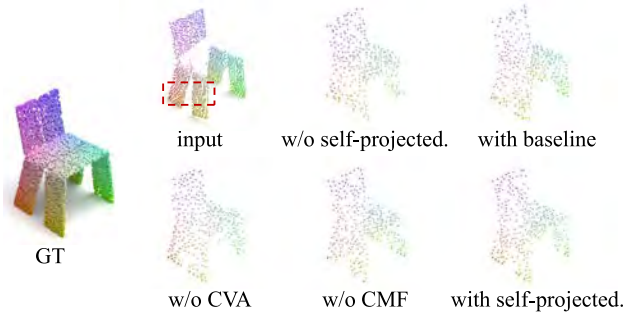

Fig. 8. Ablation study on the necessity of the self-projected views.

TABLE VII

ABLATION STUDY ON THE NECESSITY OF THE SELF-PROJECTED VIEWS. THE BOLD VALUES ARE THE BEST

Methods	CD-ℓ_1	DCD	F-score
w/o self projected.	7.27	0.606	0.593
baseline	7.01	0.593	0.587
w/o CVA	6.57	0.541	0.833
w/o CMF	6.71	0.576	0.821
with self projected.	**6.47**	**0.524**	**0.848**

TABLE VIII

ABLATION STUDY ON THE EFFECTIVENESS OF THE SELF-LEARNED IMPLICIT FIELD CONSTRAINT. THE BOLD VALUES ARE THE BEST

Methods	CD-ℓ_1	DCD	F-score
w/o self-learned IFS	6.47	0.524	0.848
with self-learned IFS	**6.43**	**0.521**	**0.850**

with incomplete point cloud inputs, self-projected views remain effective in assisting point cloud completion tasks, as confirmed by SVDFormer. Furthermore, we also note the significant benefits brought about by the CVA and CMF modules. Finally, we present a qualitative comparison of coarse-grained complete point clouds under different design configurations. Fig. 8 demonstrates that the concurrent application of cross-view augmentation and cross-modal fusion modules generates coarse-grained complete point clouds which more accurately represent the object's shape. Furthermore, we also observe that simply concatenating multi-modal information does not effectively differentiate the boundaries between the legs of stools, as indicated by the dashed red box.

2) Effectiveness of the Self-Learned Implicit Field Constraint: This section compares the performance of adopting the self-learned implicit field constraint (IFS) with that without utilizing it. Table. VIII demonstrates that using the self-learned IFS leads to improvements in our method across the CD, DCD and F-score metrics. Moreover, we also qualitatively show the variations before and after applying the self-learned IFS. Fig. 9 clearly illustrates that using the self-learned IFS could more effectively pull noise points in the generated point clouds towards the approximate surfaces, as indicated by the dashed red box. We also conduct a visual comparisons of denoising effects across different training epochs, as shown in Fig. 10. It is observed that at 2500 epochs, noise points are effectively relocated to the approximate surface. Therefore, considering the time cost, we do not train for more epochs.

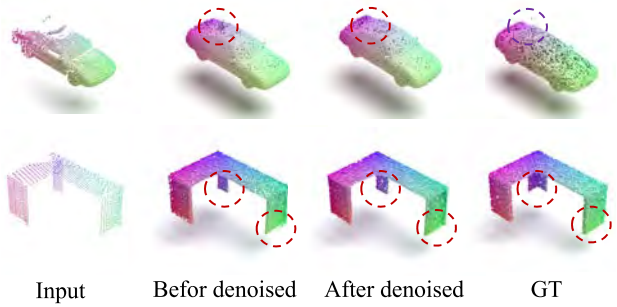

Fig. 9. Ablation study on the effectiveness of the self-learned implicit field constraint.

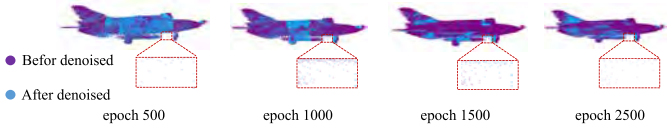

Fig. 10. Comparative analysis of denoising effects across different training epochs.

TABLE IX

COMPLEXITY ANALYSIS ON THE PCN DATASET. WE REPORT THE PARAMS, FLOPS, INFERENCE TIME, CD-ℓ_1 AND F-SCORE AS REFERENCES. THE BOLD VALUES ARE THE BEST

Method	Params	FLOPS	Inference.	CD-ℓ_1	F-score
GRNet [23]	76.71 M	25.88G	10.51ms	8.83	0.708
PoinTr [17]	30.90 M	10.41G	12.22ms	8.38	0.745
seedFormer [32]	**3.20 M**	29.61G	39.26ms	6.74	0.818
ProxyFormer [30]	12.16 M	**9.88G**	**9.97ms**	6.77	0.820
Ours	21.71M	11.28G	25.34ms	**6.43**	**0.850**

3) Complexity Analysis: Table IX presents a comparison of our method with four competitive methods in terms of parameters (Params), theoretical computational cost (FLOPs) and inference time. Our approach achieves superior performance in the CD and F-score metrics while maintaining appropriate parameters, FLOPs and inference time, effectively balancing these elements. We argue that the increased number of parameters in our method may stem from the feature extraction and fusion processes in the self-projected views. It is worth noting that our method does not provide a significant advantage in inference time. Future efforts will focus on optimizing our method to reduce its parameters and improve inference times.

V. CONCLUSION AND FUTURE WORK

In this work, we introduce a novel point cloud completion framework that significantly differs from existing 3D point cloud completion methods, which either directly use available point clouds or incorporate additional RGB images and corresponding camera parameters as input. We propose enhancing the point cloud completion task from the perspective of self-projected views, while also considering noise reduction within the generated point clouds through implicit field constraints. Extensive quantitative and qualitative experiments on PCN, ShapeNet55/34 and KITTI (car) datasets have demonstrated the superiority of our method. Moreover, we have validated the effectiveness of our key modules through necessary ablation studies and visualizations.

Future Works: Despite our work has achieved promising results, we are still aware that our method could be improved from the following aspects in the future: 1) We leverage implicit field constraints for noise reduction within generated point clouds. However, it is currently implemented as a two-stage process rather than an end-to-end generation of results. Future efforts could focus on improving our method to enable an end-to-end implementation. 2) Our work does not explicitly extract the implicit surface and then sample target resolution point clouds. We argue that constructing the complete underlying surface will help seamlessly integrate tasks of point cloud completion and arbitrary resolution point cloud upsampling. 3) Moreover, existing point cloud completion efforts primarily focus on synthetic datasets, necessitating pre-processing and scale normalization for real-world datasets. The field of image completion has demonstrated adeptness at adapting to scale differences when training on large-scale datasets. Consequently, we aim to employ image-guided pre-trained models to address the issue of scale inconsistency in input point clouds, thereby more effectively meeting the demands of real-world application scenarios.

REFERENCES

[1] Z. Yuan, X. Song, L. Bai, Z. Wang, and W. Ouyang, "Temporal-channel transformer for 3D LiDAR-based video object detection for autonomous driving," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 32, no. 4, pp. 2068–2078, Apr. 2022.

[2] F. Tang, Y. Wu, X. Hou, and H. Ling, "3D mapping and 6D pose computation for real time augmented reality on cylindrical objects," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 30, no. 9, pp. 2887–2899, Sep. 2020.

[3] Y.-H. Shiau, Y.-T. Kuo, P.-Y. Chen, and F.-Y. Hsu, "VLSI design of an efficient flicker-free video defogging method for real-time applications," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 29, no. 1, pp. 238–251, Jan. 2019.

[4] W. Yuan, T. Khot, D. Held, C. Mertz, and M. Hebert, "PCN: Point completion network," in *Proc. Int. Conf. 3D Vis. (3DV)*, Sep. 2018, pp. 728–737.

[5] L. Tan, X. Lin, D. Niu, D. Wang, M. Yin, and X. Zhao, "Projected generative adversarial network for point cloud completion," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 33, no. 2, pp. 771–781, Feb. 2023.

[6] H. Xiao, Y. Li, W. Kang, and Q. Wu, "Distinguishing and matching-aware unsupervised point cloud completion," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 33, no. 9, pp. 5160–5173, Sep. 2023.

[7] Z. Chen et al., "AnchorFormer: Point cloud completion from discriminative nodes," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2023, pp. 13581–13590.

[8] L. Pan et al., "Variational relational point completion network," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 8524–8533.

[9] X. Wang, M. H. Ang, and G. H. Lee, "Cascaded refinement network for point cloud completion," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 787–796.

[10] T. Huang et al., "RFNet: Recurrent forward network for dense point cloud completion," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 12488–12497.

[11] X. Wen et al., "PMP-Net: Point cloud completion by learning multi-step point moving paths," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 7439–7448.

[12] J. Tang, Z. Gong, R. Yi, Y. Xie, and L. Ma, "LAKe-Net: Topology-aware point cloud completion by localizing aligned keypoints," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 1716–1725.

[13] 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 *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 4617–4626.

[14] Z. Wang, W. Li, and D. Xu, "Domain adaptive sampling for cross-domain point cloud recognition," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 33, no. 12, pp. 7604–7615, Dec. 2023.

[15] T. Wu, L. Pan, J. Zhang, T. Wang, Z. Liu, and D. Lin, "Density-aware chamfer distance as a comprehensive metric for point cloud completion," 2021, *arXiv:2111.12702*.

[16] F. Lin et al., "Hyperbolic chamfer distance for point cloud completion," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2023, pp. 14595–14606.

[17] X. Yu, Y. Rao, Z. Wang, Z. Liu, J. Lu, and J. Zhou, "PoinTr: Diverse point cloud completion with geometry-aware transformers," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 12498–12507.

[18] W. Zhang, H. Zhou, Z. Dong, J. Liu, Q. Yan, and C. Xiao, "Point cloud completion via skeleton-detail transformer," *IEEE Trans. Vis. Comput. Graph.*, vol. 29, no. 10, pp. 1–14, Oct. 2022.

[19] Z. Lyu, Z. Kong, X. Xu, L. Pan, and D. Lin, "A conditional point diffusion-refinement paradigm for 3D point cloud completion," 2021, *arXiv:2112.03530*.

[20] L. Zhou, Y. Du, and J. Wu, "3D shape generation and completion through point-voxel diffusion," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 5806–5815.

[21] L. P. Tchapmi, V. Kosaraju, H. Rezatofighi, I. Reid, and S. Savarese, "TopNet: Structural point cloud decoder," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 383–392.

[22] Z. Huang, Y. Yu, J. Xu, F. Ni, and X. Le, "PF-Net: Point fractal network for 3D point cloud completion," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2020, pp. 7662–7670.

[23] H. Xie, H. Yao, S. Zhou, J. Mao, S. Zhang, and W. Sun, "GRNet: Gridding residual network for dense point cloud completion," in *Proc. Eur. Conf. Comput. Vis.*, 2020, pp. 365–381.

[24] X. Wang, M. H. Ang, and G. Hee Lee, "Voxel-based network for shape completion by leveraging edge generation," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 13169–13178.

[25] S. Liu, D. Li, W. Huang, Y. Cao, and S. Chen, "MRAC-Net: Multi-resolution anisotropic convolutional network for 3D point cloud completion," in *Proc. Pacific Rim Int. Conf. Artif. Intell. (PRICAI)*, Nov. 2021, pp. 403–414.

[26] X. Deng, X. Hu, N. E. Buris, P. An, and Y. Chen, "3D grid transformation network for point cloud completion," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Sep. 2021, pp. 3642–3646.

[27] X. Zhang et al., "View-guided point cloud completion," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 15885–15894.

[28] E. Aiello, D. Valsesia, and E. Magli, "Cross-modal learning for image-guided point cloud shape completion," in *Proc. Adv. Neural Inf. Process. Syst.*, Dec. 2022, pp. 37349–37362.

[29] Z. Zhu et al., "CSDN: Cross-modal shape-transfer dual-refinement network for point cloud completion," *IEEE Trans. Vis. Comput. Graphics*, vol. 1, no. 1, pp. 1–18, Jan. 2023.

[30] Z. Zhu, H. Chen, X. He, W. Wang, J. Qin, and M. Wei, "SVDFormer: Complementing point cloud via self-view augmentation and self-structure dual-generator," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2023, pp. 14508–14518.

[31] Z. Li, F. Liu, W. Yang, S. Peng, and J. Zhou, "A survey of convolutional neural networks: Analysis, applications, and prospects," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 33, no. 12, pp. 6999–7019, Dec. 2022.

[32] H. Zhou et al., "Seedformer: Patch seeds based point cloud completion with upsample transformer," in *Proc. Eur. Conf. Comput. Vis.*, Nov. 2022, pp. 416–432.

[33] R. Cui et al., "P2C: Self-supervised point cloud completion from single partial clouds," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2023, pp. 14351–14360.

[34] N. J. Mitra, L. J. Guibas, and M. Pauly, "Partial and approximate symmetry detection for 3D geometry," *ACM Trans. Graph.*, vol. 25, no. 3, pp. 560–568, Jul. 2006.

[35] N. J. Mitra, M. Pauly, M. Wand, and D. Ceylan, "Symmetry in 3D geometry: Extraction and applications," *Comput. Graph. Forum*, vol. 32, no. 6, pp. 1–23, Feb. 2013.

[36] S. Lee, G. Wolberg, and S. Y. Shin, "Scattered data interpolation with multilevel B-splines," *IEEE Trans. Vis. Comput. Graphics*, vol. 3, no. 3, pp. 228–244, Jul. 1997.

[37] J. R. Price and M. H. Hayes, "Resampling and reconstruction with fractal interpolation functions," *IEEE Signal Process. Lett.*, vol. 5, no. 9, pp. 228–230, Sep. 1998.

[38] M. Kazhdan, M. Bolitho, and H. Hoppe, "Poisson surface reconstruction," in *Proc. 4th Eurographics Symp. Geometry Process.*, vol. 7, 2006, pp. 1–17.

[39] M. Kazhdan and H. Hoppe, "Screened Poisson surface reconstruction," *ACM Trans. Graph.*, vol. 32, no. 3, pp. 1–13, Jun. 2013.

[40] C.-H. Shen, H. Fu, K. Chen, and S.-M. Hu, "Structure recovery by part assembly," *ACM Trans. Graph.*, vol. 31, no. 6, pp. 1–11, Nov. 2012.

[41] Y. Li, A. Dai, L. Guibas, and M. Niesner, "Database-assisted object retrieval for real-time 3D reconstruction," in *Proc. Comput. Graph. Forum*, Feb. 2015, vol. 34, no. 2, pp. 435–446.

[42] C. R. Qi, H. Su, K. Mo, and L. J. Guibas, "PointNet: Deep learning on point sets for 3D classification and segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 652–660.

[43] C. Ma et al., "Collaborative completion and segmentation for partial point clouds with outliers," *IEEE Trans. Vis. Comput. Graphics*, vol. 1, no. 1, pp. 1–13, Oct. 2024, doi: 10.1109/TVCG.2023.3328354.

[44] X. Zhao, B. Zhang, J. Wu, R. Hu, and T. Komura, "Relationship-based point cloud completion," *IEEE Trans. Vis. Comput. Graph.*, vol. 28, no. 12, pp. 4940–4950, Dec. 2022.

[45] K. W. Tesema, L. Hill, M. W. Jones, M. I. Ahmad, and G. K. L. Tam, "Point cloud completion: A survey," *IEEE Trans. Vis. Comput. Graphics*, vol. 1, no. 1, pp. 1–20, Dec. 2024, doi: 10.1109/TVCG.2023.3344935.

[46] Y. Yang, C. Feng, Y. Shen, and D. Tian, "FoldingNet: Point cloud auto-encoder via deep grid deformation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 206–215.

[47] M. Akmal Butt and P. Maragos, "Optimum design of chamfer distance transforms," *IEEE Trans. Image Process.*, vol. 7, no. 10, pp. 1477–1484, Sep. 1998.

[48] Y. Rubner, C. Tomasi, and L. J. Guibas, "The Earth Mover's distance as a metric for image retrieval," *Int. J. Comput. Vis.*, vol. 40, no. 2, pp. 99–121, Nov. 2000.

[49] K. Han et al., "A survey on vision transformer," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 45, no. 1, pp. 87–110, Jan. 2023, doi: 10.1109/TPAMI.2022.3152247.

[50] H. Zhao, L. Jiang, J. Jia, P. Torr, and V. Koltun, "Point transformer," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 16259–16268.

[51] C. Park, Y. Jeong, M. Cho, and J. Park, "Fast point transformer," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 16949–16958.

[52] C.-K. Yang, M.-H. Chen, Y.-Y. Chuang, and Y.-Y. Lin, "2D-3D interlaced transformer for point cloud segmentation with scene-level supervision," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2023, pp. 977–987.

[53] A. Ando, S. Gidaris, A. Bursuc, G. Puy, A. Boulch, and R. Marlet, "RangeViT: Towards vision transformers for 3D semantic segmentation in autonomous driving," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2023, pp. 5240–5250.

[54] Z. Fu et al., "VAPCNet: Viewpoint-aware 3D point cloud completion," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2023, pp. 12108–12118.

[55] B. Zhang, X. Zhao, H. Wang, and R. Hu, "Shape completion with points in the shadow," in *Proc. SIGGRAPH Asia Conf. Papers*, Nov. 2022, pp. 1–9.

[56] F. Liu et al., "CloudMix: Dual mixup consistency for unpaired point cloud completion," *IEEE Trans. Vis. Comput. Graphics*, vol. 2, no. 1, pp. 1–14, Oct. 2024, doi: 10.1109/TVCG.2024.3383434.

[57] X. Li et al., "Q-diffusion: Quantizing diffusion models," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2023, pp. 17535–17545.

[58] S. Gu et al., "Vector quantized diffusion model for text-to-image synthesis," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 10696–10706.

[59] R. Rombach, A. Blattmann, D. Lorenz, P. Esser, and B. Ommer, "High-resolution image synthesis with latent diffusion models," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Aug. 2022, pp. 10684–10695.

[60] S. Luo and W. Hu, "Diffusion probabilistic models for 3D point cloud generation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 2836–2844.

[61] G. K. Nakayama, M. Angelina Uy, J. Huang, S.-M. Hu, K. Li, and L. Guibas, "DiffFacto: Controllable part-based 3D point cloud generation with cross diffusion," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2023, pp. 14257–14267.

[62] S. Chen, G. Daras, and A. Dimakis, "Restoration-degradation beyond linear diffusions: A non-asymptotic analysis for ddim-type samplers," in *Proc. Int. Conf. Mach. Learn. (ICML)*, Jul. 2023, pp. 4462–4484.

[63] C. Meng et al., "On distillation of guided diffusion models," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2023, pp. 14297–14306.

[64] Z. Lyu, J. Wang, Y. An, Y. Zhang, D. Lin, and B. Dai, "Controllable mesh generation through sparse latent point diffusion models," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2023, pp. 271–280.

[65] C. R. Qi, L. Yi, H. Su, and L. J. Guibas, "PointNet++: Deep hierarchical feature learning on point sets in a metric space," in *Proc. Annu. Conf. Neural Inf. Process. Syst. (NeurIPS)*, 2017, pp. 5099–5108.

[66] M. Simonovsky and N. Komodakis, "Dynamic edge-conditioned filters in convolutional neural networks on graphs," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 3693–3702.

[67] X. Ma, C. Qin, H. You, H. Ran, and Y. Fu, "Rethinking network design and local geometry in point cloud: A simple residual MLP framework," 2022, *arXiv:2202.07123*.

[68] W. Wu, L. Fuxin, and Q. Shan, "PointConvFormer: Revenge of the point-based convolution," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2023, pp. 21802–21813.

[69] Y. Chen, J. Liu, X. Zhang, X. Qi, and J. Jia, "LargeKernel3D: Scaling up kernels in 3D sparse CNNs," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2023, pp. 13488–13498.

[70] J. Zhang, H. Zhao, A. Yao, Y. Chen, L. Zhang, and H. Liao, "Efficient semantic scene completion network with spatial group convolution," in *Proc. IEEE Eur. Conf. Comput. Vis. (ECCV)*, Sep. 2018, pp. 733–749.

[71] A. Adamyan and E. Harutyunyan, "Smaller3D: Smaller models for 3D semantic segmentation using Minkowski engine and knowledge distillation methods," 2023, *arXiv:2305.03188*.

[72] R. Zhang, L. Wang, Y. Qiao, P. Gao, and H. Li, "Learning 3D representations from 2D pre-trained models via image-to-point masked autoencoders," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2023, pp. 21769–21780.

[73] 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 Trans. Graph.*, vol. 38, no. 5, pp. 1–12, 2019.

[74] J. Zhang et al., "Unsupervised 3D shape completion through GAN inversion," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 1768–1777.

[75] B. Ma, Z. Han, Y. Liu, and M. Zwicker, "Neural-Pull: Learning signed distance function from point clouds by learning to pull space onto surface," in *Proc. Int. Conf. Mach. Learn. (ICML)*, Jul. 2021, pp. 7246–7257.

[76] G. Chou, I. Chugunov, and F. Heide, "GenSDF: Two-stage learning of generalizable signed distance functions," in *Proc. Adv. Neural Inf. Process. Syst.*, Dec. 2022, pp. 24905–24919.

[77] A. X. Chang et al., "ShapeNet: An information-rich 3D model repository," 2015, *arXiv:1512.03012*.

[78] L. Mescheder, M. Oechsle, M. Niemeyer, S. Nowozin, and A. Geiger, "Occupancy networks: Learning 3D reconstruction in function space," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2019, pp. 4460–4470.

[79] M. Tatarchenko, S. R. Richter, R. Ranftl, Z. Li, V. Koltun, and T. Brox, "What do single-view 3D reconstruction networks learn?" in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 3400–3409.

[80] P. Xiang et al., "SnowflakeNet: Point cloud completion by snowflake point deconvolution with skip-transformer," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Aug. 2021, pp. 5499–5509.

[81] X. Wen et al., "PMP-net: Point cloud completion by transformer-enhanced multi-step point moving paths," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 45, no. 1, pp. 852–867, Jan. 2023.

[82] S. Li, P. Gao, X. Tan, and M. Wei, "ProxyFormer: Proxy alignment assisted point cloud completion with missing part sensitive transformer," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2023, pp. 9466–9475.

Haihong Xiao received the M.S. degree from Nanjing Agricultural University, Nanjing, China, in 2021. He is currently pursuing the Ph.D. degree with South China University of Technology. His research interests include 3D vision, point cloud processing, and scene representation learning.

Ying He (Member, IEEE) received the B.S. and M.S. degrees in electrical engineering from Tsinghua University, China, and the Ph.D. degree in computer science from Stony Brook University, USA. He is currently an Associate Professor with the School of Computer Science and Engineering, Nanyang Technological University, Singapore. His research interests include visual computing, particularly in the problems which require geometric analysis and computation. He served as an Associate Editor for IEEE TRANSACTIONS ON VISUALIZATION AND COMPUTER GRAPHICS, *Computer Graphics Forum*, and *Journal of Computational Visual Media*.

Hao Liu received the B.E. degree from the University of Electronic Science and Technology of China in 2016, the M.E. degree from the National University of Defense Technology in 2018, and the Ph.D. degree from Sun Yat-sen University in 2023. He is currently a Research Fellow with the School of Computer Science and Engineering, Nanyang Technological University, Singapore. His research interests include 3D deep learning and NeRF, particularly in 3D object detection and multi-object tracking.

Wenxiong Kang (Member, IEEE) received the Ph.D. degree from South China University of Technology, Guangzhou, China, in 2009. He is currently a Professor with the School of Automation Science and Engineering, South China University of Technology. His research interests include computer vision, biometrics identification, image processing, and pattern recognition.

Yuqiong Li received the Ph.D. degree from Beijing Institute of Technology, Beijing, China, in 2010. He is currently a Senior Researcher with the Key Laboratory for Mechanics in Fluid Solid Coupling Systems, Institute of Mechanics, Chinese Academy of Sciences. His research interests include vehicle-terra mechanics, in-situ mechanical survey of lunar soil, artificial intelligence, and machine learning.