Pattern Recognition 180 (2026) 114230

Contents lists available at ScienceDirect

Pattern Recognition

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

RDS-Net: Recursive structure refinement network with Dual-awareness Shape transfer for point cloud completion

Xiaocui Li [a,b], Weili Chen [a], Xinyu Zhang [a,b,*], Yangtao Wang [c], Wei Liang [a,b], Keqin Li [d]

[a] Hunan University of Technology and Business, Changsha, 410205, Hunan, China
[b] Xiangjiang Laboratory, Changsha, 410205, Hunan, China
[c] Guangzhou University, Guangzhou, 510006, Guangdong, China
[d] State University of New York, NY, United States of America

ARTICLE INFO

Keywords:
Point cloud completion
Skip Set Abstraction
Cross-scale Recursive Expansion
Dual-awareness upTransformer

ABSTRACT

Emerging applications like autonomous driving and augmented reality demand high-quality 3D perception. To overcome incompleteness caused by self-occlusion and limited sensor resolution, point cloud completion aims to reconstruct full shapes by preserving the geometric shape information of the partial observations and inferring the missing regions. However, most existing methods face the dilemma that global features struggle to recover fine-grained details due to its nature of limited information, and the model lacks effective strategies to employ the learned shape priors to generate detailed and reasonable geometric structures. To address this issue, we propose a novel point cloud completion network called RDS-Net in this paper. RDS-Net introduces a Skip Set Abstraction module to reduce the information loss of structure details during hierarchical feature extraction, thereby preserving fine geometric features for fine-grained shape completion. Furthermore, RDS-Net devises a Dual-Awareness upTransformer to effectively transfer similar geometric structures from learned shape priors to missing regions. Building upon this module, we propose a Cross-scale Recursive Expansion module that captures structure details at different scales through recursive execution to predict precise geometric shapes, enabling more controllable and consistent shape recovery. We conduct extensive experiments on widely used benchmark datasets and the results demonstrate the effectiveness of our proposed method. By achieving a lower Chamfer Distance on datasets such as PCN and ShapeNet-55/34, RDS-Net substantially improves shape reconstruction accuracy and quality, thereby supporting high-fidelity 3D environmental perception.

1. Introduction

As an efficient 3D data representation, point cloud is widely used in practical scenarios. However, point clouds are usually highly sparse and incomplete due to issues such as self-occlusion and limited sensor resolution during acquisition, which hinders further applications. Therefore, completing point clouds is an indispensable step for various downstream tasks [1,2].

In recent years, deep learning-based completion methods have gradually become the mainstream, among which many studies [3,4] follow a common encoder–decoder paradigm in which the encoder learns a global shape representation from the partial input and the decoder further reconstructs the complete shape from this representation. However, such completion paradigms often incur the loss of structure details during feature extraction, caused by aggregation operations like max pooling. This makes it difficult to provide effective regional information for fine-grained recovery.

Apart from the feature extraction challenge, another common problem lies in the design of point generators. Most existing methods rely on folding or hierarchical folding [3,5] to map 2D grids onto 3D surfaces, while some adopt deconvolution for upsampling. The core limitation of these methods is their tendency to process each point independently, resulting in poor detail recovery. Although SeedFormer [6] improves refinement quality through local aggregation, it still struggles to fully exploit the potential geometric information to recover complex missing regions.

To address the aforementioned issues, we propose a **R**ecursive structure refinement network with **D**ual-awareness **S**hape transfer for point cloud completion(**RDS-Net**). Fig. 1 illustrates the overall pipeline of our method for completing partial inputs. Here, Seed represents the point cloud (with 256 points) obtained during the seed generation stage, which preliminarily characterizes the overall structure of the 3D shape. P1 and P2 represent point clouds with different resolutions

* Corresponding author at: Hunan University of Technology and Business, Changsha, 410205, Hunan, China.
E-mail address: zhangxinyu247@163.com (X. Zhang).

https://doi.org/10.1016/j.patcog.2026.114230
Received 20 March 2026; Received in revised form 10 May 2026; Accepted 6 June 2026
Available online 12 June 2026

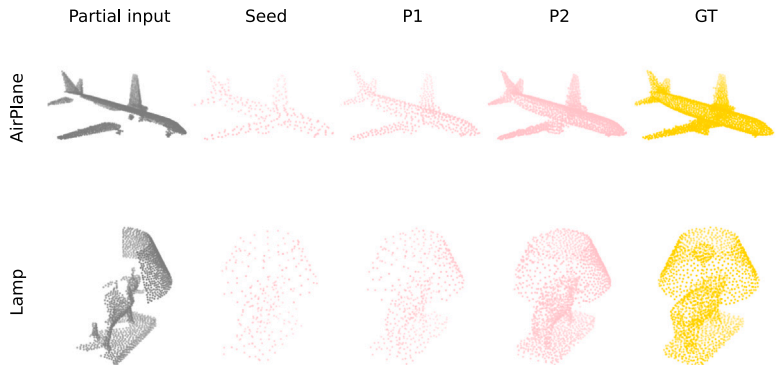

Fig. 1. The visualization process of reconstruction from coarse to fine.

(P1 has 512 points and P2 has 2048 points), which are reconstructed through several upsample blocks. The overall architecture of RDS-Net is shown in Fig. 2, where it introduces a module named Skip Set Abstraction (SSA) to preserve more structure details and then encodes the partial input through a hierarchical encoder based on SSA, yielding fine geometric features that represent structures of the target shape. Next, it focuses on the detail recovery process from the partial input, employing a seed generator to predict a seed point cloud that represents the overall structure of the complete shape. To refine the shape accurately, RDS-Net devises a Dual-Awareness upTransformer (DAT) to better capture spatial and semantic dependencies among points, enabling the model to transfer similar geometric structures from learned shape priors to missing regions. Finally, to achieve a more controllable and consistent structure refinement process, DAT is applied recursively to construct a Cross-scale Recursive Expansion (CRE) module, which is then integrated into the decoder to enable hierarchical refinement. Extensive experiments show that RDS-Net outperforms the state-of-the-art approaches on several widely used benchmark datasets. Our main contributions can be summarized as follows:

(1) We propose Skip Set Abstraction (SSA) as a novel feature extraction module. By introducing skip-connection to reduce detail loss during the aggregation, SSA can encode fine geometric features to further capture more effective local and global structures, laying the foundation for fine-grained reconstruction in subsequent stages.

(2) We propose a Dual-Awareness upTransformer (DAT) that enhances the ability to capture effective structure details of local regions. With the assistance of DAT, our model can transfer the contributing geometric structures to missing regions according to the shape priors learned in the previous layer, thereby further improving the quality of detail recovery.

(3) We design a Cross-scale Recursive Expansion module (CRE) based on DAT to propagate multi-scale regional information in a recursive manner, which helps to recover geometric structure details at different scales, achieving more controllable and consistent shape reconstruction. Moreover, experimental evaluations on several datasets demonstrate the superiority of RDS-Net over other existing methods.

2. Related work

2.1. Point cloud representation

Early works [7,8] often adopt voxel grids as intermediate representations to describe 3D shapes, and then apply 3D CNNs for processing. However, such methods inevitably incur irreversible information loss and computational resource waste. Therefore, Xie et al. [9] propose to convert point clouds into 3D meshes, with the advantage of preserving geometric details without loss. Unlike these methods, the pioneering PointNet [10] directly processes raw point clouds by applying shared MLPs to each point independently and aggregating features through pooling operations to achieve permutation invariance. Inspired by this, Yuan et al. [3] propose the first learning-based completion method PCN. To address PointNet's inability to capture local geometric details, PointNet++[11] is proposed to learn hierarchical local features using Set Abstraction (SA). This strategy has proven highly effective and has since been widely adopted by various point cloud completion methods.

Driven by these feature learning strategies, most early methods [5,12,13] tend to decode aggregated global shape representations to reconstruct complete shapes, but the loss of geometric details caused by aggregation operations makes it difficult for the network to recover fine-grained structures. To predict detailed shapes, many studies [6,14,15] focus on the process of recovering details from partial inputs. For example, SeedFormer [6] introduces patch seeds as latent representations to preserve regional information of local patterns, while AnchorFormer [14] learns a set of anchors to represent geometric patterns. Based on previous methods, we propose to learn finer geometric features via SSA, which revisits previous features through the structure of skip-connection [5,16] to enrich the context of local features. Meanwhile, we employ a seed generator to reconstruct coarse but complete seed points, providing effective regional information for the point refinement stage.

2.2. Point feature generation

Most existing methods [3,4] expand features through the folding operation, which is designed to upsample each point independently with fixed 2D variants. However, the folding operation usually ignores semantic relationships between points, resulting in poor detail recovery. Subsequently, some works [5,16] improve the folding operation and introduce hierarchical folding to preserve the complete shapes at different resolutions. AnchorFormer [14] proposes a point morphing scheme to control the 2D grid deformation at the location of each sparse point. CarvingNet [17] designs FoldingNet++ to operate each feature vector, generating a dense point cloud with rich details. In addition to folding-based methods, deconvolution is also commonly used in feature expansion, as exemplified by CRA-PCN [18], which adopts this operation to generate high-resolution point displacements to recover structure details.

Recently, more and more studies have applied Transformers to point generation, which converts the entire point set or local region into a sequence and then aggregates features through the attention mechanism. For example, SeedFormer [6] extends the Transformer architecture into the decoding phase, forming a new pattern of point generation with local aggregation. In contrast, PointAttN [19] employs multi-head self-attention and cross-attention to implicitly capture both local and global geometric contexts to predict new points. To further mitigate global context loss in windowed attention, SDA-Net [20] leverages point serialization and dual-attention mechanisms to comprehensively model spatial and channel-wise dependencies. Combining previous experience, we introduce DAT to achieve a more efficient strategy of point generation, which propagates the previous point displacements to current points through dual-awareness mechanism and expands point features by deconvolution.

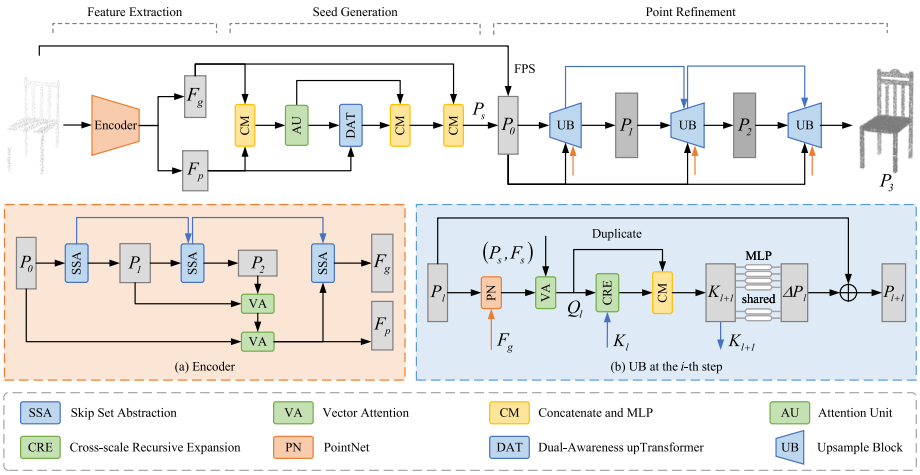

Fig. 2. The overall architecture of RDS-Net, comprising three modules: (1) Feature Extraction, which employs three SSA layers to learn local and global features; (2) Seed Generation, which predicts a coarse but complete point cloud for providing regional information to the upsample blocks at each level; (3) Point Refinement, which progressively refines the point cloud through multiple upsample blocks.

2.3. Point cloud completion

It is crucial to guide the orderly generation of point clouds through relevant constraints during the completion process. To achieve this goal, early methods [13,19] often introduce global features into the generation phase to maintain the global shape consistency. LGP-Net [21] proposes inter-scale feature consistency to maintain the geometric consistency of the predicted point clouds. The latest methods model the completion as a structured generation process through cross-space feature transfer. For example, SnowflakeNet [12] introduces skip-transformer into the decoding process, employing the attention mechanism to summarize the splitting patterns used in the previous SPD layer, thereby learning the most suitable point splitting patterns for local regions. CRA-PCN [18] designs a Cross-Resolution Transformer to aggregate useful features from the support point clouds generated in the previous stage. Furthermore, DcTr [22] is apt at using local features and preserving a well-structured generation process. SPAC-Net [23] introduces a structural prior termed "interface" to localize the boundary between observed and missing regions, guiding coarse shape prediction to alleviate detail loss from feature abstraction.

Additionally, some methods [24,25] model point clouds through multi-stage generation to improve performance, making the completion process more interpretable and easier to control. Inspired by this strategy, we further propose CRE that introduces recursive execution as a multi-stage approach into a single operation of point generation to recover geometric details at different scales. It effectively utilizes and preserves reasonable structure details in local regions, thereby ensuring geometric consistency and enabling finer shape reconstruction.

3. Proposed method

3.1. Overview

The overall architecture of RDS-Net is shown in Fig. 2, consisting of three modules: Feature Extraction, Seed Generation, and Point Refinement. We will introduce each module in detail as follows.

3.1.1. Feature extraction

Given a partial input $\mathcal{P} = \{p_i\}_{i=1}^{N} \in \mathbb{R}^{N\times 3}$, most methods extract hierarchical features by stacking multiple layers of standard modules such as Set Abstraction (SA) [11], EdgeConv [26], or Point Transformer [27]. However, during such a hierarchical encoding process, structure details are inevitably lost due to aggregation operations like max pooling, making it difficult for the network to extract fine geometric features.

To address this issue, we devise a Skip Set Abstraction (SSA) by extending the skip-connection structure into SA, which transfers neighborhood features at different scales as supporting information between adjacent layers to preserve more geometric details. Specifically, at step t, the aggregated features $\mathcal{F}_t^a \in \mathbb{R}^{N_t\times C_t}$ are fed into SSA as the primary input, while the corresponding neighborhood features $\mathcal{F}_t^n \in \mathbb{R}^{N_t\times C_t\times k}$ (k is the neighborhood size, set to 16) are provided as supporting information. Note that $\mathcal{F}_t^n$ is obtained only by grouping and non-linear projection (without undergoing max pooling for aggregation) to preserve rich local structure details of $\mathcal{F}_t^a$. The neighborhood size k is empirically set to 16, which is a common choice in point cloud completion works such as PointNet++[11] and SnowflakeNet [12]. This value enables sufficient capture of local geometric context while maintaining reasonable computational efficiency. Then SSA performs feature extraction as:

$$\mathcal{F}_{t+1}^a, \mathcal{F}_{t+1}^n = SA\left(SC\left(\mathcal{F}_t^a, \mathcal{F}_t^n\right)\right), \tag{1}$$

where $SC(\cdot)$ denotes the skip-connection to reduce the information loss, defined as:

$$\alpha = \sigma\left(\alpha_{param}\right) \tag{2}$$

$$SC\left(\mathcal{F}_t^a, \mathcal{F}_t^n\right) = MLP\left(Max\left(Cat\left(\alpha\cdot\lambda\left(\mathcal{F}_t^a\right), (1-\alpha)\cdot\mathcal{F}_t^n\right)\right)\right). \tag{3}$$

Here, $\lambda(\cdot)$ is a non-linear projection function implemented with multilayer perceptron (MLP), $Cat(\cdot)$ denotes the feature concatenation, and $Max(\cdot)$ denotes the max pooling to adjust the shape of features. The learnable parameter α_{param} is initialized uniformly in the range $[0.2, 0.8]$ to balance the contributions between $\mathcal{F}_t^a$ and neighborhood features during early training. During back-propagation, α_{param} is updated end-to-end together with other network parameters using the Adam optimizer, and its value is activated by the sigmoid function $\sigma(\cdot)$ to constrain the final α within $(0, 1)$. A larger α (close to 1) emphasizes the aggregated features for global structure perception, while a smaller α (close to 0) strengthens the neighborhood details to reduce structure loss. This adaptive adjustment mechanism enables SSA to preserve fine-grained geometric features flexibly. For the first SSA layer, where no explicit neighborhood features are available, we set $\mathcal{F}_t^n = \mathcal{F}_t^a$ and transfer $Max(\cdot)$ to $\lambda\left(\mathcal{F}_t^a\right)$ for better effect. In both cases, the skip-connection ensures that local structure details are well introduced. We present the overall pipeline of SSA in Algorithm 1.

As shown in Fig. 2(a), the feature extraction pipeline consists of three SSA layers that hierarchically aggregate features to capture both

Algorithm 1 Skip Set Abstraction

Input: Aggregated features $\mathcal{F}_t^a \in \mathbb{R}^{N_t \times C_t}$, optional neighborhood features $\mathcal{F}_t^n \in \mathbb{R}^{N_t \times C_t \times k}$, point coordinates $\mathcal{P}_t \in \mathbb{R}^{N_t \times 3}$, and learnable parameter α_{param}.
Output: Updated aggregated features $\mathcal{F}_{t+1}^a \in \mathbb{R}^{N_{t+1} \times C_{t+1}}$, updated neighborhood features $\mathcal{F}_{t+1}^n \in \mathbb{R}^{N_{t+1} \times C_{t+1} \times k}$, and new sampling point coordinates $\mathcal{P}_{t+1} \in \mathbb{R}^{N_{t+1} \times 3}$.
Initialize $\alpha \leftarrow \sigma(\alpha_{param})$, where σ is the sigmoid function.
Compute projected features: $\mathcal{F}_t^p \leftarrow \lambda(\mathcal{F}_t^a)$.
/ Introduce local structure details via skip-connection */*
if $\mathcal{F}_t^n ==$ None **then**
  $\mathcal{F}_t^{glb} \leftarrow \max_{i=1,\dots,N_t} \mathcal{F}_t^p[:,:,i]$ expand to (C_t, N_t)
  $\mathcal{F}_t^{fuse} \leftarrow \mathrm{MLP}_{fuse}([\alpha \cdot \mathcal{F}_t^a, (1-\alpha) \cdot \mathcal{F}_t^{glb}])$
else
  $\mathcal{F}_t^{glb} \leftarrow \mathcal{F}_t^p$ expand to (C_t, N_t, k)
  $\mathcal{F}_t^{fuse} \leftarrow \mathrm{MLP}_{fuse}\left(\max_{i=1,\dots,k}\left([\alpha \cdot \mathcal{F}_t^n, (1-\alpha) \cdot \mathcal{F}_t^{glb}]\right)[:,:,:,i]\right)$
end if
/ Sample and Group */*
if `group_all` **then**
  $\mathcal{P}_{t+1}, \mathcal{F}_{t+1}^{grouped} \leftarrow \mathrm{SampleAndGroupALL}(\mathcal{P}_t, \mathcal{F}_t^{fuse})$
else
  $\mathcal{P}_{t+1}, \mathcal{F}_{t+1}^{grouped} \leftarrow \mathrm{SampleAndGroupKNN}(\mathcal{P}_t, \mathcal{F}_t^{fuse})$
end if
/ Map grouped features to a high-dimensional space */*
$\mathcal{F}_{t+1}^{conv} \leftarrow \mathrm{MLP}_{conv}(\mathcal{F}_{t+1}^{grouped})$
/ Update aggregated features by fusing average and max pooling */*
$\mathcal{F}_{t+1}^a \leftarrow \mathrm{MLP}_{agg}\left(\frac{1}{k}\sum_{i=1}^{k} \mathcal{F}_{t+1}^{conv}[:,:,:,i],\ \max_{i=1,\dots,k} \mathcal{F}_{t+1}^{conv}[:,:,:,i]\right)$
Update neighborhood features: $\mathcal{F}_{t+1}^n \leftarrow \mathcal{F}_{t+1}^{conv}$
return $\mathcal{P}_{t+1}, \mathcal{F}_{t+1}^a, \mathcal{F}_{t+1}^n$

local and global representations. However, directly stacking multiple SSA layers is insufficient to preserve fine-grained structure details, as repeated downsampling and aggregation often lead to the loss of high-frequency geometric patterns. Therefore, we incorporate Vector Attention (VA) [18] to extract useful geometric information from the intermediate point clouds output from the previous SSA layers, enhancing the model's ability to understand intricate shapes. Finally, we obtain the shape vector $\mathcal{F}_g \in \mathbb{R}^C$, representing the global structure of the partial input, as well as a set of downsampled partial points $\mathcal{P}_p = \{p_i\}_{i=1}^{N_p} \in \mathbb{R}^{N_p \times 3}$ and corresponding features $\mathcal{F}_p = \{f_i\}_{i=1}^{N_p} \in \mathbb{R}^{N_p \times C_p}$ to represent local structures.

3.1.2. Seed generation

This stage aims to predict a coarse yet complete point cloud named the seed point cloud, which captures the overall structure of the complete shape and possesses low-resolution coordinates and feature representations. As shown in Fig. 2, we concatenate $\mathcal{F}_g$ with each point feature in $\mathcal{F}_p$ and apply a shared MLP to obtain enhanced features, which are then fed into a self-attention unit to aggregate cross-region geometric information, thus forming the sparse queries $\mathcal{F}_q$. Next, both $\mathcal{F}_q$ and $\mathcal{F}_p$ are fed into our proposed Dual-Awareness upTransformer (See Section 3.2 for details) to obtain the upsampled features $\mathcal{F}_u \in \mathbb{R}^{2N_p \times C_p}$:

$$\mathcal{F}_u = DAT\left(\mathcal{F}_q, \mathcal{F}_p\right), \mathcal{F}_q = AU\left(MLP\left(Cat\left(\mathcal{F}_p, \mathcal{F}_g\right)\right)\right), \tag{4}$$

where $AU(\cdot)$ and $DAT(\cdot)$ denote the attention unit and Dual-Awareness upTransformer, respectively. Then, we integrate $\mathcal{F}_u$ with $\mathcal{F}_q$ and $\mathcal{F}_g$ through two successive MLPs to generate a set of seed features $\mathcal{F}_s \in \mathbb{R}^{N_s \times C_s}$. In this process, $\mathcal{F}_u$ serves to add variations, while $\mathcal{F}_q$ and $\mathcal{F}_g$ provide the shape context and global geometric constraints, respectively. Finally, we apply a shared MLP on $\mathcal{F}_s$ to regress the corresponding point coordinates $\mathcal{P}_s \in \mathbb{R}^{N_s \times 3}$:

$$\mathcal{P}_s = MLP\left(\mathcal{F}_s\right), \mathcal{F}_s = MLP\left(Cat\left(MLP\left(Cat\left(\mathcal{F}_u, \mathcal{F}_q\right)\right), \mathcal{F}_g\right)\right). \tag{5}$$

For better optimization, following previous methods [6,12], we concatenate the partial input $\mathcal{P}$ with seed points $\mathcal{P}_s$ and downsample the combined set with Farthest Point Sampling (FPS) to obtain the initial point cloud $\mathcal{P}_0 \in \mathbb{R}^{N_0 \times 3}$, which preserves the partial structure of the original input.

3.1.3. Point refinement

RDS-Net follows the coarse-to-fine generation strategy [6,12,28] to progressively recover faithful geometric structures of objects. In detail, given the initial point cloud $\mathcal{P}_0$, we gradually produce $\mathcal{P}_1, \mathcal{P}_2$, and $\mathcal{P}_3$ through a hierarchical structure composed of multiple upsample blocks. Each upsample block takes the previous output $\mathcal{P}_l \in \mathbb{R}^{N_l \times 3}$ $(l = 0, 1, 2)$ as input and simultaneously introduces $\mathcal{F}_g, \mathcal{F}_s$ as global geometric constraints and local region information, respectively. It also utilizes parent point displacements to control the structured generation of new points. Through deep information interaction, the upsample block generates a denser point cloud $\mathcal{P}_{l+1} \in \mathbb{R}^{N_{l+1} \times 3}$ with an upsampling rate of r_l, where $N_{l+1} = N_l \times r_l$, corresponding to one step in the hierarchical generation process.

This process not only achieves gradual growth in point count, but also preserves and enhances local geometric information at each step through a feature propagation mechanism, ensuring the coherence and rationality of the completion results at both the overall structure and detail levels (more details are discussed in Section 3.2).

3.2. Upsample block

As shown in Fig. 2(b), given the input point cloud $\mathcal{P}_l \in \mathbb{R}^{N_l \times 3}$, we first adopt a mini-PointNet [10] to learn per-point features $\mathcal{F}_l \in \mathbb{R}^{N_l \times D}$. Then, we apply VA [18] on $\mathcal{F}_l$ to extract structural information from seed features $\mathcal{F}_s$ to generate point-wise queries $\mathcal{Q}_l = \{q_i^l\}_{i=1}^{N_l}$. Following [12], we use the displacement features learned in the previous layer as keys $\mathcal{K}_l = \{k_i^l\}_{i=1}^{N_l}$, which serve to preserve the learned features of the existing input points. Then both $\mathcal{Q}_l$ and $\mathcal{K}_l$ are fed into Cross-scale Recursive Expansion (CRE) to obtain a new set of features $\mathcal{H}_l = \{h_i^l\}_{i=1}^{r_l N_l}$.

To preserve the original context, $\mathcal{H}_l$ and $\mathcal{Q}_l$ are integrated through a shared MLP, forming new displacement features $\mathcal{K}_{l+1} = \{k_i^{l+1}\}_{i=1}^{N_{l+1}}$, $N_{l+1} = N_l \times r_l$, which are then fed to the next layer via skip-connection to participate in the expansion process. Finally, we apply MLP on $\mathcal{K}_{l+1}$ to obtain a set of point displacement offsets $\Delta\mathcal{P}_l$, which are added to $\mathcal{P}_l$ to form $\mathcal{P}_{l+1}$. This operation is defined as $\mathcal{P}_{l+1} = \widehat{\mathcal{P}}_l + \Delta\mathcal{P}_l$, where $\widehat{\mathcal{P}}_l$ is the duplicated point cloud.

3.2.1. Cross-scale recursive expansion

As illustrated in Fig. 3, RDS-Net proposes a module named Cross-scale Recursive Extension that captures multi-scale geometric relationships to recover fine-grained structure details. Given the input queries $\mathcal{Q}_l \in \mathbb{R}^{N_l \times D}$ and the keys $\mathcal{K}_l \in \mathbb{R}^{N_l \times D}$, which share the same geometric coordinates $\mathcal{P}_l \in \mathbb{R}^{N_l \times D}$, we first adopt FPS to hierarchically downsample them to obtain subsets $\{\mathcal{P}_l^m\}_{m=0}^{M-1}$, $\{\mathcal{Q}_l^m\}_{m=0}^{M-1}$, and $\{\mathcal{K}_l^m\}_{m=0}^{M-1}$, where $M = 3$ denotes the number of scales. Then, we aggregate and upsample features through the Dual-Awareness upTransformer (DAT) on the $(M-1)$-th scale:

$$\mathcal{H}_l^m = DAT\left(\mathcal{P}_l^m, \mathcal{Q}_l^m, \mathcal{K}_l^m, \mathcal{V}_l^m\right), m = M - 1, \tag{6}$$

where $\mathcal{Q}_l^m$ and $\mathcal{K}_l^m$ are queries and keys on the mth scale, respectively. The values $\mathcal{V}_l^m$ are obtained by applying an MLP on concatenated $\mathcal{Q}_l^m$ and $\mathcal{K}_l^m$.

Next, we concatenate $\mathcal{H}_l^m$, $\mathcal{Q}_l^{m-1}$, and $\mathcal{K}_l^{m-1}$, then fuse them by MLP to obtain new values $\mathcal{V}_l^{m-1}$, which allows the module to continue executing DAT on the $(M-2)$-th scale. Finally, the process is iterated until we obtain $\mathcal{H}_l^0$, which is then concatenated with the pooling results of each scale output and fused by MLP to obtain a set of upsampled point features. We present the overall pipeline of CRE in Algorithm 2.

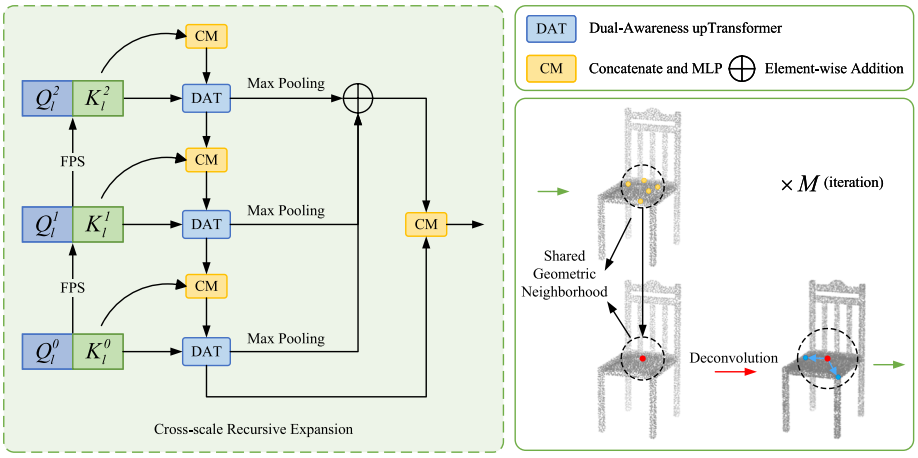

Fig. 3. Cross-scale Recursive Expansion, which considers iterating on M scales to recover detailed geometric structures.

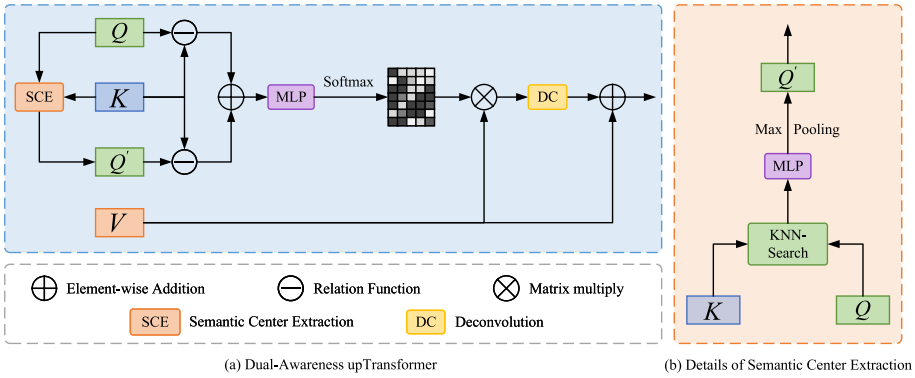

Fig. 4. (a) Dual-Awareness upTransformer, which captures feature relationships through dual awareness mechanism and generates new points using deconvolution; (b) Semantic Center Extraction, which serves to extract virtual queries.

3.2.2. *Dual-awareness uptransformer*

Fig. 4(a) depicts the detailed structure of the Dual-Awareness upTransformer ("dual-awareness" represents the capture of both spatial and semantic relationships). Given the query $\mathcal{Q}^m$, key $\mathcal{K}^m$, and value $\mathcal{V}^m$ (to avoid notational clutter, we omit the layer index l and retain only the scale index m), we generate virtual queries using the Semantic Center Extraction module(whose structure is shown in Fig. 4(b)). Specifically, for each query point q_i^m, we identify its k nearest neighbors in $\mathcal{K}^m$ based on semantic similarity. These neighboring points are first projected through an MLP, followed by the max pooling to produce semantic center points $\mathcal{S}^m$ as virtual queries. Next, we construct a neighborhood for q_i^m by finding its k nearest neighbors in the geometric space. The attention relations in the neighborhood of each query point can be calculated as follows:

$$qk_{ij}^m = \beta^m\left(q_i^m\right) - \gamma^m\left(k_j^m\right), sk_{ij}^m = s_i^m - \gamma^m\left(k_j^m\right), \tag{7}$$

$$\hat{a}_{ij}^m = \alpha^m\left(qk_{ij}^m + sk_{ij}^m + \delta_{ij}^m\right), j = 1, 2, \ldots, k. \tag{8}$$

Here, α^m is a non-linear projection function implemented with MLP, while β^m, and γ^m are linear mapping functions implemented with convolution. $\delta_{ij}^m \in \mathbb{R}^D$ is the position encoding. Afterwards, we normalize the weights $\hat{a}_{ij}^m$ with the softmax function and aggregate local features:

$$a_{ij}^m = \frac{exp\left(\hat{a}_{ij}^m\right)}{\sum_{j=1}^k exp\left(\hat{a}_{ij}^m\right)}, \tag{9}$$

$$h_i^m = \sum_{j=1}^k a_{ij}^m \odot \left(\psi^m\left(v_j^m\right) + \delta_{ij}^m\right). \tag{10}$$

Here, ψ^m is a linear mapping function implemented with convolution, and $\odot$ denotes the Hadamard product. This yields the final features $\mathcal{H}^m \in \mathbb{R}^{N_m \times D}$.

Finally, $\mathcal{H}^m$ is upsampled by a factor of r_m via deconvolution and further refined through a residual connection to obtain the final output $\mathcal{Z}^m = \left\{z_i^m \mid i = 1, 2, \ldots, r_m N_m\right\} \in \mathbb{R}^{r_m N_m \times D}$, as follows:

$$\mathcal{Z}^m = DeConv\left(\mathcal{H}^m\right) \oplus Upsample\left(\mathcal{V}^m\right). \tag{11}$$

Here, $DeConv(\cdot)$ and $\oplus$ denote the deconvolution and element-wise addition, respectively, and $Upsample(\cdot)$ denotes the upsampling operation realized by interpolation.

3.3. *Loss function*

To better train and optimize the model, this method introduces a multi-resolution supervision mechanism and constructs a multi-scale loss function. Specifically, the ground truth $\mathcal{P}_t$ is first downsampled using FPS to generate subsets of different resolutions to match the number of points in each predicted set $\mathcal{P}_s, \mathcal{P}_1, \mathcal{P}_2, \mathcal{P}_3$. Subsequently, the L1 Chamfer Distance (CD) between the predicted point cloud and the corresponding ground truth is calculated at each output scale. Finally, the total loss is defined as the sum of CD losses across all scales:

$$\mathcal{L} = \sum_{i \in \{s,1,2,3\}} CD_{L1}\left(\mathcal{P}_i, FPS\left(\mathcal{P}_t, \left|\mathcal{P}_i\right|\right)\right). \tag{12}$$

Here, $FPS\left(\mathcal{P}_t, \left|\mathcal{P}_i\right|\right)$ denotes the downsampled ground truth with the same number of points as $\mathcal{P}_i$, and $FPS(\cdot)$ stands for the Farthest Point Sampling algorithm.

Algorithm 2 Cross-scale Recursive Expansion

Input: Query point cloud $\mathcal{Q} \in \mathbb{R}^{N\times D}$, support point cloud $\mathcal{K} \in \mathbb{R}^{N\times D}$, shared geometric coordinates $\mathcal{P} \in \mathbb{R}^{N\times 3}$, downsampling rate list $down\_rate$, and neighborhood size list $knns$.
Output: Final upsampled features $\mathcal{H}^0 \in \mathbb{R}^{rN\times D}$.
Initialize $M \leftarrow \text{len}(down\_rate)$, $\mathcal{F}_{mp} \leftarrow \emptyset$.
for $m = M-1$ **downto** 0 **do**
  /* Sample subsets via FPS */
  $idx \leftarrow \text{FPS}(\mathcal{P}, down\_rate[m])$, $\mathcal{Q}^m \leftarrow \text{gather}(\mathcal{Q}, idx)$
  $\mathcal{K}^m \leftarrow \text{gather}(\mathcal{K}, idx)$, $\mathcal{P}^m \leftarrow \text{gather}(\mathcal{P}, idx)$
  if $m == M-1$ **then**
    $\mathcal{V}^m \leftarrow \text{MLP}_{res}(\mathcal{Q}^m, \mathcal{K}^m)$
  else
    $\mathcal{V}^m \leftarrow \text{MLP}_{res}(\mathcal{Q}^m, \mathcal{K}^m, \mathcal{H}^{m+1})$
  end if
  /* Expand features with Dual-Awareness upTransformer */
  $\mathcal{H}^m \leftarrow \text{DAT}(\mathcal{P}^m, \mathcal{Q}^m, \mathcal{K}^m, \mathcal{V}^m)$
  /* Max pooling on the upsampled features */
  $\mu^m \leftarrow \max_{i=1,\dots,N_m} \mathcal{H}^m[:,:,i]$
  if $\mathcal{F}_{mp} ==$ None **then**
    /* Save current scale pooling result */
    $\mathcal{F}_{mp} \leftarrow \mu^m$
  else
    /* Accumulate pooling results */
    $\mathcal{F}_{mp} \leftarrow \mathcal{F}_{mp} + \mu^m$
  end if
end for
Fuse $\mathcal{H}^0$ and historical pooling sum: $\mathcal{H}^0 \leftarrow \text{MLP}_{cat}(\mathcal{H}^0, \mathcal{F}_{mp})$
return $\mathcal{H}^0$

4. Experiments

4.1. Datasets

PCN. The PCN dataset [3], as a subset of ShapeNet, comprises 30,974 shapes from 8 categories, including 28,974 shapes for training, 1200 shapes for testing, and 800 shapes for validation. Each shape is composed of 16,384 uniformly sampled points, and partial observations are generated by back-projecting 2.5D depth images from 8 different views into 3D.

ShapeNet-55/34. The ShapeNet-55 and ShapeNet-34 datasets [29] are also derived from ShapeNet. Specifically, ShapeNet-55 has shapes from all 55 categories, including 41,952 shapes for training and 10,518 shapes for testing. ShapeNet-34, on the other hand, uses shapes from 34 categories for training, with a training set containing 46,765 shapes. Its testing set is divided into two parts: 3400 shapes from 34 seen categories and 2305 shapes from 21 unseen categories. Following the previous evaluation strategy [29], we generate partial observations by masking out 25%, 50%, and 75% of the complete shapes, and evaluate model performance under these settings to simulate point completion tasks at three difficulty levels: simple (S), moderate (M) and hard (H).

MVP. The MVP dataset [30] is a high-quality multi-view point cloud dataset consisting of 16 categories of partial and complete point clouds generated from CAD models in ShapeNet. Specifically, it contains 62,400 shape pairs for training and 41,600 shape pairs for testing. For each complete shape, 26 partial point clouds are generated by selecting 26 camera poses which are uniformly distributed on a unit sphere. In addition, we choose the 2048-point ground truth among various resolutions provided to conduct experiments.

KITTI. KITTI [31] is a real-world dataset comprising a sequence of outdoor LiDAR scans where car objects are extracted in each frame based on provided 3D bounding boxes, resulting in a total of 2401 partial car point clouds. Unlike other synthetic datasets, the scanned data in KITTI is highly sparse and lacks complete point cloud as ground truth. To ensure fair comparison, this study follows the experimental setup of [32], directly evaluating the pre-trained model (trained on MVP) on the KITTI dataset without any fine-tuning or retraining operations.

RGB-D. The RGB-D dataset [33] comprises 300 common household objects, spanning 51 categories. It was captured using a Kinect-style 3D camera that records synchronized and aligned 640 × 480 RGB and depth images. Furthermore, the point cloud of each object exhibits authentic geometric noise, occlusion, and non-uniform sampling characteristics.

The detailed description of each dataset is provided in Table 1.

4.2. Experimental settings

Implementation Details. Implemented in PyTorch [34], RDS-Net is evaluated on five datasets: PCN, ShapeNet-55/34, MVP, KITTI, and RGB-D. We set the global feature dimension to 512, while the feature dimension in all upsample blocks is 128. For training settings, we use Adam optimizer [35] with $\beta_1 = 0.9, \beta_2 = 0.999$, and set the initial learning rate to 0.0001 (0.0005 for ShapeNet-55/34) with a continuous decay of 0.5 for every 100 epochs. Additionally, all models are trained on one NVIDIA GeForce RTX 3090 graphic card. Specifically, we train our model on MVP for 200 epochs with a batch size of 15, while on PCN and ShapeNet-55/34 for 320 epochs with a batch size of 8. The other two datasets (KITTI and RGB-D) are not used for model training.

Evaluation Metrics. In this paper, We employ the L1/L2 Chamfer Distance (CD) and F-Score [36] as quantitative metrics to evaluate model performance, where CD measures the minimum distance between the predicted point cloud and the ground truth, while F-Score evaluates the percentage of points that are correctly reconstructed. For the KITTI dataset, since it does not provide complete ground truth, Fidelity Distance (FD) and Minimal Matching Distance (MMD) are employed for evaluation. The former measures how well the input is preserved, while the latter measures how much the output resembles a typical car.

4.3. Comparisons with state-of-the-art methods

Evaluation on PCN. We adopt the standard evaluation metric from [3] to test the performance of different methods and summarize the L1 Chamfer Distance (CD-$\ell 1$) comparisons on eight categories in Table 2. In general, RDS-Net achieves the best performance on all categories except two (Chair and Couch), with an average CD-$\ell 1$ of 6.62. Compared to the second-ranked SeedFormer, our method reduces the average CD-$\ell 1$ by 0.12 (a 1.8% relative improvement), achieving more accurate shape reconstruction. Specifically, RDS-Net obtains notable improvements in three categories: Lamp, Table, and Boat. This indicates that it effectively captures complex multi-part geometries and curved surface details, thanks to our enhanced feature extraction and structure prior transfer. However, for the Chair and Couch categories, performance shows a slight decline, possibly due to the inherent challenges posed by their thin structural components (e.g., legs, armrests) and non-rigid, deformable surfaces (e.g., cushions).

Evaluation on ShapeNet-55/34. We further evaluate RDS-Net on the ShapeNet-55 and ShapeNet-34 datasets. Table 3 presents the L2 Chamfer Distance (CD-$\ell 2$) for specific categories: three categories (Table, Chair, Plane) with the most training samples and three (Birdhouse, Bag, Remote) with the least. Additionally, to evaluate robustness under varying degrees of data incompleteness, we compare the average CD-$\ell 2$ of different methods across three difficulty levels. The overall average CD-$\ell 2$ is also included. In all experimental settings, RDS-Net achieves superior performance compared to other methods. For three categories with limited data, our method outperforms others on two categories (Birdhouse and Bag) and achieves performance comparable to previous SOTA method on the third (Remote), verifying its ability to learn effectively from limited training samples. In particular, RDS-Net reduces

Table 1
Detailed description of the datasets. For KITTI, "Classes=1" indicates evaluation on the single "car" category (standard practice in related works). The symbol "–" indicates that the corresponding attribute is not applicable or unavailable for the dataset.

Datasets	Training	Testing	Resolution	Classes	Views
PCN [3]	28,974	1200	16,384	8	8
ShapeNet-55/34 [29]	41,952	10,518	8192	55/34	all views
MVP [30]	62,400	41,600	2048	16	26
KITTI [31]	–	2401	–	1	–
RGB-D [33]	–	–	2048	51	–

Table 2
Results on PCN in terms of L1 Chamfer Distance $\times 10^3$ (lower is better).

Methods	Plane	Cabinet	Car	Chair	Lamp	Couch	Table	Boat	Average
FoldingNet [4]	9.49	15.80	12.61	15.55	16.41	15.97	13.65	14.99	14.31
TopNet [37]	7.61	13.31	10.90	13.82	14.44	14.78	11.22	11.22	12.15
PCN [3]	5.50	22.70	10.63	8.70	11.00	11.34	11.68	8.59	9.64
GRNet [9]	6.45	10.37	9.45	9.41	7.96	10.51	8.44	8.04	8.83
PoinTr [29]	4.75	10.47	8.68	9.39	7.75	10.93	7.78	7.29	8.38
PMP-Net++[38]	4.39	9.96	8.53	8.09	6.06	9.82	7.17	6.52	7.56
SnowflakeNet [12]	4.29	9.16	8.08	7.89	6.07	9.23	6.55	6.40	7.21
FBNet [32]	3.99	9.05	7.90	7.38	5.82	8.85	6.35	6.18	6.94
SeedFormer [6]	3.85	9.05	8.06	**7.06**	5.21	8.85	6.05	5.85	6.74
WalkFormer [39]	3.73	9.17	8.26	7.28	5.35	**8.69**	6.12	5.74	6.79
RDS-Net	**3.70**	**8.97**	**7.93**	7.08	**5.05**	8.73	**5.90**	**5.60**	**6.62**

Table 3
Results on ShapeNet-55 in terms of L2 Chamfer Distance $\times 10^3$ (lower is better).

Methods	Table	Chair	Airplane	Birdhouse	Bag	Remote	CD-S	CD-M	CD-H	CD-Avg
FoldingNet [4]	2.53	2.81	1.43	4.71	2.79	1.44	2.67	2.66	4.05	3.12
PCN [3]	2.13	2.29	1.02	4.50	2.86	1.33	1.94	1.96	4.08	2.66
TopNet [37]	2.21	2.53	1.14	4.83	2.93	1.49	2.26	2.16	4.30	2.91
PFNet [24]	3.95	4.24	1.81	6.21	4.96	2.91	3.83	3.87	7.97	5.22
GRNet [9]	1.63	1.88	1.02	2.97	2.06	1.09	1.35	1.71	2.85	1.97
PoinTr [29]	0.81	0.95	0.44	1.86	0.93	0.53	0.58	0.88	1.79	1.09
SeedFormer [6]	0.72	0.81	0.40	1.51	0.79	0.46	0.50	0.77	1.49	0.92
CRA-PCN [18]	0.66	0.74	0.37	1.36	0.73	**0.43**	0.48	0.71	1.37	0.85
RDS-Net	**0.58**	**0.66**	**0.36**	**1.29**	**0.67**	0.46	**0.44**	**0.66**	**1.26**	**0.78**

Table 4
Results on ShapeNet-34 in terms of L2 Chamfer Distance $\times 10^3$ (lower is better).

Methods	34 seen categories				21 unseen categories			
	CD-S	CD-M	CD-H	CD-Avg	CD-S	CD-M	CD-H	CD-Avg
FoldingNet [4]	1.86	1.81	3.38	2.35	2.76	2.74	5.36	3.62
PCN [3]	1.87	1.81	2.97	2.22	3.17	3.08	5.29	3.85
TopNet [37]	1.77	1.61	3.54	2.31	2.62	2.43	5.44	3.50
PFNet [24]	3.16	3.19	7.71	4.68	5.29	5.87	13.33	8.16
GRNet [9]	1.26	1.39	2.57	1.74	1.85	2.25	4.87	2.99
PoinTr [29]	0.76	1.05	1.88	1.23	1.04	1.67	3.44	2.05
SeedFormer [6]	0.48	0.70	1.30	0.83	0.61	1.07	2.35	1.34
CRA-PCN [18]	0.45	0.65	1.18	0.76	0.55	0.97	2.19	1.24
RDS-Net	**0.42**	**0.60**	**1.11**	**0.71**	**0.53**	**0.91**	**2.12**	**1.18**

the average CD-$\ell 2$ by 0.07 compared to the second-ranked CRA-PCN, indicating that RDS-Net recovers more detailed and faithful geometric shapes.

For the ShapeNet-34 dataset, we follow [29] to evaluate the generalization ability of RDS-Net for novel object shape completion. The evaluation results are summarized in Table 4, which describes CD-$\ell 2$ on both the seen and unseen categories in detail. Generally, RDS-Net achieves the best average CD-$\ell 2$ of 0.71 on 34 seen categories, outperforming the second-ranked CRA-PCN by 0.05. For 21 unseen categories that differ significantly from the training data, RDS-Net still achieves 1.18 average CD-$\ell 2$ and obtains 0.06 gain over CRA-PCN. These results basically validate the better generalization ability of RDS-Net for novel object reconstruction.

Evaluation on MVP. We evaluate RDS-Net on the MVP dataset in terms of L2 Chamfer Distance (CD-$\ell 2$) and F-Score. Table 5 shows that our method outperforms all competitors on both metrics. In detail, compared to CRA-PCN, RDS-Net reduces CD-$\ell 2$ by 0.32 and improves F-Score by 0.003. This gain is primarily attributed to the proposed DAT module, which enables the network to better capture similar and detailed geometric structures to infer the missing regions. Moreover, the proposed recursive refinement mechanism preserves high-frequency details, especially excelling in reconstructing sharp edges, corners, and complex surface structures, while effectively mitigating over-smoothing in missing regions. Overall, the performance improvement of RDS-Net not only demonstrates its ability to preserve existing geometric structures of partial inputs, but also validates its effectiveness in reconstructing missing regions.

Furthermore, we visually compare different methods on all eight categories of the MVP dataset. As shown in Fig. 5, the shapes predicted by RDS-Net are more detailed and complete, closely resembling the ground truth. Meanwhile, the relatively uniform point distribution indicates smoother and clearer surfaces, and noise is reduced compared

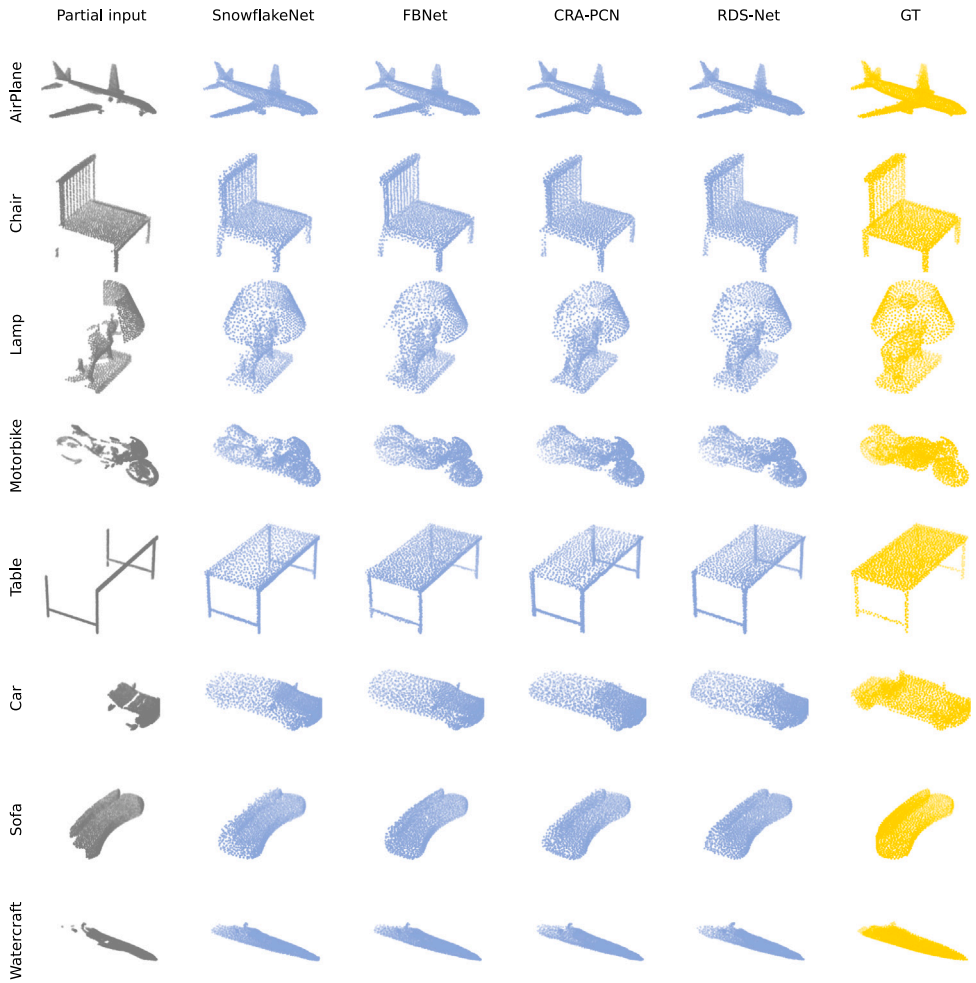

Fig. 5. Visual comparisons on MVP dataset.

Table 5
Results on MVP in terms of L2 Chamfer Distance $\times 10^4$ (lower is better) and F-Score (higher is better).

Methods	CD-ℓ2 ↓	F-Score%↑
TopNet [37]	10.11	0.308
MSN [40]	7.90	0.432
CDN [41]	7.25	0.434
ECG [42]	6.64	0.476
VRCNet [43]	5.96	0.499
SnowflakeNet [12]	6.06	0.500
CRA-PCN [18]	5.33	0.529
RDS-Net	**5.01**	**0.532**

Table 6
Results on KITTI in terms of FD $\times 10^4$ and MMD $\times 10^2$ (lower is better).

Methods	FD↓	MMD↓
PCN [3]	11.12	2.50
SnowflakeNet [12]	2.08	2.81
FBNet [32]	**0.52**	2.97
CRA-PCN [18]	3.37	0.44
RDS-Net	3.79	**0.34**

to previous methods. For instance, in the Lamp category, RDS-Net generates more complete and accurate structures, whereas other methods not only fail to recover the correct structure, but also introduce substantial noise. In the Table category, the completed regions exhibit a more uniform distribution, while the original geometry is faithfully preserved, as evidenced by the clearly defined leg structures. For categories including Car, Sofa, and Watercraft, RDS-Net also achieves more reasonable structure recovery and finer detail preservation than other methods. This shows that two completion goals are well achieved by RDS-Net steadily across different object categories.

Evaluation on KITTI. Next, we also experiment with our RDS-Net on KITTI to test point cloud completion on real 3D car shape. The quantitative results are shown in Table 6. Notably, RDS-Net achieves a moderate FD, reflecting that the input structure is well preserved. Meanwhile, it obtains the lowest MMD, indicating that the predicted shape is more like a car than other methods.

Qualitative results are shown in Fig. 6. The four cases are selected to cover typical challenges encountered in real-world LiDAR-scanned car point clouds, and detailed descriptions of each specific case are provided in the figure caption. It can be observed that while CRA-PCN can roughly recover the car contour, its output is relatively blurry with missing details. In contrast, although RDS-Net exhibits minor limitations in preserving some fine-grained structures, it generates more realistic car shapes. This verifies RDS-Net's advantage in modeling the overall shape of the target in point cloud completion tasks.

Evaluation on RGB-D. This study further evaluates the applicability of RDS-Net in real-world scenarios using the RGB-D object dataset. Specifically, we apply the pre-trained RDS-Net (trained on MVP) directly to this dataset without any fine-tuning or adaptive training. As shown in Fig. 7, although RDS-Net performs well on synthetic data, it exhibits a notable "tailing" issue on real-world inputs. This manifests as

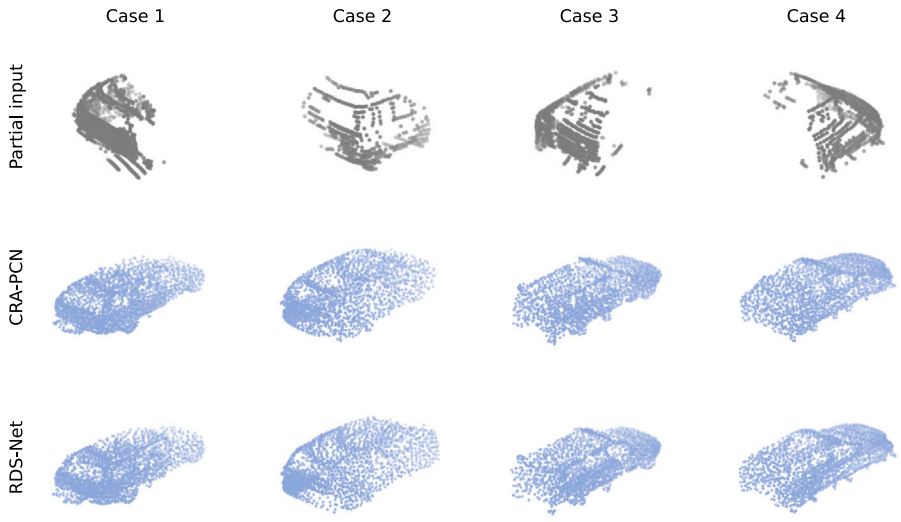

Fig. 6. Visual comparisons on KITTI dataset. The four cases represent different challenging scenarios: Case 1 (moderate front occlusion), Case 2 (severe side sparsity), Case 3 (partial rear occlusion), and Case 4 (highly incomplete oblique view).

spurious geometric extensions that protrude beyond the actual object boundaries.

This phenomenon is primarily attributed to the domain shift that induces a mismatch in geometric priors. Unlike the structured, noise-free synthetic data, the point clouds in the RGB-D dataset exhibit significant sparsity and sensor noise. Since RDS-Net is fully trained on MVP, its learned geometric priors are biased towards idealized, CAD-like distributions. When directly applied to unseen real RGB-D data, the model struggles to accurately capture object boundaries and topological structures, making it difficult to infer reasonable shapes. This domain shift causes the model to rely excessively on local context for extrapolation, resulting in the generation of spurious "tailing" artifacts.

Nevertheless, compared to CRA-PCN, RDS-Net exhibits significant advantages, primarily manifested as follows: (1) Higher structural integrity: While CRA-PCN outputs often suffer from severe fragmentation or distortion (e.g., the Cereal box is nearly distorted), RDS-Net robustly recovers the object's main body, maintaining geometric continuity; (2) Stronger geometric fidelity: For curved objects (e.g., Bowl), the surface generated by RDS-Net is smoother. For cubic objects (e.g., Food box), RDS-Net more accurately recovers their edges, corners, and proportions; (3) Better multi-scale adaptability: For objects of different scales (e.g., Food box, Binder), RDS-Net stably recovers their basic shapes. In contrast, CRA-PCN exhibits instability, often resulting in noticeable deformations.

In summary, although RDS-Net encounters reconstruction challenges on real-world data, specifically the "tailing" artifacts arising from domain shift without fine-tuning, it outperforms existing baseline methods in maintaining structural integrity and ensuring geometric plausibility. Looking ahead, future work will focus on developing self-supervised domain adaptation strategies to enhance the model's robustness and generalization in complex environments.

4.4. Efficiency analysis

In addition, we evaluate the operational efficiency and resource usage of our model, and present the results in Table 7. All methods are evaluated on a single NVIDIA GeForce RTX 3090 graphic card in inference mode (batch size: 32, point cloud resolution: 2048) to ensure a fair comparison. The results indicate that RDS-Net achieves a favorable trade-off between latency and parameter count. However, it exhibits deficiencies in memory usage, suggesting that the model is not yet fully optimized for deployment in resource-constrained scenarios. Therefore, reducing memory overhead and further improving inference efficiency [44] will be important directions for future work.

Table 7
Run-time memory usage, latency and model parameters (lower is better).

Methods	Latency↓	Memory↓	Parameters↓
SnowflakeNet [12]	**0.179 s**	3493 MB	19,320,812
FBNet [32]	0.802 s	**2721 MB**	**3,799,375**
CRA-PCN [18]	0.314 s	3412 MB	7,471,212
RDS-Net	0.644 s	10,986 MB	6,934,745

Table 8
Analysis of different feature extraction designs.

Methods	CD-$\ell 2$ ↓	F-Score%↑
Set Abstraction [11]	5.19	0.520
EdgeConv [26]	5.23	0.522
Point Transformer [27]	5.18	0.520
RDS-Net	**5.01**	**0.532**

4.5. Ablation study

In order to evaluate the contribution of each key component in RDS-Net, we conduct ablation studies on MVP by systematically replacing individual modules with baseline versions, while keeping the rest of the architecture unchanged. Furthermore, all experiments are conducted on the MVP dataset under identical training and evaluation settings.

4.5.1. Effectiveness of SSA

Table 8 presents the evaluation results of models with different feature extraction designs. The comparative models are obtained by replacing SSA with SA [11], EdgeConv [26], and PointTransformer [27] individually, while keeping the other structures unchanged. Specifically, our method achieves 5.01 CD-$\ell 2$ and 0.532 F-Score, outperforming all previous feature extraction designs. This performance gain is primarily attributed to SSA's superior capability in capturing fine geometric features for fine-grained completion.

Fig. 8 visually compares the completion effects of different feature extraction designs. As observed, the SSA module enables the capture of more refined geometric structures, yielding reconstructed shapes that are more consistent with the target objects. In contrast, other variants introduce significant noise and exhibit ambiguous edges. For instance, the model equipped with Point Transformer fails to correctly recover the surface structure of the Lamp category, producing disorganized and sparse points instead. Meanwhile, the EdgeConv-based variant generates an overly smooth surface, thereby losing fine-grained details of the original structure.

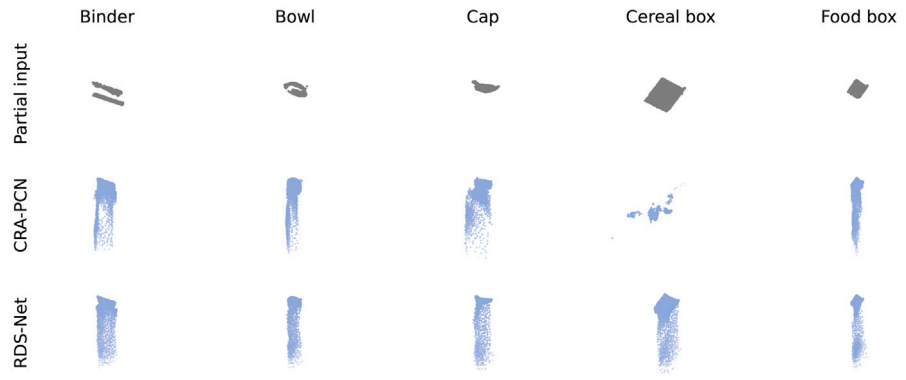

Fig. 7. Visual comparisons on RGB-D dataset.

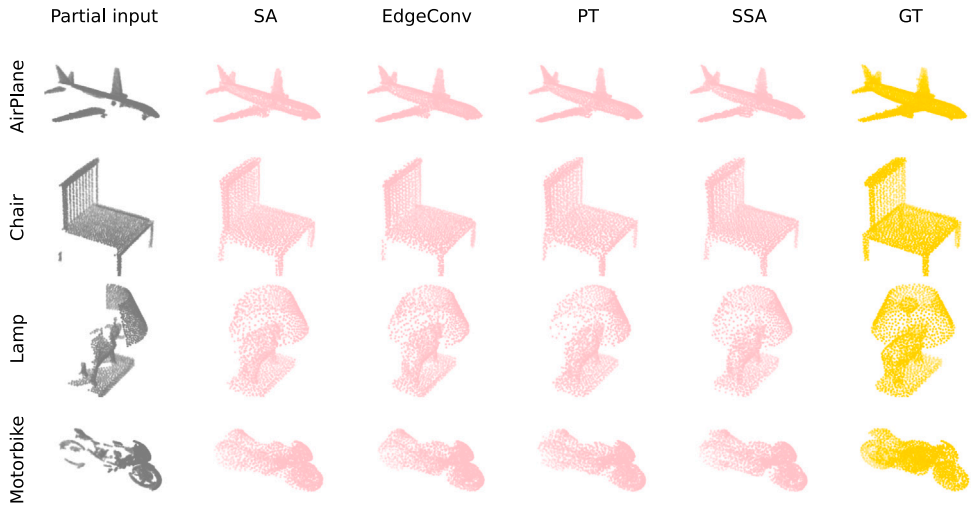

Fig. 8. Visual comparisons on different feature extraction designs.

4.5.2. *Effectiveness of DAT*

For most of the previous generation operations such as deconvolution [12] and folding [4], they are classic and representative paradigms in point cloud generation, and many advanced methods are extended based on these two basic schemes. In contrast, the Upsample Transformer [6] achieves better performance by local aggregation, which is different from the limitations of deconvolution and folding that independently process each point without considering the context information of point clouds. Therefore, we adopt three representative comparative operations (deconvolution, graph convolution [26], and Upsample Transformer) to conduct the ablation study. Specifically, we replace DAT in both the seed generator and the upsample blocks, while keeping the rest of the architecture unchanged. It is worth noting that Transformer-based models such as PointAttN [19] focus on modeling long-range dependencies for topology representation instead of designing novel point generation operators, and they still adopt traditional reshape and MLP schemes for point generation. Thus, such methods are not included in the comparison. As shown in Table 9, our method outperforms all ablated variants, indicating its superior ability to recover faithful details.

With the qualitative results presented in Fig. 9, it is evident that our method produces more reasonable and complete shapes across the four categories: Bench, Guitar, Pistol, and Skateboard. In contrast, other variants struggle to reconstruct accurate and detailed local structures, and some even introduce substantial noise. For instance, in the Pistol category, which features complex geometry and severe incompleteness, our method can better capture the underlying geometric structures and relationships. Although a noticeable discrepancy from the ground truth remains, the original structure is better preserved, and missing regions are inferred with higher quality. Conversely, the EdgeConv-based variant yields scattered noise points and compromises the original geometric integrity. This limitation stems from its tendency to focus on original geometric features while overlooking the holistic shape inference.

Table 9
Analysis of different point generator designs.

Methods	CD-$\ell 2$ ↓	F-Score%↑
Deconvolution [12]	5.28	0.504
EdgeConv [26]	5.69	0.481
UpTransformer [6]	5.07	0.521
RDS-Net	**5.01**	**0.532**

5. Conclusion and future work

In this paper, we propose the recursive structure refinement network with dual-awareness shape transfer (RDS-Net), which further reduces structural information loss in hierarchical feature extraction through the proposed Skip Set Abstraction, enabling the model to capture fine geometric features for fine-grained completion. To effectively utilize potential geometric information to infer the missing regions, this paper designs Dual-Awareness upTransformer to capture similar geometric structures from learned local patterns via Dual-awareness mechanism. By constructing Cross-scale Recursive Expansion with DAT, RDS-Net transfers the multi-scale geometric structures from previous

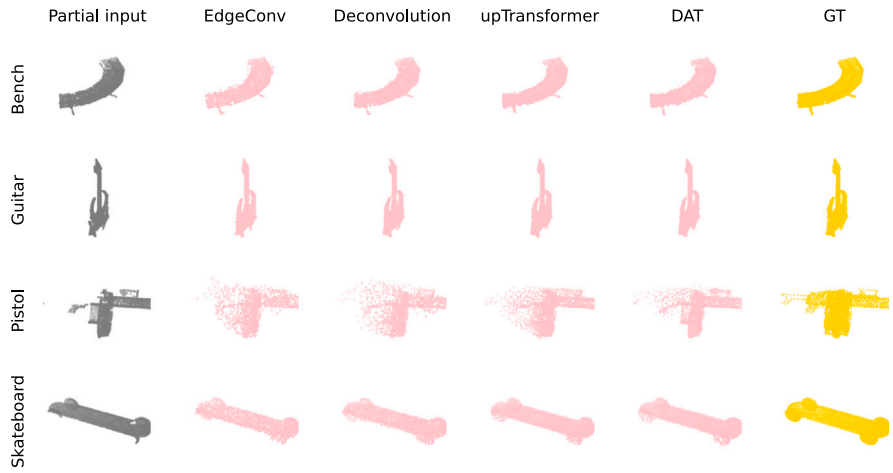

Fig. 9. Visual comparisons on different generator designs.

shape priors to missing regions recursively to recover faithful details, enabling a more controllable and consistent shape completion. Extensive experiments also demonstrate its effectiveness in preserving fine geometric details from partial inputs and inferring missing parts. Beyond the performance improvement, the proposed method can provide researchers related to this field with new ideas for feature extraction, shape inference, and multi-scale detail recovery. The components within RDS-Net can also be flexibly integrated into other networks.

Despite the strengths described above, several limitations still exist in this work. Firstly, it performs relatively less satisfactorily on objects with thin structures, and some fine-grained details are not fully preserved. Secondly, the current framework consumes more memory and computational resources, limiting its deployment in real-time scenarios. Thirdly, the generalization ability on real-scanned point clouds still needs further improvement. In the future, we will address the above limitations along three main directions.(1) For thin and complex structures, we will optimize the feature revisit mechanism in SSA and strengthen cross-region geometric communication to better capture fine-grained structural patterns.(2) For efficiency, we will lightweight the overall architecture by optimizing the CRE module and refining the DAT with sparse coding and efficient attention, thus reducing memory and computation costs for real-time applications.(3) For real-scanned data generalization, we will explore weakly supervised and self-supervised learning, such as multi-view consistency and physical rendering losses, to learn robust geometric priors without full ground truth. Moreover, advanced similarity-preserving representation learning strategies [45–47] can be further incorporated to enhance the model's adaptation and robustness in real-world scenarios.

CRediT authorship contribution statement

Xiaocui Li: Project administration, Methodology, Conceptualization. **Weili Chen:** Writing – original draft, Visualization, Software. **Xinyu Zhang:** Writing – review & editing, Supervision, Methodology. **Yangtao Wang:** Software, Investigation, Data curation. **Wei Liang:** Writing – review & editing, Methodology, Formal analysis. **Keqin Li:** Supervision, Conceptualization.

Declaration of generative AI and AI-assisted technologies in the manuscript preparation process

During the preparation of this work, the authors used generative AI tools to improve language clarity, readability, and manuscript presentation. After using these tools, the authors carefully reviewed and edited the content as needed and take full responsibility for the content of the published article.

Declaration of competing interest

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

Acknowledgments

This work was supported by the National Natural Science Foundation of China (No. 62502156), the Projects of Xiangjiang Laboratory (No. 25XJ03025, No. 24XJJCYJ01002), and the National Key Research and Development Program of China (No. 2023YFC3306204).

Data availability

Data will be made available on request.

References

[1] F. Xu, Y. Xiao, J. Mei, Y. Hu, Q. Fu, H. Shi, Domain-adaptive point cloud semantic segmentation via knowledge-augmented deep learning, Pattern Recognit. (2025) 112528.
[2] J. Li, L. Ding, J. Wang, T. Xu, Local grid rendering networks for 3D object detection in point clouds, Pattern Recognit. (2026) 113244.
[3] W. Yuan, T. Khot, D. Held, C. Mertz, M. Hebert, Pcn: Point completion network, in: 2018 International Conference on 3D Vision (3DV), IEEE, 2018, pp. 728–737.
[4] Y. Yang, C. Feng, Y. Shen, 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.
[5] X. Wen, T. Li, Z. Han, Y.-S. Liu, Point cloud completion by skip-attention network with hierarchical folding, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2020, pp. 1939–1948.
[6] H. Zhou, Y. Cao, W. Chu, J. Zhu, T. Lu, Y. Tai, C. Wang, Seedformer: Patch seeds based point cloud completion with upsample transformer, in: European Conference on Computer Vision, Springer, 2022, pp. 416–432.
[7] A. Dai, C. Ruizhongtai Qi, 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.
[8] D. Stutz, 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.
[9] H. Xie, H. Yao, S. Zhou, J. Mao, S. Zhang, W. Sun, Grnet: Gridding residual network for dense point cloud completion, in: European Conference on Computer Vision, Springer, 2020, pp. 365–381.
[10] C.R. Qi, H. Su, K. Mo, L.J. Guibas, Pointnet: Deep learning on point sets for 3d classification and segmentation, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 652–660.
[11] C.R. Qi, L. Yi, H. Su, L.J. Guibas, Pointnet++: Deep hierarchical feature learning on point sets in a metric space, Adv. Neural Inf. Process. Syst. 30 (2017).

[12] P. Xiang, X. Wen, Y.-S. Liu, Y.-P. Cao, P. Wan, W. Zheng, 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.
[13] Y. Chang, C. Jung, Y. Xu, FinerPCN: High fidelity point cloud completion network using pointwise convolution, Neurocomputing 460 (2021) 266–276.
[14] Z. Chen, F. Long, Z. Qiu, T. Yao, W. Zhou, J. Luo, T. Mei, Anchorformer: Point cloud completion from discriminative nodes, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2023, pp. 13581–13590.
[15] J. Tang, Z. Gong, R. Yi, Y. Xie, L. Ma, Lake-net: Topology-aware point cloud completion by localizing aligned keypoints, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2022, pp. 1726–1735.
[16] D. Zong, S. Sun, J. Zhao, ASHF-net: Adaptive sampling and hierarchical folding network for robust point cloud completion, in: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 35, (4) 2021, pp. 3625–3632.
[17] L. Li, G. Liu, F. Xu, L. Deng, Carvingnet: Point cloud completion by stepwise refining multi-resolution features, Pattern Recognit. 156 (2024) 110780.
[18] Y. Rong, H. Zhou, L. Yuan, C. Mei, J. Wang, T. Lu, Cra-pcn: Point cloud completion with intra-and inter-level cross-resolution transformers, in: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 38, (5) 2024, pp. 4676–4685.
[19] J. Wang, Y. Cui, D. Guo, J. Li, Q. Liu, C. Shen, Pointattn: You only need attention for point cloud completion, in: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 38, (6) 2024, pp. 5472–5480.
[20] W. Wu, Y. Xu, Z. Xie, SDA-net: A global feature point cloud completion network based on serialization and dual attention, IEEE Trans. Vis. Comput. Graphics (2025).
[21] Y. Wang, J. Wang, Y. Shi, L. Sun, B. Yin, LGP-net: local geometry preserving network for point cloud completion, in: 2022 IEEE International Conference on Multimedia and Expo, ICME, IEEE, 2022, pp. 01–06.
[22] B. Fei, W. Yang, L. Ma, W.-M. Chen, Dctr: Noise-robust point cloud completion by dual-channel transformer with cross-attention, Pattern Recognit. 133 (2023) 109051.
[23] Z. Wu, J. Shi, X. Deng, C. Zhang, G. Yang, M. Zeng, Y. Wang, SPAC-net: Rethinking point cloud completion with structural prior, IEEE Trans. Vis. Comput. Graphics 31 (9) (2024) 6268–6279.
[24] Z. Huang, Y. Yu, J. Xu, F. Ni, X. Le, Pf-net: Point fractal network for 3d point cloud completion, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2020, pp. 7662–7670.
[25] F. Gao, Y. Jin, Y. Liu, S. Li, H. Zheng, J. Yu, Bidirectional cross-modal image-guided point cloud completion with multi-scale progressive refinement, Pattern Recognit. (2026) 113123.
[26] Y. Wang, Y. Sun, Z. Liu, S.E. Sarma, M.M. Bronstein, J.M. Solomon, Dynamic graph cnn for learning on point clouds, ACM Trans. Graph. (Tog) 38 (5) (2019) 1–12.
[27] H. Zhao, L. Jiang, J. Jia, P.H. Torr, V. Koltun, Point transformer, in: Proceedings of the IEEE/CVF International Conference on Computer Vision, 2021, pp. 16259–16268.
[28] Y. Wang, D.J. Tan, N. Navab, F. Tombari, Learning local displacements for point cloud completion, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2022, pp. 1568–1577.
[29] X. Yu, Y. Rao, Z. Wang, Z. Liu, J. Lu, J. Zhou, Pointr: Diverse point cloud completion with geometry-aware transformers, in: Proceedings of the IEEE/CVF International Conference on Computer Vision, 2021, pp. 12498–12507.
[30] L. Pan, T. Wu, Z. Cai, Z. Liu, X. Yu, Y. Rao, J. Lu, J. Zhou, M. Xu, X. Luo, et al., Multi-view partial (mvp) point cloud challenge 2021 on completion and registration: Methods and results, 2021, arXiv preprint arXiv:2112.12053.
[31] A. Geiger, P. Lenz, C. Stiller, R. Urtasun, Vision meets robotics: The kitti dataset, Int. J. Robot. Res. 32 (11) (2013) 1231–1237.
[32] X. Yan, H. Yan, J. Wang, H. Du, Z. Wu, D. Xie, S. Pu, L. Lu, Fbnet: Feedback network for point cloud completion, in: European Conference on Computer Vision, Springer, 2022, pp. 676–693.
[33] K. Lai, L. Bo, X. Ren, D. Fox, A large-scale hierarchical multi-view rgb-d object dataset, in: 2011 IEEE International Conference on Robotics and Automation, IEEE, 2011, pp. 1817–1824.
[34] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, et al., Pytorch: An imperative style, high-performance deep learning library, Adv. Neural Inf. Process. Syst. 32 (2019).
[35] K.D.B.J. Adam, et al., vol. 1412, (6) 2014, arXiv preprint arXiv:1412.6980.
[36] M. Tatarchenko, S.R. Richter, R. Ranftl, Z. Li, V. Koltun, T. Brox, What do single-view 3d reconstruction networks learn? in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2019, pp. 3405–3414.
[37] L.P. Tchapmi, V. Kosaraju, H. Rezatofighi, I. Reid, S. Savarese, Topnet: Structural point cloud decoder, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2019, pp. 383–392.
[38] X. Wen, P. Xiang, Z. Han, Y.-P. Cao, P. Wan, W. Zheng, Y.-S. Liu, PMP-net++: Point cloud completion by transformer-enhanced multi-step point moving paths, IEEE Trans. Pattern Anal. Mach. Intell. 45 (1) (2022) 852–867.
[39] M. Zhang, Y. Li, R. Chen, Y. Pan, J. Wang, Y. Wang, R. Xiang, Walkformer: Point cloud completion via guided walks, in: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, 2024, pp. 3293–3302.
[40] M. Liu, L. Sheng, S. Yang, J. Shao, S.-M. Hu, Morphing and sampling network for dense point cloud completion, in: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 34, (07) 2020, pp. 11596–11603.
[41] X. Wang, M.H. Ang Jr., 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.
[42] L. Pan, ECG: Edge-aware point cloud completion with graph convolution, IEEE Robot. Autom. Lett. 5 (3) (2020) 4392–4398.
[43] L. Pan, X. Chen, Z. Cai, J. Zhang, H. Zhao, S. Yi, Z. Liu, Variational relational point completion network, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2021, pp. 8524–8533.
[44] Z. Fu, J. Zhang, L. Wang, L. Xu, H. Laga, Y. Guo, F. Boussaid, M. Bennamoun, CompletionMamba: Taming state space model for point cloud completion, IEEE Trans. Image Process. (2025).
[45] Q. Qin, M. Ge, W. Zhang, L. Huang, J. Nie, Deep stochastic spherical hashing with von mises-Fisher distributions for cross-modal retrieval, IEEE Trans. Knowl. Data Eng. (2026).
[46] Q. Qin, M. Dou, W. Zhang, L. Huang, J. Nie, Deep neighbor discriminant binary embedding for multi-label image retrieval, IEEE Trans. Multimed. (2026).
[47] K. Cheng, Q. Qin, W. Zhang, L. Huang, J. Nie, Deep distance weighted sampling hashing for cross-modal retrieval, IEEE Trans. Multimed. (2026).

Xiaocui Li received the Ph.D. degree in computer system architecture from Huazhong University of Science and Technology, Wuhan, China. She is currently an Associate Professor with Hunan University of Technology and Business, Changsha, China. She has published more than 20 articles in TIP, AAAI, PR, TITS, CIKM, WWWJ, WSDM, ACM TDS, DASFAA, and NCAA. Her current research interests include multi-view feature learning, pattern recognition, computer vision and intelligent transportation systems.

Weili Chen received the B.E. degree in educational technology from Zhejiang Normal University, Jinhua, China, in 2024. He is currently pursuing the M.E. degree with Hunan University of Technology and Business, Changsha, China. His current research interest is point cloud completion.

Xinyu Zhang received the Ph.D. degree in computer application technology from Wuhan University, Wuhan, China, in 2021. He is currently a Post-Doctoral Fellow with Central South University and an Associate Professor with Hunan University of Technology and Business, Changsha, China. He has published more than 30 articles, such as TPAMI, TIP, TITS, TCB, AAAI, and Pattern Recognition. His research interests include pattern recognition, computer vision, data mining, and machine learning.

Yangtao Wang currently serves as an associate professor at the School of Computer Science and Cyber Engineering, Guangzhou University, Guangzhou, China. He received the Ph.D. degree from Huazhong University of Science and Technology, Wuhan, China, in 2021. His main research interests include computer vision, multimodal representation, and graph neural networks. He has published more than 30 papers in international conferences and journals including AAAI, IJCAI, CIKM, ICMR, ICME, WSDM, World Wide Web Journal, Pattern Recognition, IEEE Transactions on Multimedia, ACM Transactions on Data Science, etc.

Wei Liang received the Ph.D. degree in engineering. He is currently a Professor with Hunan University of Technology and Business. He has published more than 40 high-quality papers in prestigious international journals including IEEE JSAC, TNNLS, TII, and Information Fusion. He has been granted more than 10 national invention patents with an H-index of 26, and was listed among the World's Top 2% Scientists by Stanford University for 2023–2024. His research interests include computational intelligence, industrial internet, and bioinformatics.

Keqin Li is a Member of Academia Europaea and IEEE Fellow. He is currently a SUNY Distinguished Professor with the State University of New York at New Paltz, New Paltz, USA, and a National Distinguished Professor with Hunan University, Changsha, China. He has published more than 1000 articles in reputable journals and conferences, including TPDS, TC, TCC, TSC, JPDC, and IEEE Internet of Things Journal. His research interests include parallel and distributed computing, cloud computing, fog/edge computing, high-performance computing, and machine learning.