CompletionMamba: Taming State Space Model for Point Cloud Completion

Zhiheng Fu, Jiehua Zhang, Longguang Wang, Lian Xu, Hamid Laga, Yulan Guo, *Senior Member, IEEE*, Farid Boussaid, and Mohammed Bennamoun, *Senior Member, IEEE*

***Abstract*—Point cloud completion aims to reconstruct complete 3D shapes from partial scans. The long-range dependencies between points and shape perception are crucial for this task. While Transformers are effective due to their global processing ability, the quadratic complexity of their attention mechanism makes them unsuitable for long sequences when computational resources are constrained. As an alternative, State Space Models (SSMs) provide a memory-efficient solution for handling long-range dependencies, yet applying them directly to unordered point clouds presents challenges because of their intrinsic causality requirements. Existing methods attempt to address this by sorting points along a single axis. This, however, often overlooks complex causal relationships in 3D space since adjacency relationships based on Euclidean distance between points in the 3D space may not be preserved by this linear arrangement. To overcome this issue, we introduce CompletionMamba, a novel SSM-based network designed to harness SSMs for capturing both global and local dependencies within a point cloud. Initially, the input point cloud is causally structured by rearranging its coordinates. Then, a local SSM framework is proposed that defines neighborhood spaces around each point based on Euclidean distance, enhancing the causal structure. Although local SSM enhances relationships in short and long distance sequences, it still lacks full shape modeling of point cloud. To address this, we propose a novel shape-aware Mamba by integrating the shape code of each 3D shape into the model, enabling shape information propagation to all points. Our experiments show that CompletionMamba achieves state-of-the-art performance on both the MVP and PCN datasets.**

***Index Terms*—Point cloud completion, state space models, short-long-range dependency.**

Received 4 December 2024; revised 4 May 2025; accepted 22 July 2025. Date of publication 25 August 2025; date of current version 28 August 2025. This work was supported in part by Australian Research Council under Grant ARC DP210101682 and Grant ARC DP220102197, in part by The University of Western Australia (UWA) Research Collaboration Award under Grant 2023/GR001286, in part by Jilin Province Science and Technology Development Projects under Grant 20250102209JC, in part by the Science and Technology Research Projects of the Education Office of Jilin Province under Grant JJKH20251951KJ, and in part by China Postdoctoral Science Foundation under Grant 2025T180433. The associate editor coordinating the review of this article and approving it for publication was Prof. Xuejin Chen. *(Corresponding author: Yulan Guo.)*

Zhiheng Fu, Lian Xu, and Mohammed Bennamoun are with the Department of Computer Science and Software Engineering, The University of Western Australia, Perth, Crawley, WA 6009, Australia (e-mail: zhiheng.fu@research.uwa.edu.au; mohammed.bennamoun@uwa.edu.au).

Jiehua Zhang is with the College of Computer Science, The University of Oulu, 90100 Oulu, Finland.

Longguang Wang is with the College of Electronic Science, Aviation University of Air Force, Changchun 130022, China.

Hamid Laga is with the Information Technology Discipline, Murdoch University, Murdoch, WA 6150, Australia.

Yulan Guo is with the School of Electronics and Communication Engineering, Sun Yat-sen University (SYSU), Guangzhou 510275, China (e-mail: guoyulan@sysu.edu.cn).

Farid Boussaid is with the Department of Electrical, Electronic and Computer Engineering, The University of Western Australia (UWA), Perth, Crawley, WA 6009, Australia.

Digital Object Identifier 10.1109/TIP.2025.3597041

I. Introduction

THE popularization of Light Detection and Ranging (LiDAR) and depth cameras has significantly accelerated the development of 3D computer vision. However, due to limited viewpoints, occlusions from other objects, and resolution constraints, captured point clouds often contain missing parts. Point cloud completion aims to fill in these missing parts, thereby facilitating downstream tasks such as autonomous driving and robotic grasping [1], [2], [3], [4].

Point cloud completion methods can be roughly classified into traditional methods and deep learning based methods. Traditional methods mainly include geometry-based and alignment-based approaches. They either use geometric cues [5], [6], [7], [8], [9], [10], [11], [12], [13], such as local interpolations and objects' symmetry, or match the partial input with template models from a large shape database [14], [15], [16], [17], [18]. However, these methods can only fill objects that exhibit small holes. They also require expensive optimization during inference, making them impractical for online applications. On the other hand, deep learning-based methods [19], [20], [21], [22], [23], [24], [25], [26], [27], [28], [29], [30], [31], [32] follow an Encoder-Decoder structure. Normally, they adopt PointNet [33], PointNet++ [34] or DGCNN [35] to achieve feature extraction in the encoder. Recent research on point cloud completion has focused on how to obtain the skeleton of objects based on anchor points [28], [31], [32], [36]. These methods use Transformer to build relationships between anchor points of incomplete input point cloud and missing points of complete 3D shapes. Nevertheless, these approaches face a major obstacle: the quadratic complexity of attention mechanisms hampers their scalability to large-scale point cloud. Consequently, transformer-based methods typically employ a high down-sampling rate, resulting in a significant loss of local details in the input point cloud.

Recent progress in State Space Models (SSMs) [37], [38], [39] such as Mamba, an efficient hardware perception algorithm based on structured SSMs with time-varying parameters, effectively handle long-range dependencies, demonstrating

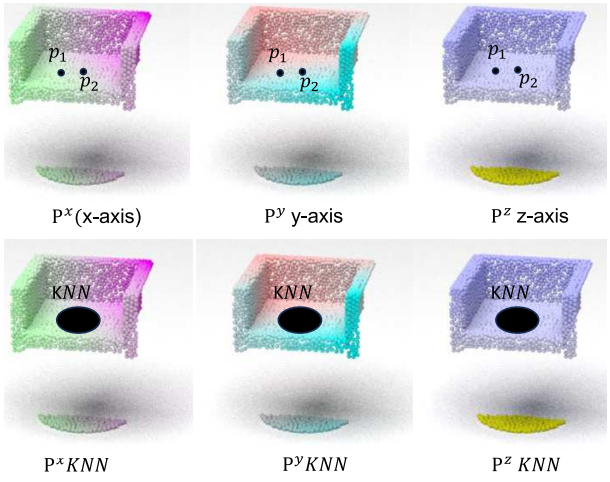


Fig. 1. Visualization of point adjacency. The partial object (**P**) is rendered using x, y and z values, respectively. Similar colors indicate similar x, y and z coordinate values. Points p_1 and p_2 appear adjacent in the z-order point cloud sequence ($\mathbf{P}^z$) but are distant in the x-order and y-order point cloud sequences ($\mathbf{P}^x$ and $\mathbf{P}^y$). Despite this, p_1 and p_2 maintain adjacency in Euclidean space since their Euclidean distances to the central point are unchanged in different reordered point cloud sequences ($\mathbf{P}^xKNN$, $\mathbf{P}^yKNN$ and $\mathbf{P}^zKNN$). This demonstrates that local neighborhood structure is preserved across different coordinate-based orderings.

impressive results in 2D visual tasks. Point Mambas [40], [41], [42], [43] extend the Mamba framework to 3D data, effectively capturing point cloud structures and showcasing its capabilities in 3D visual tasks. There are two primary reasons that Mamba is suitable for point cloud completion. (1) The global receptive field characteristic of Mamba is beneficial for learning global relationships between points, which is the key to the point cloud completion task. (2) The linear complexity of Mamba allows longer point relationship building compared to transformers, leading to detailed preservation and reconstruction. Addressing the unordered nature of point clouds poses a significant challenge for vanilla Mamba due to its causality requirement, which assumes a well-defined order in the input data. Although previous reordering strategies [40], [41], [42] can mitigate the unordered characteristic of point clouds, substantial differences in the 3D spatial coordinates between adjacent points in the reordered sequence remain inevitable (as shown in Fig. 1). Furthermore, the vanilla Mamba framework only establishes relationships between points, neglecting the relationship between points and the entire shape. This oversight results in weak detail preservation and weak global perception abilities, as illustrated in Fig. 2 (a). The mismatch between the unordered nature of point clouds and the sequential processing of Mamba necessitates novel architectural and algorithmic solutions to effectively capture both local and global relationships within the point cloud data.

Our approach aims to address the unordered nature of point clouds for Mamba via leveraging local neighborhood information. A key insight is that, no matter how a point cloud is rearranged, the local k nearest neighbors of each point in 3D space remains consistent. By utilizing these local neighborhood points surrounding each point, we can capture the intrinsic local geometric structure and relationships, which are invariant to the overall ordering. Moreover, to enhance the shape perception ability of vanilla Mamba, we propose to integrate global features of objects into Mamba. Thus, in this paper, we propose CompletionMamba, which combines the proposed Local Mamba block (LM) and Shape-aware Mamba block (SM) to form the Local and Shape-aware Mamba block (LSM), for point cloud completion, as shown in Fig. 2 (b). Specifically, inspired by [40], we first reorder the point cloud as well as the point-wise features in ascending order of their (x, y, z) coordinates to mimic the causal data structure required by the LSM block. After the reordering operation, we obtain three reordered point cloud sequences (e.g., $\mathbf{P}^x$ represents the point cloud sequence **P** sorted in ascending order along the x-axis). Unlike [40], we apply the proposed LSM block to sequentially process the $\mathbf{P}^x$, $\mathbf{P}^y$ and $\mathbf{P}^z$ point cloud and point-wise features rather than directly concatenating these three ordered point clouds. In the LM module, we use k-Nearest Neighbors (KNN) to establish local neighborhoods in the 3D space and then apply Mamba to build relationships within these local areas. Note that, KNN refers to the k nearest neighbors of each point in the Euclidean space rather than the adjacent k points in the reordered point cloud. This approach not only effectively extracts local geometric structures but also promotes causality between points in the reordered point cloud. For point cloud completion, global shape perception (normally represented by a global feature) can enhance the completeness of point cloud completion. In the proposed SM block, we introduce the shape code as the first hidden state in Mamba to increase shape awareness for point-wise features. Experimental results show that our CompletionMamba can recover detailed complete 3D shapes and achieve state-of-the-art results on both MVP [44] and PCN datasets [19]. Our contributions are as follows:

- We propose a new Mamba-based point cloud completion framework, termed CompletionMamba, which builds short and long distance dependency with linear complexity;
- The proposed CompletionMamba framework integrates two novel key components: a local Mamba module that captures fine-grained details, and a shape-aware Mamba module that models global shape information. These components are combined with a reordering strategy that mimics causal data for point clouds, enabling better local preservation and global perception abilities;
- The proposed CompletionMamba achieves state-of-the-art completion results on MVP, PCN and ShapeNet-55/34 datasets.

II. Related Work

In this section, we review existing research on point cloud processing, focusing on two main areas: point cloud completion (Sec. II-A) and State Space Models (SSMs) (Sec. II-B).

A. Point Cloud Completion

Historically, significant progress in 3D reconstruction and shape completion has been made using structured volumetric methods and robust 3D convolutions [45], [46], [47], [48],

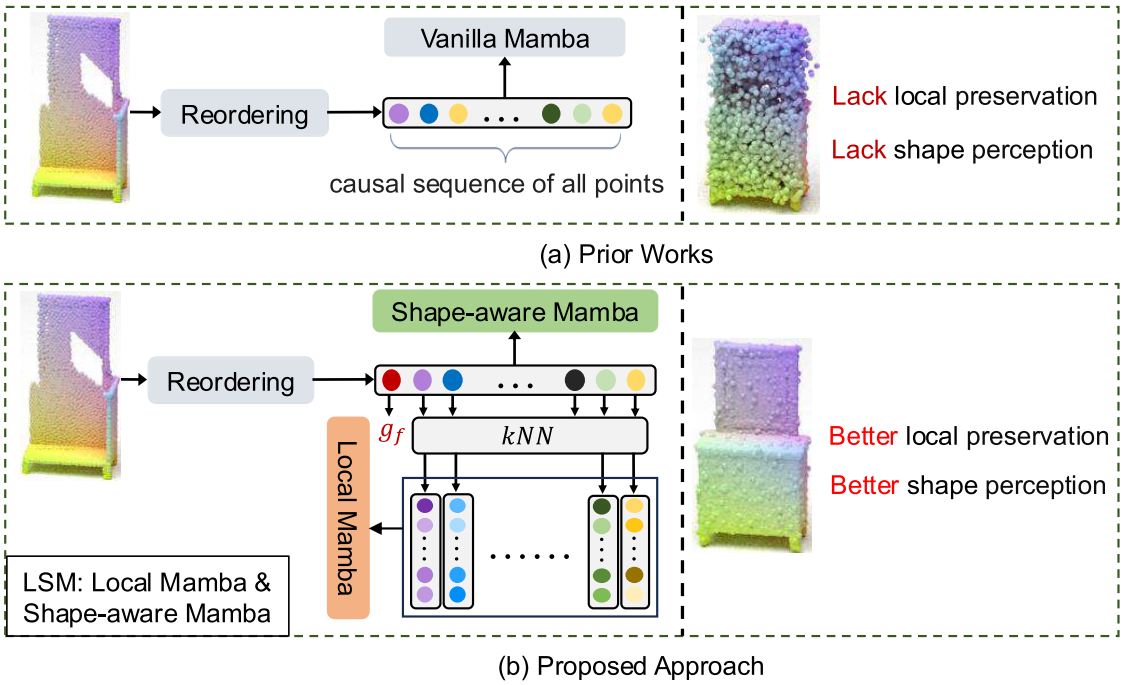


Fig. 2. Comparison of prior approaches and the proposed method. (a) **Prior Works:** Vanilla Mamba processes the reordered point cloud as a simple causal sequence of all points, resulting in a lack of local preservation and shape perception, as shown by the disordered structure. (b) **Proposed Approach:** Our method introduces a novel Local-Shape Mamba (LSM) framework, consisting of Local Mamba and Shape-aware Mamba. Local Mamba operates within the local neighborhood of each point, determined by its k-nearest neighbors (kNN) in Euclidean space, preserving local structure effectively. Shape-aware Mamba integrates shape codes with Vanilla Mamba, applying them across the entire point cloud to establish global relationships among points, enhancing both local preservation and shape perception. Here, $\boldsymbol{g}_f$ represents the global shape feature, contributing to a more coherent representation of the object's structure.

[49], [50]. These approaches, however, come with high computational and memory requirements. Sparse representation techniques [51], [52] attempt to address these issues, but often result in the loss of detailed information due to the quantification involved.

Recent shifts toward unstructured point clouds as representations for 3D objects have helped to reduce memory usage and better preserve fine details. This transition brings new challenges, as standard convolution operations do not translate well to the unordered nature of point clouds. Innovations like PointNet and its variants [33], [34] have allowed for the direct handling of 3D points across multiple downstream tasks. The PCN network [19] adopts a global feature extraction method inspired by PointNet [33] and introduces a folding technique [23] for point generation. Efforts to capture local structures in point clouds have led to multiscale feature extraction methods [53]. Lyu et al. [54] and Chen et al. [55] approached point cloud completion as a conditional generation task using Denoising Diffusion Probabilistic Models (DDPM) [56], [57], [58]. Although these methods achieve fine-grained completion using a simple mean squared error loss function [59], they are computationally demanding, limiting their usage to the preliminary stages of coarse point cloud generation. Examining the role of viewpoint information, Fu et al. [60] found that it enhances completion quality and performance but requires an additional trained model for viewpoint representation learning. More recent architectures [28], [29], [30], [31], [32], [36], [43] use Transformers for point cloud completion due to their ability to establish long-range dependencies. However, the quadratic complexity of their attention mechanism makes them unsuitable for long sequences (a large number of points) when computational resources are limited. In this paper, we achieve point cloud completion using Mamba [37], a memory-efficient solution for handling long-range dependencies.

B. State Space Models

State Space Models (SSMs) have emerged as a promising solution to effectively manage long sequence data, enhancing long sequence modeling. For example, HiPPO [61] combines linear state space equations with deep learning techniques. Gu et al. [62] employs linear state space equations in conjunction with a simple neural network, leading to state-of-the-art performance on temporal sequences. The Structured State Space Sequence Model (S4) [63] introduces a parameterization method for stable diagonalization, while Gated State Space [64] reduces the dimensionality of the state space module. Smith et al. [65] simplifies the S4 structure for practical implementation.

Unlike previous time-invariant systems, Mamba [37] has undergone significant development with the introduction of a selection mechanism and hardware-aware algorithms for efficient information processing. MoE-Mamba [66] integrates the Mixed Expert Model (MoE) with Mamba, improving state selection and improving model expressiveness and generalization. MambByte [67] operates directly on bytes, eliminating errors and overhead from word segmentation. GraphMamba [68] extends the state space model to non-sequential graph data, improving contextual reasoning through node priorities and arrangement strategies. Mamba has also achieved

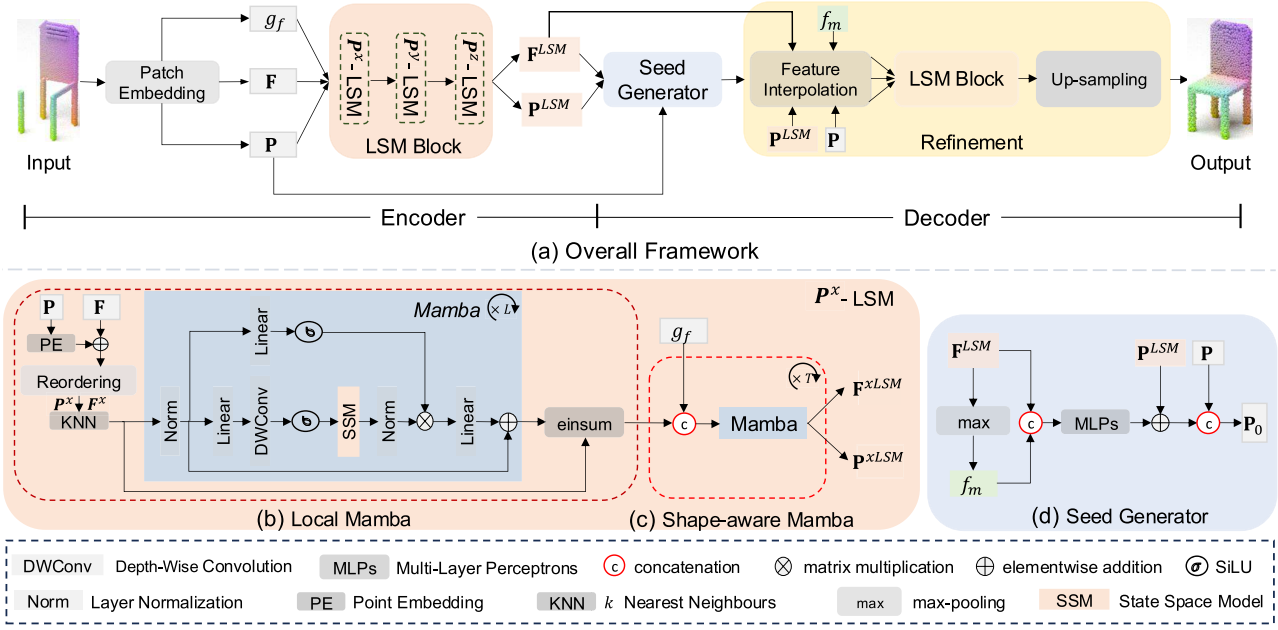


Fig. 3. Overview of the CompletionMamba framework. The overall architecture is illustrated in (a), comprising four main stages: patch embedding, the LSM block (details in (b) Local Mamba and (c) Shape-aware Mamba), seed generation (d), and refinement. The proposed Local-Shape Mamba (LSM) block integrates **Local Mamba** and **Shape-aware Mamba** modules to capture local and global shape information, respectively. Local Mamba leverages local neighborhood information using k-nearest neighbors (kNN), while Shape-aware Mamba establishes shape coherence by incorporating global shape features (gf). The refinement stage includes feature interpolation and up-sampling, producing a coherent output that preserves structural details. Key components include Depth-Wise Convolution (DWConv), Multi-Layer Perceptrons (MLPs), point embedding (PE), and normalization layers.

remarkable results in biomedical imaging [69], [70], [71], [72]. For instance, U-Mamba [73] enhances image segmentation accuracy and robustness, while SegMamba [74] performs segmentation on 3D images by leveraging Mamba. FDVMNet [75] introduces a network based on the frequency domain for the correction of image exposure in endoscopic images. Vivim [76] effectively handles changes in medical videos using Mamba in video frames. Mamba has been applied in point cloud applications [40], [41], [42], [43], [77], leveraging its efficient linear complexity for long-sequence modeling. However, attempts to address Mamba's intrinsic causality requirements by sorting points along a single axis may overlook complex causal relationships in 3D space. This linear arrangement often fails to preserve adjacency relationships between points, which are crucial for capturing intricate spatial dependencies.

In contrast to previous approaches [40], [41], [42], [43], [77], we propose to leverage local neighborhood information to address their limitations. Additionally, we introduce a shape-aware Mamba block to establish connections between individual points and the overall shape, enhancing Mamba's ability to perceive accurate shapes.

III. Method

A. CompletionMamba

1) Overview: In this paper, we follow an Encoder-Decoder structure to achieve point cloud completion (as shown in Fig. 3 (a)). The encoder is composed of patch embedding and the proposed LSM block. The decoder consists of the seed generator and the refinement module. Given an input incomplete point cloud shape $\mathbf{I} \in \mathcal{R}^{N\times 3}$, we first use a set abstraction module as in [34] to achieve patch embedding, obtaining down-sampled M points $\mathbf{P}$ as well as the corresponding point-wise features $\mathbf{F}$. In addition, we apply the max-pooling operation on the point-wise features to extract the global feature g_f. Then, the features are fed to the proposed local and shape-aware Mamba (LSM) block to build the mapping between observed points and missing points. Next, we propose a seed generator to obtain a coarse prediction. Finally, the proposed refinement, which consists of the feature interpolation module, the LSM block, and the upsampling module, is utilized to predict the refined shape.

2) LSM Block: After getting points $\mathbf{P}$ and point-wise features $\mathbf{F}$ from the patch embedding, we use the proposed LSM Block to predict missing-point features. Specifically, we first apply Multi-layer Perceptrons (MLPs) to achieve position embedding $\mathbf{P}_e$, which is added to $\mathbf{F}$, obtaining hidden states of the LSM block. Then, these hidden states are reordered in ascending order along the x-axis, y-axis, and z-axis ($\mathbf{P}^x$, $\mathbf{P}^y$ and $\mathbf{P}^z$). Next, the three reordered point sets ($\mathbf{P}^x$, $\mathbf{P}^y$ and $\mathbf{P}^z$) and hidden states are sequentially fed to the proposed LSM block. Here, we describe the LSM using the x coordinate as an example (shown in Fig. 3 (b) and Fig. 3 (c)). To be specific, we rearrange the input point cloud $\mathbf{P}$ and the hidden states in ascending order of the x-coordinate values, obtaining $\mathbf{P}^x$, $\mathbf{F}^x$.

In local Mamba, we use KNN to search k nearest neighbors for each point based on the Euclidean distance and then use the Mamba block (Fig. 3 (b)) to build local relationships between k closest neighbors. In each Mamba block, layer normalization (LN), SSM, depth-wise convolution [78], and residual connections are employed. A standard Mamba layer is shown in Fig. 3 (b), and the output can be summarized as

follows:

$$S'_l = \mathrm{DW}(\mathrm{MLP}(\mathrm{LN}(S_{l-1}))),$$
$$S_l = \mathrm{MLP}(\mathrm{LN}(\mathrm{SSM}(\sigma(S'_l))) \times \sigma(\mathrm{LN}(S_{l-1}))) + S_{l-1} \quad (1)$$

where $S_l \in \mathcal{R}^{M\times C}$ is the output of the l-th block and S_0 is equal to $\mathbf{F}^x$. DW means the depth-wise convolution. Following the DW, a SiLU [79] and SSM are adopted. Next, we aggregate local geometric information from k nearest neighbors to each point using the einsum operation. In shape-aware Mamba (Fig. 3 (c)), we treat the output features $\mathbf{F}^{LM}$ of local Mamba as input hidden states. To enable each point feature to better capture the overall geometric structure of objects, we concatenate the global feature g_f with $\mathbf{F}^{LM}$ and then use the Mamba block to build relationships among all points and the relationship between points and the shape. Note that, the global feature is not updated iteratively. The outputs of the LSM are $\mathbf{F}^{xLSM}$ and $\mathbf{P}^{xLSM}$.

We use the proposed LSM block to process the reordered point cloud ($\mathbf{P}^x$, $\mathbf{P}^y$ and $\mathbf{P}^z$) sequentially and obtain the processed points $\mathbf{P}^{LSM}$ and features $\mathbf{F}^{LSM}$ treated as the point-wise features of missing points. The LM module and SM module are iteratively used for L and T times, respectively.

3) Seed Generator: In the seed generator (as shown in Fig. 3 (d), we first use the max-pooling operation on $\mathbf{F}^{LSM}$ to get the presentation f_m of the missing shape. Inspired by [80], we treat point cloud completion as a partial point cloud transformation problem. Thus, we concatenate the presentation f_m and $\mathbf{F}^{LSM}$ to predict residuals of $\mathbf{P}^{LSM}$, obtaining missing points $\mathbf{P}_s$. Finally, we concatenate the observed points $\mathbf{P}$ and missing points $\mathbf{P}_s$ to achieve the coarse prediction $\mathbf{P}_0$.

4) Refinement: In the refinement module, we aim to reconstruct a high-resolution complete point cloud with detailed structures. To better build the relationships between the coarse prediction and the refined prediction, we adopt two stages. In the first stage, we build the relationships between the observed points $\mathbf{P}$ and the predicted missing points $\mathbf{P}_s$ in the feature space. Specifically, we use trilinear interpolation to acquire the point-wise features $\mathbf{F}_i$ of $\mathbf{P}$ using $\mathbf{F}^{LSM}$ according to the position relationships between down-sampled $\mathbf{P}$ from input point cloud and predicted missing points $\mathbf{P}_s$. Then, MLPs are used to project $\mathbf{P}_0$ to point-wise features $\mathbf{F}_w$, which are concatenated with $\mathbf{F}_i$ and f_m. The concatenated features are first processed to initial hidden states of LSM block using MLPs. Then, the max-pooling operation is used to acquire the global representation of $\mathbf{P}_0$. In the second stage, the point-wise features $\mathbf{F}_w$, the global representation of $\mathbf{P}_0$ and the initial hidden states are fed to the proposed LSM block to obtain global relationships between points and relationships between entire shape and each point. Next, nearest neighbor interpolation and 1D deconvolution operations are used on the output points and features of LSM block to achieve point and feature up-sampling, respectively. Finally, the up-sampled features are fed to MLPs to acquire the residuals of up-sampled points, deriving the refined prediction $\mathbf{P}_1$.

B. Loss Function

To quantify the disparity between two point clouds, we adopted the Chamfer Distance (CD) for its efficiency compared to Earth Mover's Distance (EMD).

$$\mathcal{L}_{CD}(P,G) = \frac{1}{\|P\|_1}\sum_{p\in P}\min_{g\in G}\|p-g\|_2 + \frac{1}{\|G\|_1}\sum_{g\in G}\min_{p\in P}\|g-p\|_2, \quad (2)$$

where P and G represent the predicted complete point clouds and the actual ground truth, respectively.

To impose specific constraints on point clouds created during coarse and fine completion, we down-sampled the ground truth point clouds to match the sampling density of $\mathbf{P}_0$. In the refinement stage, CD is used to measure the difference between the refined prediction $\mathbf{P}_1$ and ground truth. We also exploit the partial matching loss from [81] to preserve the structural shape integrity of the input point cloud. This is a unidirectional constraint designed to align one shape to another. The partial matching loss ensures that the output point cloud partially matches the input to a certain extent, which we refer to as the preservation loss, $\mathcal{L}_{pre}$. The overall training loss is given as:

$$\mathcal{L} = \mathcal{L}_{CD}(\mathbf{P}_0, \mathbf{P}_{gt_0}) + \mathcal{L}_{CD}(\mathbf{P}_1, \mathbf{P}_{gt}) + \mathcal{L}_{pre}. \quad (3)$$

where $\mathbf{P}_{gt_0}$ is down-sampled ground truth point clouds corresponding to $\mathbf{P}_0$.

IV. Experiments

In this section, we will first introduce the datasets and discuss the implementation details (Sec. IV-A), then move on our completion results on both the MVP (Sec. IV-B) and PCN datasets (Sec. IV-C).

A. Datasets and Implementation Details

1) MVP Dataset [44]: The MVP dataset includes 16 categories with 4000 CAD models. Each model is virtually scanned from 26 camera positions to create partial scans. In our experiments, we generate complete point clouds with point count totals of 2048, 4096, 8192, and 16384 points, paired with incomplete point clouds of 2048 points. We use the $\mathcal{L}_2$ Chamfer Distance version to assess our results.

2) PCN Dataset [19]: Originating from a subset of the ShapeNet dataset [47], the PCN dataset includes complete point clouds of 16384 points against incomplete point clouds with 2048 points. Here, we use the $\mathcal{L}_1$ Chamfer Distance version, in line with the methodology in [19].

3) ShapeNet-55 Dataset [31]: The ShapeNet-55 dataset contains 55 categories and includes 41,952 and 10,518 shapes in the training and testing sets, respectively.

4) ShapeNet-34 Dataset [31]: The ShapeNet-34 training set comprises 46,765 shapes across 34 categories. For evaluation, the test set consists of 5,705 shapes, split into two subsets: 3,400 shapes from the 34 seen categories, and 2,305 shapes from 21 unseen categories.

5) Implementation Details: For patch embedding, we used FPS to down-sample 512 points from the input incomplete point cloud in the encoder. For coarse completion, we predicted an initial complete shape with 1024 points. In the local Mamba block, we set k to 12 for KNN. We opted for the

TABLE I

SHAPE COMPLETION RESULTS ON THE MULTI-VIEW PARTIAL (MVP) POINT CLOUD DATASET (16,384 POINTS) IN TERMS OF L_2 CHAMFER DISTANCE $\times 10^4$. THE LOWER, THE BETTER

Method	airplane	cabinet	car	chair	lamp	sofa	table	watercraft	bed	bench	bookshelf	bus	guitar	motorbike	pistol	skateboard	CD-Avg.
PCN [19]	2.95	4.13	3.04	7.07	14.93	5.56	7.06	6.08	12.72	5.73	6.91	2.46	1.02	3.53	3.28	2.99	6.02
TopNet [21]	2.72	4.25	3.40	7.95	17.01	6.04	7.42	6.04	11.56	5.62	8.22	2.37	1.37	3.90	3.97	2.09	6.36
MSN [22]	2.07	3.82	2.76	6.21	12.72	4.74	5.32	4.80	9.93	3.89	5.85	2.12	0.69	2.48	2.91	1.58	4.90
Wang et. al. [26]	1.59	3.64	2.60	5.24	9.02	4.42	5.45	4.26	9.56	3.67	5.34	2.23	0.79	2.23	2.86	2.13	4.30
ECG [24]	1.41	3.44	2.36	4.58	6.95	3.81	4.27	3.38	7.46	3.10	4.82	1.99	0.59	2.05	2.31	1.66	3.58
GRNet [25]	1.61	4.66	3.10	4.72	5.66	4.61	4.85	3.53	7.82	2.96	4.58	2.97	1.28	2.24	2.11	1.61	3.87
NSFA [53]	1.51	4.24	2.75	4.68	6.04	4.29	4.84	3.02	7.93	3.87	5.99	2.21	0.78	1.73	2.04	2.14	3.77
VRCNet [44]	1.15	3.20	2.14	3.58	5.57	3.58	4.17	2.47	6.90	2.76	3.45	1.78	0.59	1.52	1.83	1.57	3.12
VAPCNet [60]	0.78	3.19	2.10	3.05	3.16	3.14	3.26	2.15	5.36	1.92	3.08	1.68	0.33	1.39	1.34	0.95	2.40
CP3 [82]	0.74	**2.94**	2.25	2.78	2.54	**2.87**	2.84	2.00	5.24	1.98	2.87	1.67	0.45	1.45	1.23	0.92	2.27
GTNet [80]	**0.64**	3.08	2.15	**2.73**	**2.46**	2.90	2.72	**1.85**	**4.24**	**1.78**	3.09	1.70	0.35	1.36	**1.10**	**0.69**	**2.19**
CompletionMamba	0.70	2.98	**2.06**	2.78	2.76	2.92	**2.60**	1.99	4.75	1.82	**3.05**	**1.59**	**0.31**	**1.32**	1.22	0.83	2.20

TABLE II

SHAPE COMPLETION RESULTS ON THE MULTI-VIEW PARTIAL (MVP) POINT CLOUD DATASET (16,384 POINTS) IN TERMS OF F-SCORE@1%. THE HIGHER, THE BETTER

Method	airplane	cabinet	car	chair	lamp	sofa	table	watercraft	bed	bench	bookshelf	bus	guitar	motorbike	pistol	skateboard	F1
PCN [19]	0.861	0.641	0.686	0.517	0.455	0.552	0.646	0.628	0.452	0.694	0.546	0.779	0.906	0.665	0.774	0.861	0.638
TopNet [21]	0.798	0.621	0.612	0.443	0.387	0.506	0.639	0.609	0.405	0.680	0.524	0.766	0.868	0.619	0.726	0.837	0.601
MSN [22]	0.879	0.692	0.693	0.599	0.604	0.627	0.730	0.696	0.569	0.797	0.637	0.806	0.935	0.728	0.809	0.885	0.710
Wang et. al. [26]	0.898	0.688	0.725	0.670	0.681	0.641	0.748	0.742	0.600	0.797	0.659	0.802	0.931	0.772	0.843	0.902	0.740
ECG [24]	0.906	0.680	0.716	0.683	0.734	0.651	0.766	0.753	0.640	0.822	0.706	0.804	0.945	0.780	0.835	0.897	0.753
GRNet [25]	0.861	0.641	0.686	0.517	0.455	0.552	0.646	0.628	0.452	0.694	0.546	0.779	0.906	0.665	0.774	0.861	0.638
NSFA [53]	0.903	0.694	0.721	0.737	0.783	0.705	0.817	0.799	0.687	0.845	0.747	0.815	0.932	0.815	0.858	0.894	0.783
VRCNet [44]	0.928	0.721	0.756	0.743	0.789	0.696	0.813	0.800	0.674	0.863	0.755	0.832	0.960	0.834	0.887	0.930	0.796
VAPCNet [60]	0.942	0.762	0.758	0.786	0.844	0.759	0.845	0.824	0.736	0.892	0.808	0.854	0.978	0.845	0.902	0.944	0.829
CP3 [82]	0.94	0.74	0.75	0.77	0.84	0.74	0.82	0.82	0.72	0.87	0.77	0.84	0.97	0.85	0.90	0.93	0.814
GTNet [80]	0.942	0.744	0.757	0.782	0.839	0.749	0.849	0.818	0.722	0.894	0.786	0.847	0.979	0.835	0.893	0.950	0.823
CompletionMamba	**0.952**	**0.774**	**0.777**	**0.816**	**0.864**	**0.785**	**0.879**	**0.832**	**0.757**	**0.916**	**0.841**	**0.856**	**0.982**	**0.858**	**0.907**	**0.962**	**0.847**

Adam optimization method [83] with $\beta_1 = 0.9$ and $\beta_2 = 0.999$ for training, running for 30 epochs on the MVP dataset and 400 epochs on the PCN dataset. The learning rate was set to 10^{-4}, and was reduced by 30% every 20 epochs. All tests were conducted on an NVIDIA 3090Ti GPU.

B. Completion on the MVP Dataset

We assessed the MVP dataset against a variety of leading-edge baseline methods using the $\mathcal{L}_2$ Chamfer Distance and F-Score@1% (F1 is the harmonic average of the accuracy and the completeness) as performance metrics. The results for the baseline methods [27], [29], [32], [44], [54] were produced from the codes and pre-trained models available in their respective official Github projects. The results for the remaining methods were taken directly from [44] and [54] and the original publication [60], [80], [82].

1) Quantitative Comparison: The performance of all methods, measured by CD loss and F-score@1%, is reported in Tables I and II. The proposed CompletionMamba achieves comparable CD performance to GTNet [80] (see Table I), while demonstrating superior F-score@1% across all categories (see Table II). Although GTNet [80] effectively captures global shape priors through repetitive geometric structures (e.g., symmetry, rotation, translation, and uniform scaling), its capacity for recovering fine-grained local details is limited. Our LSM block addresses this by jointly modeling local and global geometric dependencies, resulting in better performance in terms of F-score@1%. Moreover, the proposed CompletionMamba is significantly lighter than GTNet (2.34 M VS 11.2 M), offering a more efficient solution. These comprehensive experiments robustly validate the effectiveness of our proposed framework.

We also evaluated our method against others that support multi-resolution completion, as shown in Table III. In this comparison, CompletionMamba demonstrated superior and comparable performance over all the compared methods at multiple resolutions. The potential reason is that our proposed LSM block can effectively establish short and long range dependencies for both coarse and refined predictions, thereby promoting the completion performance.

2) Qualitative Comparison: Visual comparison, as displayed in Fig. 4, shows that CompletionMamba can produce

TABLE III

SHAPE COMPLETION RESULTS ON MULTI-VIEW PARTIAL POINT CLOUD (MVP) DATASET WITH VARIOUS POINT CLOUD RESOLUTIONS IN TERMS OF L_2 CHAMFER DISTANCE $\times 10^4$

#Points	2048		4096		8192		16384	
	CD↓	F1↑	CD↓	F1↑	CD↓	F1↑	CD↓	F1↑
PCN [19]	9.77	0.320	7.96	0.458	6.99	0.563	6.02	0.638
TopNet [21]	10.11	0.308	8.20	0.440	7.00	0.533	6.36	0.601
MSN [22]	7.90	0.432	6.17	0.585	5.42	0.659	4.90	0.710
ECG [24]	6.64	0.476	5.41	0.585	4.18	0.690	3.58	0.753
GRNet [25]	7.61	0.353	5.73	0.493	4.51	0.616	3.54	0.700
VRCNet [44]	5.96	0.499	4.70	0.636	3.64	0.727	3.12	0.791
PoinTr [31]	5.79	0.499	4.29	0.638	3.52	0.725	2.95	0.783
PMP-Net++ [84]	-	-	-	-	-	-	3.38	0.687
SnowflakeNet [29]	5.71	0.503	4.45	0.648	3.48	0.743	2.69	0.796
PDR [54]	5.66	0.499	4.26	0.649	3.35	0.754	2.61	0.817
LDM [55]	5.28	0.459	3.92	0.665	3.05	0.766	2.58	0.823
AnchorFormer [32]	5.89	0.482	4.35	0.655	3.21	0.763	2.60	0.819
VAPCNet [60]	5.40	0.521	3.96	0.658	3.02	0.763	2.40	0.829
CP3 [82]	5.10	0.526	**3.49**	0.682	3.14	0.756	2.27	0.814
GTNet [80]	5.76	0.510	4.18	-	3.05	-	**2.19**	0.823
CompletionMamba	**4.99**	**0.536**	3.60	**0.688**	**2.79**	**0.783**	2.20	**0.847**

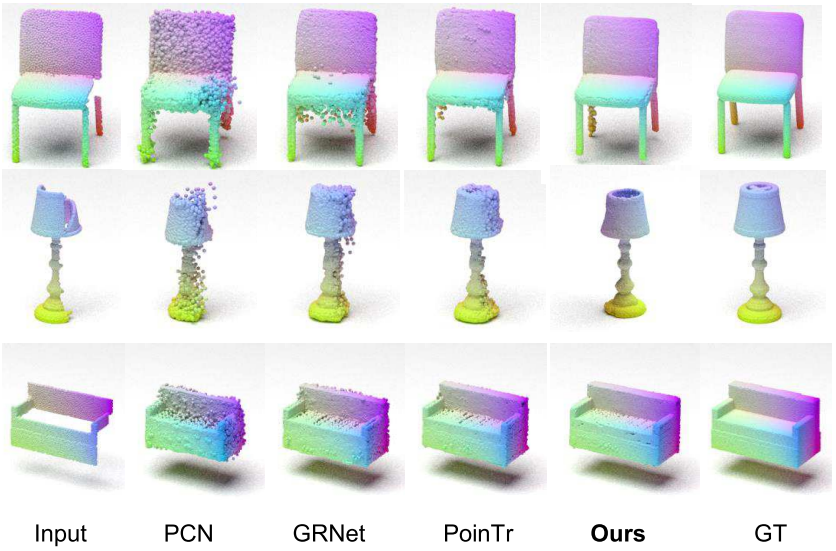


Fig. 4. Visual comparisons on the MVP dataset. Note that, the partial point clouds (2048 points) are sparse and self-occluded, as opposed to the reconstructed and ground truth point clouds (16,384 points), which are dense and complete.

more accurate complete shapes than competing methods. Although PoinTr [31] can preserve the details of input point clouds, it struggles to accurately predict the missing parts of the point cloud. For instance, CompletionMamba can recover the chair's missing legs (first row of Fig. 4) by referencing the visible legs, taking into account the relationships between visible legs and missing legs. However, PoinTr [85] cannot fully recover the missing legs. A possible reason is that the Transformer struggles to establish long-range dependencies between points in an unordered point cloud and lacks the ability to perceive the overall shape. In contrast, the proposed CompletionMamba can better learn local and global structural relationships under the causal point cloud sequence.

C. Completion on the PCN Dataset

On the PCN dataset, we benchmarked our network against SOTA baseline methods. The $\mathcal{L}_1$ Chamfer Distance served as our metric for evaluation. The baseline methods' results of [24] and [53] were produced from the codes and pre-trained models provided in their official Github repositories. We gathered results for other methods from [29], [32], and [86] along with their respective original publications [27], [31], [60].

1) Quantitative Comparison: The data, presented in Table IV, demonstrates that our network achieves the lowest average $\mathcal{L}_1$ Chamfer Distance (CD). In all categories, our method gets comparable or better performances. Especially, compared to 3DmambaComplete, which directly uses Mamba on the unordered point cloud without any modifications, the proposed CompletionMamba achieves significant improvements in all categories due to the establishment of causal sequences for point clouds by the carefully crafted LSM block.

2) Qualitative Comparison: Fig. IV presents the qualitative comparison results. Our method stands out by predicting shapes with greater accuracy and finer details. For example, as shown in the second and third rows of Fig. IV, our approach more effectively restores the complex structures on the legs

TABLE IV

QUANTITATIVE COMPARISON OF SOTA METHODS ON THE PCN DATASET, USING $\mathcal{L}_1$ CHAMFER DISTANCE $\times 10^3$ AS THE EVALUATION METRIC. LOWER $\mathcal{L}_1$ CD VALUES INDICATE BETTER PERFORMANCE

Models	Avg.	airplane	cabinet	car	chair	lamp	couch	table	watercraft
AtlasNet [20]	10.58	6.37	11.94	10.10	12.06	12.37	12.99	10.33	10.61
FoldingNet [23]	14.31	9.49	15.80	12.61	15.55	16.41	15.97	13.65	14.99
PCN [19]	9.64	5.50	22.70	10.63	8.70	11.00	11.34	11.68	8.59
TopNet [21]	12.15	7.61	13.31	10.90	13.82	14.44	14.78	11.22	11.12
GRNet [25]	8.83	6.45	10.37	9.45	9.41	7.96	10.51	8.44	8.04
Wang et. al. [26]	8.51	4.79	9.97	8.31	9.49	8.94	10.69	7.81	8.05
PMP-Net [86]	8.73	5.65	11.24	9.64	9.51	6.95	10.83	8.72	7.25
ECG [24]	8.63	5.23	10.12	8.36	9.43	8.53	10.94	7.98	8.16
NSFA [53]	8.32	5.03	10.51	9.11	9.16	7.45	10.46	7.56	7.28
SK-PCN [87]	8.49	5.09	9.98	8.22	9.29	8.39	10.80	7.84	8.02
PoinTr [31]	8.38	4.75	10.47	8.68	9.39	7.75	10.93	7.78	7.29
SnowflakeNet [44]	7.21	4.29	9.16	8.08	7.89	6.07	9.23	6.55	6.40
Wang et. al [27]	7.96	-	-	-	-	-	-	-	-
VAPCNet [60]	7.02	4.10	9.28	8.15	7.51	5.55	9.18	6.28	6.10
Anchorformer [32]	6.59	3.70	8.94	7.57	7.05	5.21	8.40	6.03	5.81
3DMambaComplete [43]	6.91	3.86	9.12	7.72	7.41	5.73	9.04	6.29	6.09
CP3 [82]	7.02	4.34	9.02	7.90	7.41	6.35	8.52	6.32	6.26
GTNet [80]	7.15	4.17	9.33	8.38	7.66	5.49	9.44	6.69	6.07
AdaPoinTr [85]	6.53	3.68	**8.82**	**7.47**	**6.85**	5.47	8.35	5.80	5.76
CompletionMamba	**6.50**	**3.60**	8.88	7.50	7.02	**5.11**	**8.28**	**5.78**	**5.68**

TABLE V

PERFORMANCE COMPARISON IN TERMS OF L_2 CHAMFER DISTANCE $\times 10^3$ (CD_{L_2}) AND F-SCORE@1% (F1) ON THE SHAPENET-55 DATASET. THE PER-CATEGORY L_2 CHAMFER DISTANCE RESULTS ARE REPORTED ON 5 CATEGORIES WITH MOST TRAINING SAMPLES AND 5 CATEGORIES WITH THE LEAST TRAINING SAMPLES. CD_{L_2}-S, CD_{L_2}-M AND CD_{L_2}-H DENOTE THE L_2 CHAMFER DISTANCE ON THE MASKED POINT CLOUD WITH THE RATIO OF 25%, 50% AND 75%, RESPECTIVELY. CD_{L_2} AND F1 ARE THE AVERAGED RESULTS ON ALL CATEGORIES AND ALL DIFFICULTIES. (LOWER CD_{L_2} AND HIGHER F1 ARE BETTER)

Models	Table	Chair	airplane	car	sofa	Birdhouse	Bag	Remote	Keyboard	Rocket	CD_{L_2}-S	CD_{L_2}-M	CD_{L_2}-H	CD_{L_2}	F1
FoldingNet [23]	2.53	2.81	1.43	1.98	2.48	4.71	2.79	1.44	1.24	1.48	2.67	2.66	4.05	3.12	0.082
TopNet [21]	2.21	2.53	1.14	2.18	2.36	4.83	2.93	1.49	0.95	1.32	2.26	2.16	4.30	2.91	0.126
PCN [19]	2.13	2.29	1.02	1.85	2.06	4.50	2.86	1.33	0.89	1.32	1.94	1.96	4.08	2.66	0.133
PoinTr [31]	0.81	0.95	0.44	0.91	0.79	1.86	0.93	0.53	0.38	0.57	0.58	0.88	1.79	1.09	0.464
AdaPoinTr [85]	0.62	0.69	0.33	0.81	0.63	1.33	0.68	0.38	0.33	0.34	0.49	0.69	1.24	0.81	0.503
SeedFormer [30]	0.72	0.81	0.40	0.89	0.71	1.51	0.79	0.46	0.36	0.50	0.50	0.77	1.49	0.92	0.472
SVDFormer [88]	-	-	-	-	-	-	-	-	-	-	0.48	0.70	1.30	0.83	0.451
AnchorFormer [32]	0.58	0.67	0.33	0.69	0.58	1.35	0.64	0.36	0.27	0.42	0.41	0.61	1.26	0.76	**0.558**
CompletionMamba	**0.52**	**0.60**	**0.28**	**0.62**	**0.52**	**1.29**	**0.55**	**0.31**	**0.23**	**0.20**	**0.36**	**0.56**	**1.19**	**0.70**	0.555

of the tables, while the reconstructions from other methods appear significantly noisier. This highlights our network's ability to refine the shape with localized details.

D. Completion on the ShapNet-55/34 Dataset

Following AnchorFormer [32], we report the average CD_{L_2} and F1 scores across all categories, along with the CD_{L_2} performance on masked point cloud data under three different occlusion ratios (CD_{L_2}-S, CD_{L_2}-M, and CD_{L_2}-H). Additionally, we evaluate per-category CD_{L_2} for two subsets (high-data categories (table, chair, plane, car, sofa; >2,500 training samples) and low-data categories (birdhouse, bag, remote, keyboard, rocket; <80 training samples)). From Table V, we can see that our CompletionMamba consistently outperforms competing methods.

To evaluate the generalization capability of CompletionMamba for novel object shape completion, we conducted experiments on ShapeNet-34, measuring performance via CD_{L_2} and F-Score @1% for both seen and unseen categories. As shown in Table VI, CompletionMamba achieves performance comparable to or better than previous state-of-the-art methods. This improvement can be attributed to the model's enhanced capacity for joint global-local structure modeling, which enables more robust shape completion under varying data conditions.

E. Completion on the KITTI Dataset

To demonstrate the effectiveness of our method in real-world scenarios, we fine-tuned our trained model on ShapeNetCars [19] following the approach in [25]. We then evaluated our model's performance on the KITTI dataset [89], which features incomplete point clouds of cars captured in real-world scenes via LiDAR scans. We assessed the performance using two key metrics: 1/ **Fidelity:** This metric calculates the average distance from each point in the input to its nearest neighbor in the output. It serves as a measure of how well the input is preserved in the output. 2/ **Minimal Matching Distance (MMD):** This is determined by the Chamfer Distance (CD) between the output and the closest car point cloud from ShapeNet in terms of CD. MMD assesses how closely the output resembles a typical car shape. These metrics are reported in Table VII, where our method shows superior quantitative performance compared to other approaches.

TABLE VI

PERFORMANCE COMPARISON IN TERMS OF L_2 CD $\times 10^3$ (CD_{L_2}) AND F-SCORE@ 1% (F1) ON THE SHAPENET-34 DATASET. THE L_2 CD PERFORMANCES ON BOTH OF THE 34 SEEN CATEGORIES AND 21 UNSEEN CATEGORIES ARE REPORTED. CD_{L_2}-S, CD_{L_2}-M AND CD_{L_2}-H DENOTE THE L_2 CD ON THE MASKED POINT CLOUD WITH A RATIO OF 25%, 50% AND 75%, RESPECTIVELY. CD_{L_2} AND F1 ARE THE AVERAGED RESULTS ON CORRESPONDING CATEGORY SUBSET (SEEN/UNSEEN) ACROSS ALL DIFFICULTIES. (LOWER CD_{L_2} AND HIGHER F1 ARE BETTER). THE BOLD VALUES ARE THE BEST

Models	34 seen categories					21 unseen categories				
	CD_{L_2}-S ↓	CD_{L_2}-M ↓	CD_{L_2}-H ↓	CD_{L_2} ↓	F1↑	CD_{L_2}-S ↓	CD_{L_2}-M ↓	CD_{L_2}-H ↓	CD_{L_2} ↓	F1↑
FoldingNet [23]	1.86	1.81	3.38	2.35	0.139	2.76	2.74	5.36	3.62	0.095
TopNet [21]	1.77	1.61	3.54	2.31	0.171	2.62	2.43	5.44	3.50	0.121
PCN [19]	1.87	1.81	2.97	2.22	0.154	3.17	3.08	5.29	3.85	0.101
PoinTr [31]	0.76	1.05	1.88	1.23	0.421	1.04	1.67	3.44	2.05	0.384
AdaPoinTr [85]	0.48	0.63	1.07	0.73	0.469	0.61	0.96	2.11	1.23	0.416
SVDFormer [88]	0.46	0.65	1.13	0.75	0.457	0.61	1.05	2.19	1.28	0.427
AnchorFormer [32]	0.41	0.57	1.12	0.70	**0.564**	0.52	0.90	2.16	1.19	**0.535**
CompletionMamba	**0.35**	**0.52**	**1.06**	**0.64**	0.560	**0.44**	**0.73**	**2.01**	**1.04**	0.531

TABLE VII

PERFORMANCE RESULTS ON LIDAR SCANS FROM THE KITTI DATASET ARE EVALUATED USING THE FIDELITY AND MMD (MINIMAL MATCHING DISTANCE) METRICS. THE MODEL WAS FINE-TUNED ON PCN CARS, FOLLOWING [25]

CD-l_2 ($\times 10^3$)	AtlasNet [20]	PCN [19]	FoldingNet [23]	TopNet [21]	MSN [22]	NSFA [53]	PFNet [90]	GRNet [25]	**CompletionMamba**
Fidelity↓	1.759	2.235	7.467	5.354	0.434	1.281	1.137	0.816	**0.150**
MMD↓	2.108	1.366	0.537	0.636	2.259	0.891	0.792	0.568	**0.512**

TABLE VIII

ABLATION STUDIES FOR COMPLETIONMAMBA, EXAMINING THE EFFECTIVENESS OF THE REORDERING (*Model*2), THE LOCAL MAMBA BLOCK (LM) (*Model*3 AND *Model*5), THE SHAPE-AWARE MAMBA BLOCK (SM) (*Model*4), LSM IN THE ENCODER (ELSM) AND LSM IN THE DECODER (DLSM). 'VM', 'CD' AND 'F1' DENOTE VANILLA MAMBA, $\mathcal{L}_2$ CD (MULTIPLIED BY 10^4) AND F SCORE @1%

Model	Description	CD ↓	F1 ↑
0	Baseline	6.40	0.469
1	Baseline + VM	6.26	0.476
2	Baseline + VM + reordering	5.82	0.487
3	Baseline + reordering + LM	5.45	0.501
4	Baseline + reordering + SM	5.54	0.513
5	Baseline + reordering + LM + VM	5.15	0.523
6	CompletionMamba w/o ELSM	5.65	0.502
7	CompletionMamba w/o DLSM	5.71	0.499
8	Baseline + PointMamba [40]	5.92	0.480
9	**CompletionMamba**	**4.99**	**0.536**

V. ABLATION STUDY

This section analyzes the performance of the proposed LSM block in both the encoder and decoder, specifically targeting the MVP dataset [44] and working with point clouds consisting of 2048 points. First, we evaluated the impact of the reordering operation. Then, we separately tested the effectiveness of our local Mamba module and shape-aware Mamba module. Next, we evaluated the impact of LSM block on both the encoder and decoder. Finally, we analyzed the complexity of the proposed model. To quantify the performance and improvements brought by these modifications, we employed CD and F-Score @1% threshold as our evaluation metrics.

Baseline: To better assess the effectiveness of the proposed modules, we first built the baseline structure consisting of patch embedding, seed generator, feature interpolation, and upsampling modules (*Model*0). *In the baseline, Mamba blocks are not incorporated.* The performance of the baseline is shown in Table VIII.

A. Vanilla Mamba

To test the impact of the direct usage of vanilla Mamba to unordered point cloud completion, we incorporated it after both patch embedding module and feature interpolation module, respectively (*Model*1). From Table VIII, we can see that the performance is slightly improved by introducing the vanilla Mamba. This is because Mamba struggles to model long-range dependencies in non-causal point clouds.

B. Reordering

To demonstrate the impact of the proposed reordering strategy, the randomly ordered point cloud and point-wise features are reordered along x-axis, y-axis and z-axis. Then, we used the Mamba block to successively process the reordered points and features (*Model*2). Compared to the performance of *Model*1, the reordered structure improved the completion performance both in terms of CD and F-Score @1%. This comparison indicates the effectiveness of the proposed reordering strategy and also demonstrates that the Mamba block is adept at handling causal data.

C. Local Mamba Block

To validate the effectiveness of the proposed local Mamba block, we designed two experiments. Firstly, we directly replaced the vanilla Mamba of *Model*2 with the proposed local Mamba to form *Model*3. Secondly, we added the local Mamba block to *Model*2 to form *Model*5. *Model*3 outperforms *Model*2 in terms of both CD and F1 metrics. This enhanced performance can be attributed to the proposed Local Mamba Block that effectively models the relationship between local points in the 3D space. This avoids the substantial variations in coordinate positions that can occur in the adjacent points of the reordered point cloud sequence when using the vanilla Mamba. *Model*5 integrates the proposed Local Mamba

Fig. 5. Visual comparisons on the PCN dataset. Note that the partial point clouds (2048 points) are sparse and self-occluded, as opposed to the reconstructed and ground truth point clouds (16,384 points) which are dense and complete.

block and vanilla mamba, leading to notable improvements in results. This underscores the complementary strengths of these two modules in optimising point cloud completion.

D. Shape-aware Mamba Block

To evaluate the impact of the proposed shape-aware Mamba block, we directly replaced the vanilla Mamba of the *Model*2 with the proposed shape-aware Mamba, forming *Model*4. By comparing *Model*2 with *Model*4, we find that the proposed shape-aware Mamba block significantly promotes point cloud completion, especially in F-Score @1%. This is because the mechanism of the Mamba allows shape codes to be propagated to each individual point, thereby obtaining a better perception of the whole geometric structure.

E. LSM in the Encoder

To further test the effectiveness of the proposed LSM block in the encoder, we removed this module from the encoder of CompletionMamba, forming *Model*6. Then, we directly used the global feature g_f and P_f extracted from patch embedding to predict missing points. From the result in Table VIII, we can see that the performance experiences a significant drop in terms of CD and F-Score @1%, from 4.99 and 0.536 to 5.65 and 0.502, respectively. This demonstrates that the proposed LSM block can effectively build the relationships between the observed points and the missing points.

F. LSM in the Decoder

To validate the impact of the proposed LSM block in the decoder, we directly removed it from the decoder of CompletionMamba, forming *Model*7. From Table VIII, we can conclude that the proposed LSM block has a significant impact on the performance of point cloud completion. This is because the proposed LSM block can effectively model the overall shape of the point cloud based on the input coarse prediction, thereby facilitating detailed reconstruction.

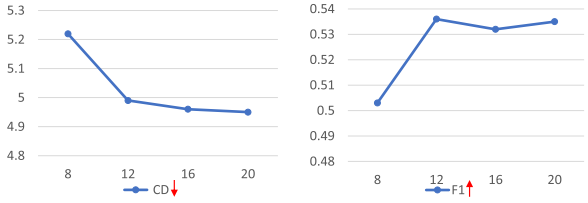


Fig. 6. The impact of k in KNN.

TABLE IX
COMPLEXITY ANALYSIS ON THE PCN DATASET [19], WITH THE NUMBER OF PARAMS, FLOPS AND INFERENCE TIME (MS) AS REFERENCES. THE BOLD VALUES ARE THE BEST

Models	Params↓	FLOPs ↓	Time	CD ↓
GRNet [25]	76.71 M	25.88 G	**11.76 ms**	8.83
PoinTr [31]	42.51 M	11.21 G	14.15 ms	8.38
SeedFormer [30]	3.20 M	29.61 G	43.52 ms	6.74
AnchorFormer [32]	29.05 M	7.60 G	39.46 ms	6.59
Ours	**2.34 M**	**7.54 G**	15.15 ms	**6.50**

G. PointMamba [40] in Point Cloud Completion

To further validate the validity of the proposed LSM block and our reordering strategy, we replace them with the Mamba block in PointMamba [40] (*Model*8). The comparison between *Model*2 and *Model*8 shows the effectiveness of our reordering strategy. Moreover, compared to *Model*8, our Completion-Mamba achieves significant improvement both in CD and F-Score @1%. This is attributed to our LSM block, which can better deal with the reordered point cloud.

H. Impact of k in KNN

We present the impact of k in KNN in terms of CD and F-Score@1% in Fig. 6. Considering the trade-off between performance and computational cost, we set k to 12.

I. Complexity Analysis

We also conducted a complexity analysis in Table IX, demonstrating our superior completion results with smaller

computational cost and memory consumption. Note that, the number of parameters in our method is only 8% of that in AnchorFormer.

VI. Conclusion

In this paper, we presented a novel segmentation-by-tracking approach In this paper, we proposed a novel Mamba-based network, named CompletionMamba, for point cloud completion. Directly applying Mamba to point cloud completion is non-trivial due to its causality constraint and the unordered nature of point clouds. To address this challenge, we proposed a two-stage approach to transform the unordered point cloud into an approximately causal sequence. We first reorganized the point cloud independently along the x-, y-, and z-axes, resulting in three structured point sequences. Then, inspired by the observation that the k nearest neighborhood points of each point remain unchanged in the 3D space regardless of reordering, we proposed the local Mamba module to model local relationships between points within these neighborhoods. This alleviates the impact of spatial inconsistency between adjacent points introduced by axis-based reordering. Furthermore, while the vanilla Mamba is effective at capturing long-range dependencies between reordered points, it lacks the capacity to explicitly model the relationship between points and the whole shape. Thus, we proposed the shape-aware Mamba, treating the shape code as the first hidden state to propagate shape information to each point. By combining local and shape-aware Mamba, CompletionMamba achieves superior performance with lower computational and memory costs on MVP, PCN and ShapeNet-55/34 datasets. The limitation of this work is that our method struggles to complete input point clouds that are missing a large portion of their structure. In this paper, we opted for a simple and scalable sorting strategy. Future work will investigate advanced reordering approaches that balance computational efficiency with stronger preservation of local geometry. Also, we plan to extend CompletionMamba to other point cloud tasks.

References

[1] Y. Guo, H. Wang, Q. Hu, H. Liu, L. Liu, and M. Bennamoun, "Deep learning for 3D point clouds: A survey," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 43, no. 12, pp. 4338–4364, Dec. 2021.

[2] B. Fei et al., "Comprehensive review of deep learning-based 3D point cloud completion processing and analysis," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 12, pp. 22862–22883, Dec. 2022.

[3] Y. Liu and Z. Liu, "Low overlapping point cloud registration using mutual prior based completion network," *IEEE Trans. Image Process.*, vol. 33, pp. 4781–4795, 2024.

[4] J. Mei et al., "Camera-based 3D semantic scene completion with sparse guidance network," *IEEE Trans. Image Process.*, vol. 33, pp. 5468–5481, 2024.

[5] M. Berger et al., "State of the art in surface reconstruction from point clouds," *Eurographics State Art Rep.*, pp. 161–185, 2014.

[6] J. Davis, S. R. Marschner, M. Garr, and M. Levoy, "Filling holes in complex surfaces using volumetric diffusion," in *Proc. 1st Int. Symp. 3D Data Process. Visualizat. Transmiss.*, Jun. 2002, pp. 428–441.

[7] 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.

[8] 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, Sep. 2013.

[9] M. Pauly, N. J. Mitra, J. Wallner, H. Pottmann, and L. J. Guibas, "Discovering structural regularity in 3D geometry," in *Proc. ACM SIGGRAPH Papers*, Aug. 2008, pp. 1–11.

[10] I. Sipiran, R. Gregor, and T. Schreck, "Approximate symmetry detection in partial 3D meshes," *Comput. Graph. Forum*, vol. 33, no. 7, pp. 131–140, Oct. 2014.

[11] T. Xue, J. Liu, and X. Tang, "3-D modeling from a single view of a symmetric object," *IEEE Trans. Image Process.*, vol. 21, no. 9, pp. 4180–4189, Sep. 2012.

[12] Y. Li and G. Baciu, "HSGAN: Hierarchical graph learning for point cloud generation," *IEEE Trans. Image Process.*, vol. 30, pp. 4540–4554, 2021.

[13] W. Zhao, S. Gao, and H. Lin, "A robust hole-filling algorithm for triangular mesh," *Vis. Comput.*, vol. 23, no. 12, pp. 987–997, Nov. 2007.

[14] V. Blanz and T. Vetter, "A morphable model for the synthesis of 3D faces," in *Proc. 26th Annu. Conf. Comput. Graph. Interact. Techn. (SIGGRAPH)*, 1999, pp. 187–194.

[15] D. Forsyth, "Object detection with discriminatively trained part-based models," *Computer*, vol. 47, no. 2, pp. 6–7, Feb. 2014.

[16] S. Gupta, P. Arbeláez, R. Girshick, and J. Malik, "Aligning 3D models to RGB-D images of cluttered scenes," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 4731–4740.

[17] G. Li, L. Liu, H. Zheng, and N. J. Mitra, "Analysis, reconstruction and manipulation using arterial snakes," *ACM Trans. Graph.*, vol. 29, no. 6, pp. 1–10, Dec. 2010.

[18] J. Rock, T. Gupta, J. Thorsen, J. Gwak, D. Shin, and D. Hoiem, "Completing 3D object shape from one depth image," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 2484–2493.

[19] 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.

[20] T. Groueix, M. Fisher, V. G. Kim, B. C. Russell, and M. Aubry, "A papier-mache approach to learning 3D surface generation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 216–224.

[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)*, Long Beach, CA, USA, Jun. 2019, pp. 383–392.

[22] M. Liu, L. Sheng, S. Yang, J. Shao, and S. Hu, "Morphing and sampling network for dense point cloud completion," in *Proc. AAAI Conf. Artif. Intell.*, vol. 34, 2020, pp. 11596–11603.

[23] 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.

[24] L. Pan, "ECG: Edge-aware point cloud completion with graph convolution," *IEEE Robot. Autom. Lett.*, vol. 5, no. 3, pp. 4392–4398, Jul. 2020.

[25] 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.* Cham, Switzerland: Springer, 2020, pp. 365–381.

[26] 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.

[27] Y. Wang, D. J. Tan, N. Navab, and F. Tombari, "Learning local displacements for point cloud completion," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 1558–1567.

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

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

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

[31] 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.

[32] 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.

[33] 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.

[34] C. R. Qi, Y. Li, H. Su, and L. Guibas, "PointNet ++: Deep hierarchical feature learning on point sets in a metric space," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, pp. 5105–5111.

[35] 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, Oct. 2019.

[36] 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.

[37] A. Gu and T. Dao, "Mamba: Linear-time sequence modeling with selective state spaces," 2023, *arXiv:2312.00752*.

[38] M. M. Islam and G. Bertasius, "Long movie clip classification with state-space video models," in *Proc. Eur. Conf. Comput. Vis.*, 2022, pp. 87–104.

[39] J. Wang et al., "Selective structured state-spaces for long-form video understanding," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2023, pp. 6387–6397.

[40] D. Liang et al., "PointMamba: A simple state space model for point cloud analysis," 2024, *arXiv:2402.10739*.

[41] T. Zhang et al., "Point cloud mamba: Point cloud learning via state space model," 2024, *arXiv:2403.00762*.

[42] J. Liu et al., "Point mamba: A novel point cloud backbone based on state space model with octree-based ordering strategy," 2024, *arXiv:2403.06467*.

[43] Y. Li, W. Yang, and B. Fei, "3DMambaComplete: Exploring structured state space model for point cloud completion," 2024, *arXiv:2404.07106*.

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

[45] C. B. Choy, D. Xu, J. Gwak, K. Chen, and S. Savarese, "3D-R2N2: A unified approach for single and multi-view 3D object reconstruction," in *Proc. Eur. Conf. Comput. Vis.* Cham, Switzerland: Springer, 2016, pp. 628–644.

[46] R. Girdhar, D. F. Fouhey, M. Rodriguez, and A. Gupta, "Learning a predictable and generative vector representation for objects," in *Proc. Eur. Conf. Comput. Vis.*, Amsterdam, The Netherlands, Sep. 2016, pp. 484–499.

[47] A. Dai, C. R. Qi, and M. Nießner, "Shape completion using 3D-encoder-predictor CNNs and shape synthesis," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 5868–5877.

[48] X. Han, Z. Li, H. Huang, E. Kalogerakis, and Y. Yu, "High-resolution shape completion using deep neural networks for global structure and local geometry inference," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 85–93.

[49] Z. Wu et al., "3D ShapeNets: A deep representation for volumetric shapes," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 1912–1920.

[50] L. Zhao, Y. Wei, J. Li, J. Zhou, and J. Lu, "Structure-aware cross-modal transformer for depth completion," *IEEE Trans. Image Process.*, vol. 33, pp. 1016–1031, 2024.

[51] H. Su et al., "SPLATNet: Sparse lattice networks for point cloud processing," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 2530–2539.

[52] P.-S. Wang, Y. Liu, Y.-X. Guo, C.-Y. Sun, and X. Tong, "O-CNN: Octree-based convolutional neural networks for 3D shape analysis," *ACM Trans. Graph.*, vol. 36, no. 4, pp. 1–11, Aug. 2017.

[53] W. Zhang, Q. Yan, and C. Xiao, "Detail preserved point cloud completion via separated feature aggregation," in *Proc. Eur. Conf. Comput. Vis.* Cham, Switzerland: Springer, 2020, pp. 512–528.

[54] 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*.

[55] Z. Chen et al., "Learning 3D shape latent for point cloud completion," *IEEE Trans. Multimedia*, vol. 26, pp. 8717–8729, 2024.

[56] J. Ho, A. Jain, and P. Abbeel, "Denoising diffusion probabilistic models," in *Proc. NIPS*, vol. 33. Vancouver, BC, Canada: Curran Associates, 2020, pp. 6840–6851.

[57] 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.

[58] 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.

[59] J. Shao, *Mathematical Statistics*. Cham, Switzerland: Springer, 2003.

[60] Z. Fu et al., "VAPCNet: Viewpoint-aware 3D point cloud completion," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2023, pp. 12074–12084.

[61] A. Gu, T. Dao, S. Ermon, A. Rudra, and C. Ré, "HiPPO: Recurrent memory with optimal polynomial projections," in *Proc. Adv. Neural Inf. Process. Syst.*, 2020, pp. 1474–1487.

[62] A. Gu et al., "Combining recurrent, convolutional, and continuous-time models with linear state space layers," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 34, 2021, pp. 572–585.

[63] A. Gu, K. Goel, and C. Ré, "Efficiently modeling long sequences with structured state spaces," *arXiv preprint arXiv:2111.00396*, 2021.

[64] H. Mehta, A. Gupta, A. Cutkosky, and B. Neyshabur, "Long range language modeling via gated state spaces," 2022, *arXiv:2206.13947*.

[65] J. T. H. Smith, A. Warrington, and S. W. Linderman, "Simplified state space layers for sequence modeling," 2022, *arXiv:2208.04933*.

[66] M. Pióro et al., "MoE-mamba: Efficient selective state space models with mixture of experts," 2024, *arXiv:2401.04081*.

[67] J. Wang, T. Gangavarapu, J. N. Yan, and A. M. Rush, "MambaByte: Token-free selective state space model," 2024, *arXiv:2401.13660*.

[68] C. Wang, O. Tsepa, J. Ma, and B. Wang, "Graph-mamba: Towards long-range graph sequence modeling with selective state spaces," 2024, *arXiv:2402.00789*.

[69] T. Guo et al., "MambaMorph: A mamba-based framework for medical MR-CT deformable registration," 2024, *arXiv:2401.13934*.

[70] J. Liu et al., "Swin-UMamba: Mamba-based UNet with ImageNet-based pretraining," 2024, *arXiv:2402.03302*.

[71] J. Ruan, J. Li, and S. Xiang, "VM-UNet: Vision mamba UNet for medical image segmentation," 2024, *arXiv:2402.02491*.

[72] Z. Ye, T. Chen, F. Wang, H. Zhang, and L. Zhang, "P-mamba: Marrying perona malik diffusion with mamba for efficient pediatric echocardiographic left ventricular segmentation," 2024, *arXiv:2402.08506*.

[73] J. Ma, F. Li, and B. Wang, "U-mamba: Enhancing long-range dependency for biomedical image segmentation," 2024, *arXiv:2401.04722*.

[74] Z. Xing, T. Ye, Y. Yang, G. Liu, and L. Zhu, "SegMamba: Long-range sequential modeling mamba for 3D medical image segmentation," 2024, *arXiv:2401.13560*.

[75] Z. Zheng and J. Zhang, "FD-vision mamba for endoscopic exposure correction," 2024, *arXiv:2402.06378*.

[76] Y. Yang, Z. Xing, L. Yu, C. Huang, H. Fu, and L. Zhu, "Vivim: A video vision mamba for medical video segmentation," 2024, *arXiv:2401.14168*.

[77] Q. Zhou et al., "3DMambaIPF: A state space model for iterative point cloud filtering via differentiable rendering," 2024, *arXiv:2404.05522*.

[78] F. Chollet, "Xception: Deep learning with depthwise separable convolutions," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 1251–1258.

[79] D. Hendrycks and K. Gimpel, "Gaussian error linear units (GELUs)," 2016, *arXiv:1606.08415*.

[80] S. Zhang et al., "Learning geometric transformation for point cloud completion," *Int. J. Comput. Vis.*, vol. 131, no. 9, pp. 2425–2445, Sep. 2023.

[81] X. Wen, Z. Han, Y.-P. Cao, P. Wan, W. Zheng, and Y.-S. Liu, "Cycle4Completion: Unpaired point cloud completion using cycle transformation with missing region coding," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 13075–13084.

[82] M. Xu, Y. Wang, Y. Liu, T. He, and Y. Qiao, "CP3: Unifying point cloud completion by pretrain-prompt-predict paradigm," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 45, no. 8, pp. 9583–9594, Aug. 2023.

[83] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2014, *arXiv:1412.6980*.

[84] 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.

[85] X. Yu, Y. Rao, Z. Wang, J. Lu, and J. Zhou, "AdaPoinTr: Diverse point cloud completion with adaptive geometry-aware transformers," 2023, *arXiv:2301.04545*.

[86] 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.

[87] Y. Nie et al., "Skeleton-bridged point completion: From global inference to local adjustment," in *Proc. Adv. Neural Inf. Process. Syst.*, 2020, pp. 16119–16130.

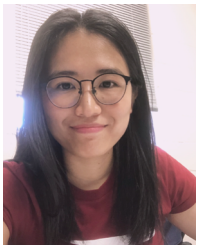
[88] Z. Zhu, H. Chen, X. He, W. Wang, J. Qin, and M. Wei, "SVDFormer: Complementing point cloud via self-view augmentation and self-structure dual-generator," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2023, pp. 14462–14472.

[89] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun, "Vision meets robotics: The KITTI dataset," *Int. J. Robot. Res.*, vol. 32, no. 11, pp. 1231–1237, Sep. 2013.
[90] 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)*, Jun. 2020, pp. 7662–7670.

Zhiheng Fu received the B.E. degree in electrical engineering from Northeastern University (NEU), Shenyang, China, in 2015, the M.E. degree in information and communication engineering from the National University of Defense Technology (NUDT), Changsha, China, in 2018, and the Ph.D. degree from the Department of Computer Science and Software Engineering (CSSE) from The University of Western Australia (UWA) in 2025. He is currently a Postdoctoral Research Fellow with The Hong Kong Polytechnic University (PolyU). His research interests include 3D vision and spatial intelligence.

Jiehua Zhang received the B.E. degree in detection, guidance, and control technology from the University of Electronic Science and Technology of China (UESTC), Chengdu, China, in 2018, and the M.S. degree in aeronautical and astronautical science and technology from the National University of Defense Technology (NUDT), Changsha, China, in 2020. He is currently pursuing the Ph.D. degree in computer science with The University of Oulu, Finland. His research interests include deep learning, computer vision, and efficient network design.

Longguang Wang received the B.E. degree in electrical engineering from Shandong University (SDU), Jinan, China, in 2015, and the Ph.D. degree in information and communication engineering from the National University of Defense Technology (NUDT), Changsha, China, in 2022. His current research interests include low-level vision and 3D vision.



Lian Xu received the Ph.D. degree from the Department of Computer Science and Software Engineering, The University of Western Australia. She is currently a Research Fellow with The University of Western Australia. Her research interests include computer vision and machine learning.

Hamid Laga is currently a Professor with Murdoch University, Australia. His research interests include machine learning, computer vision, computer graphics, and pattern recognition, with a special focus on the 3D reconstruction, modeling and analysis of static and deformable 3D objects, and on machine learning for agriculture and health. He was a recipient of the best paper awards at SGP2017, DICTA2012, and SMI2006.

Yulan Guo (Senior Member, IEEE) received the B.E. and Ph.D. degrees from the National University of Defense Technology (NUDT) in 2008 and 2015, respectively. He is currently a Professor with the School of Electronics and Communication Engineering, Sun Yat-sen University, China. He has authored over 150 articles at highly referred journals and conferences, including IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE and IJCV. His research interests include 3D vision, low-level vision, and machine learning. He served as an Associate Editor for IEEE TRANSACTIONS ON IMAGE PROCESSING, *IET Computer Vision*, *IET Image Processing, Computers and Graphics*, and *The Visual Computer*. He also served as an Area Chair for CVPR 2023/2021, ICCV 2021, ACM Multimedia 2021, and ICPR 2021. He organized eight workshops at prestigious conferences CVPR 2019/2022/2023, ICCV 2021/2023, ECCV 2022, he also edited four special issues for journals IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE, *The Visual Computer*, and *IET Computer Vision* in recent years. He is a Senior Member of ACM.

Farid Boussaid received the M.S. and Ph.D. degrees in microelectronics from the National Institute of Applied Science, Toulouse, France, in 1996 and 1999, respectively. He joined Edith Cowan University, Perth, Australia, as a Postdoctoral Research Fellow and the Visual Information Processing Research Group as a member in 2000. He joined The University of Western Australia, Crawley, Australia, in 2005, where he is currently a Professor. His current research interests include smart CMOS vision sensors and image processing.

Mohammed Bennamoun (Senior Member, IEEE) is currently a Winthrop Professor with the Department of Computer Science and Software Engineering, UWA, and a Researcher of computer vision, machine/deep learning, robotics, and signal/speech processing. He has published four books (available on Amazon), one edited book, one Encyclopedia article, 14 book chapters, more than 160 journal articles, more than 250 conference publications, and 16 invited and keynote publications. His H-index is 75 and his number of citations is more than 29 000 (Google Scholar). He was awarded more than 65 competitive research grants, from Australian Research Council, and numerous other Government, UWA, and industry Research Grants. He successfully supervised more than 26 Ph.D. students to completion. He won the Best Supervisor of the Year Award at QUT in 1998, and received award for research supervision at UWA from 2008 and 2016 and the Vice-Chancellor Award for Mentorship in 2016. He delivered conference tutorials at major conferences, including: IEEE Computer Vision and Pattern Recognition (CVPR 2016), Interspeech 2014, IEEE International Conference on Acoustics Speech and Signal Processing (ICASSP), and European Conference on Computer Vision (ECCV). He was also invited to give a Tutorial at an International Summer School on Deep Learning (DeepLearn 2017).