

Point Cloud Mamba: Point Cloud Learning via State Space Model

Tao Zhang^{1,2 *}, Haobo Yuan¹, Lu Qi¹, Jiangning Zhang⁴, Qianyu Zhou⁵,
Shunping Ji^{1 ‡}, Shuicheng Yan^{2,3}, Xiangtai Li^{2,3 †}

¹ Wuhan University, China

² Skywork AI, Singapore

³ Nanyang Technological University, Singapore

⁴ YouTu Lab, Tencent, China

⁵ Shanghai Jiao Tong University, China

zhang_tao@whu.edu.cn, xiangtai94@gmail.com

Abstract

Recently, state space models have exhibited strong global modeling capabilities and linear computational complexity in contrast to transformers. This research focuses on applying such architecture to more efficiently and effectively model point cloud data globally with linear computational complexity. In particular, for the first time, we demonstrate that Mamba-based point cloud methods can outperform previous methods based on transformer or multi-layer perceptrons (MLPs). To enable Mamba to process 3-D point cloud data more effectively, we propose a novel Consistent Traverse Serialization method to convert point clouds into 1-D point sequences while ensuring that neighboring points in the sequence are also spatially adjacent. Consistent Traverse Serialization yields six variants by permuting the order of x , y , and z coordinates, and the synergistic use of these variants aids Mamba in comprehensively observing point cloud data. Furthermore, to assist Mamba in handling point sequences with different orders more effectively, we introduce point prompts to inform Mamba of the sequence’s arrangement rules. Finally, we propose positional encoding based on spatial coordinate mapping to inject positional information into point cloud sequences more effectively. Point Cloud Mamba surpasses the state-of-the-art (SOTA) point-based method PointNeXt and achieves new SOTA performance on the ScanObjectNN, ModelNet40, ShapeNetPart, and S3DIS datasets. It is worth mentioning that when using a more powerful local feature extraction module, our PCM achieves 79.6 mIoU on S3DIS, significantly surpassing the previous SOTA models, DeLA and PTv3, by 5.5 mIoU and 4.9 mIoU, respectively.

Code — <https://github.com/zhang-tao-whu/PCM>

Introduction

Point cloud analysis (Qi et al. 2017a,b; Li et al. 2018; Zhao et al. 2021; Qian et al. 2022; Wu, Qi, and Fuxin 2019; Qian

*This work was performed when Tao Zhang was an Intern at Skywork AI.

† Project Leader.

‡ Corresponding Author.

Copyright © 2025, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.

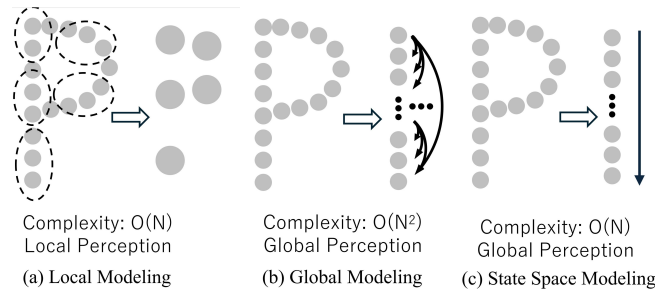


Figure 1: Several pipelines of point cloud modeling. (a) denotes point-based methods with only local perception, including point-based methods, such as PointNet (Qi et al. 2017a), PointNet++ (Qi et al. 2017b), PointMLP (Ma et al. 2022), and PointNeXt (Qian et al. 2022). (b) is the transformer-based method with global perception but quadratic computational cost, including Point Transformer (Zhao et al. 2021) and Point-MAE (Pang et al. 2022). (c) represents Mamba-based methods, which offer advantages of global modeling and linear computational complexity.

et al. 2022) has become a popular topic in 3D understanding and has drawn attention from the research community. Unlike 2D image processing, point clouds are composed of unordered and irregular point sets, making it difficult to apply the 2D image processing methods directly. Thus, recent deep-learning-based approaches (Qi et al. 2017a,b; Li et al. 2018; Wu, Qi, and Fuxin 2019; Ma et al. 2022; Qian et al. 2022; Zhao et al. 2021; Guo et al. 2021) propose using various methods, such as voxel-based and point-based, for point cloud representation. The representative works, Point-Net (Qi et al. 2017a) and Point-Net++ (Qi et al. 2017b), adopt the MLP-based design with deep hierarchical local priors, as shown in Figure 1 (a). After that, many research works (Ma et al. 2022; Boulch 2020; Shi and Rajkumar 2020; Wang et al. 2019; Zhao et al. 2021; Guo et al. 2021; Wu et al. 2022, 2024) focus on advanced local geometric modeling via convolution, graph modeling, or attention.

Meanwhile, with the rapid progress of vision transformers, several works (Zhao et al. 2021; Guo et al. 2021; Wu et al.

2022, 2024) enhance the global modeling with transformer structure in point cloud, as shown in Figure 1 (b). In addition, transformer architectures also work effectively in mask point pre-training, 3D segmentation, and in-context learning. However, the computation and memory costs are still huge. Recently, state space models (Gu and Dao 2023; Gu, Goel, and Ré 2022) (SSMs) have been proven to model long-range dependency in sequential data. In particular, Mamba (Gu and Dao 2023) is proven effective as Transformer (Vaswani et al. 2017) for several challenging NLP tasks. After that, recent works (Zhu et al. 2024; Liu et al. 2024; Li, Singh, and Grover 2024; Behrouz and Hashemi 2024; Ma, Li, and Wang 2024; Xing et al. 2024; Yang, Xing, and Zhu 2024; Ruan and Xiang 2024; He et al. 2024) explore SSMs in various vision tasks, including image representation learning, medical segmentation, and low-level vision tasks. One concurrent work, PointMamba, uses the Mamba layer to model the global context. However, there are still significant performance *gaps* between PointMamba and previous point-based methods.

In this work, we ask an essential question: Can we design an efficient point cloud analysis architecture using Mamba and surpass the performance of point-based and transformer-based methods? In particular, we introduce the Point Cloud Mamba (PCM), a combining local and global modeling framework that outperforms the SOTA point-based and transformer-based methods.

PCM utilizes Mamba architecture to model the global features of point clouds while maintaining linear computational complexity. To process 3-D point cloud data effectively by Mamba layers, we propose a novel Consistent Traverse Serialization (CTS) method to serialize point clouds into a 1-D point sequence while ensuring that neighboring points in the sequence are also adjacent in space. Then, CTS can easily derive six variants by simply permuting the order of x , y , and z coordinates. Additionally, when these six variants of CTS are combined, Mamba layers can more effectively model point cloud features. This is because the different variants provide various perspectives of the point cloud, ensuring that spatially adjacent points are also adjacent in a serialized point sequence. To help Mamba handle specific point sequences better, we introduce *order* prompts to provide Mamba with the arrangement rules of the current point sequence. Finally, we propose simple spatial coordinate mapping as positional embedding for points, more suitable for irregular point cloud data than RoPE (Su et al. 2024) and learnable embedding.

Thanks to the above improvements, we successfully introduced Mamba into Point Cloud analysis and obtained Point Cloud Mamba (PCM). PCM outperforms the SOTA point-based method PointNeXt on three datasets: ScanObjectNet (Uy et al. 2019), ModelNet40 (Wu et al. 2015), and ShapeNetPart (Yi et al. 2016). When enhancing the local feature extraction layers, PCM achieved 79.6 mIoU on the S3DIS dataset, significantly surpassing the previous SOTA PTV3 (Wu et al. 2024) by 4.9 mIoU.

In summary, we have the following contributions: 1) We introduce Mamba into point cloud analysis and construct a combined local and global modeling framework named Point Cloud Mamba. 2) We propose consistent traverse serialization, order prompts, and positional encoding based on

spatial coordinate mapping to assist Mamba in better handling point cloud data. 3) Point Cloud Mamba is the first Mamba-based method that works well in point cloud analysis. It outperforms the SOTA point-based method PointNeXt and transformer-based method PTV3 on ScanObjectNet, ModelNet40, ShapeNetPart, and S3DIS datasets.

Related Work

3D Point Cloud Classification. Recent works have used deep neural networks to process 3D point clouds. In particular, representative works, PointNet (Qi et al. 2017a) and PointNet++ (Qi et al. 2017b), are the pioneering point-based approaches to handle the point clouds using MLPs directly. Meanwhile, several works explore graph-based modeling to utilize 3D geometric topology. Then, several works (Wu, Qi, and Fuxin 2019; Thomas et al. 2019; Shen et al. 2018; Li et al. 2018; Xu et al. 2021a; Komarichev, Zhong, and Hua 2019; Graham, Engelcke, and Van Der Maaten 2018; Choy, Gwak, and Savarese 2019; Zhu et al. 2021) explore the local geometric features via different kernel modeling. Moreover, several works (Liu et al. 2019; Xu et al. 2021b; Xiang et al. 2021; Ran, Liu, and Wang 2022; Chen et al. 2023b; Jiang et al. 2022; Liu, Cai, and Lee 2022; Xie et al. 2020; Zhang et al. 2021; Ma et al. 2022; Lang et al. 2019; Dai and Nießner 2018; Yan et al. 2022) explore other point cloud architecture designs, including MLPs and transformers. Several works also explore different pre-training methods (Pang et al. 2022; Yu et al. 2022; Sanghi 2020; Sauder and Sievers 2019; Wang and Solomon 2019; Hou et al. 2022; Jiang et al. 2023; Wu et al. 2023b; Zhu et al. 2023; Yang et al. 2023) or in-context abilities (Fang et al. 2023; Wang et al. 2024; Wu et al. 2023a) inspired by the NLP field. Recently, state space models (Gu, Goel, and Ré 2022; Gu and Dao 2023) have achieved significant progress. Compared with transformers, they have advantages in efficient global modeling. A concurrent work (Liang et al. 2024) explores such architecture in point clouds. However, there are still significant performance gaps compared with previous point cloud methods. Our PCM shows that Mamba architecture can achieve comparable or even better results than transformer-based models.

3D Visual Transformers. With the rise of the transformer in 2D version (Dosovitskiy et al. 2021; Li et al. 2023; Carion et al. 2020), several works (Lahoud et al. 2022; Guo et al. 2021; Schult et al. 2023; Sun et al. 2023; Lai et al. 2022; Liu et al. 2023; Wang 2023) also explore transformer architectures in the point cloud. Earlier works (Guo et al. 2021; Zhao et al. 2021) have focused on the point cloud process. PCT (Guo et al. 2021) performs global attention directly to each point, following the ViT (Dosovitskiy et al. 2021). However, it has memory consumption and computational complexity issues. Point Transformer (Zhao et al. 2021) solves this issue by introducing local attention. Then, the updated versions (Wu et al. 2022, 2024) explore the different architectures to improve performance and efficiency. Inspired by these studies, our works combine local point processing and a new traverse serialization strategy, which leads to better results than direct SSM traverse.

Transformer-based methods (Zhao et al. 2021; Guo et al. 2021; Wu et al. 2022, 2024; Lai et al. 2022; Robert,

Raguet, and Landrieu 2023; Park et al. 2023; Wang 2023) use Transformers to model point cloud sequences and have extensively explored a lot of aspects, such as point cloud serialization (Wang 2023; Zhao et al. 2021; Guo et al. 2021; Wu et al. 2022, 2024) and point positional embedding (Wu et al. 2022; Park et al. 2023). Although it’s possible to simply implement a Mamba-based network for point cloud analysis by replacing Transformer layers with Mamba layers, due to the differences between Transformers and Mamba, substantial exploration is still needed to find the most suitable network architecture, serialization method and positional embedding strategy for Mamba-based models. This is precisely the objective of this paper.

State Space Models. Inspired by continuous state space models in control systems, recently, state space models (Gu and Dao 2023; Gu, Goel, and Ré 2022) have been proven to model long-range dependency. In particular, S4 (Gu, Goel, and Ré 2022) proposes to normalize the parameter into the diagonal structure, which results in less computation cost and memory usage. After that, Mamba (Gu and Dao 2023) presents a selection mechanism that leads to better results than transformers. Recently, several works have explored such architecture in different tasks, including image classification (Zhu et al. 2024; Liu et al. 2024; Li, Singh, and Grover 2024), graph modeling (Behrouz and Hashemi 2024), medical segmentation (Ma, Li, and Wang 2024; Xing et al. 2024; Yang, Xing, and Zhu 2024; Ruan and Xiang 2024), and low-level version tasks (He et al. 2024). As a concurrent work, we further prove the potential of SSMs in the 3D point clouds, where we can achieve even better results than previous architectures.

Method

The SSM-based architecture, Mamba (Gu and Dao 2023), is attractive for point cloud representation learning due to its global modeling capability and linear computational complexity. However, Mamba is designed for the causal modeling of 1-D sequences, making it difficult to be directly applied to the modeling of non-causal 3-D point cloud data, posing many challenges to be addressed. This section explores how to effectively integrate Mamba into architectures based on local modeling to capture global features.

Preliminaries

PointMLP formulation. PointMLP (Ma et al. 2022) models the representation of point clouds using simple MLP and expands the receptive field of each point through point cloud downsampling and local feature aggregation. The process can be described by Equ. 1, 2, and 3:

$$f_i^{l+1} = \Phi_2(A(\Phi_1(GAM(\{f_{i,j}^l\})), |j = 1, \dots, K)), \quad (1)$$

$$GAM(\{f_{i,j}^l\}) = \alpha \odot \frac{\{f_{i,j}^l\} - f_i^l}{\sigma + \delta}, \quad (2)$$

$$\sigma = \sqrt{\frac{1}{k \times n \times d} \sum_{i=1}^n \sum_{j=1}^k (f_{i,j}^l - f_i^l)^2}, \quad (3)$$

where Φ represents a network composed of a series of residual MLPs, $\{f_{i,j}^l | j = 1, \dots, k\}$ represents the K neighboring points of f_i^l , A denotes the max-pooling operation. GAM refers to the Geometric Affine Module proposed by PointMLP to enhance local features.

Mamba formulation. The state-space equation can describe a multi-input, multi-output continuous system where the current inputs and states jointly determine the change in the state space of this system:

$$h'(t) = \mathbf{A}h(t) + \mathbf{B}x(t), \quad y(t) = \mathbf{C}h(t), \quad (4)$$

where $x(t)$, $h(t)$, and $y(t)$ are the inputs, states, and outputs of the current system, respectively. \mathbf{A} , \mathbf{B} , and \mathbf{C} are all continuous parameters of the system.

The continuous state-space equation mentioned above can be transformed into a discrete formulation using a timescale parameter Δ based on the zero-order hold rule:

$$\bar{\mathbf{A}} = \exp(\Delta\mathbf{A}), \quad \bar{\mathbf{B}} = (\Delta\mathbf{A})^{-1}(\exp(\Delta\mathbf{A}) - I) \cdot \Delta\mathbf{B}, \quad (5)$$

$$h_t = \bar{\mathbf{A}}h_{t-1} + \bar{\mathbf{B}}x_t, \quad y_t = \mathbf{C}h_t, \quad (6)$$

where x_t , h_t , and y_t are the system’s discrete inputs, states, and outputs. $\bar{\mathbf{A}}$, $\bar{\mathbf{B}}$ are all discrete parameters of the system.

Inspired by Equ. 5 and 6, Mamba (Gu and Dao 2023), a new SSM-based model that introduces time-varying system parameters, has been introduced. Specifically, Δ , $\bar{\mathbf{A}}$, and $\bar{\mathbf{B}}$ are all functions of x_t . Mamba demonstrates long sequence modeling capabilities comparable to Transformer and achieves linear computational complexity during inference following Equ. 6. However, Equ. 6 is difficult to compute in parallel. It can be expanded and implemented using global convolution to enhance the efficiency of training Mamba on GPUs:

$$\bar{\mathbf{K}} = (\mathbf{C}\bar{\mathbf{B}}, \mathbf{C}\bar{\mathbf{A}}\bar{\mathbf{B}}, \dots, \mathbf{C}\bar{\mathbf{A}}^{M-1}\bar{\mathbf{B}}), \quad y = x * \bar{\mathbf{K}}, \quad (7)$$

where M is the length of the input sequence x , and $\bar{\mathbf{K}}$ is the kernel of the global convolution.

A Naive Mamba-based Point Cloud Network

Architecture. Using Mamba (Gu and Dao 2023) to replace the MLP operator in PointMLP (Ma et al. 2022) to achieve global modeling capabilities is a natural idea. However, this still requires addressing two challenges: 1) Point clouds are 3-D data, so how can we transform them into 1-D sequences? 2) Mamba is designed for causal modeling, so how can Mamba handle non-causal point cloud data? To address these challenges, we first adopt the z-order (Morton 1966) serialization method (Wang 2023) to flatten 3-D point cloud data into 1-D sequences, allowing point cloud data to be processed by Mamba. Secondly, inspired by (Zhu et al. 2024) and (Liu et al. 2024), we use bidirectional Mamba to allow each point to obtain features from any other point. At this point, we have implemented a naive Mamba-based network for the point cloud:

$$f_i^{l+1} = Mamba(x^l, reverse(x^l)), \quad (8)$$

$$x^l = S_z(A(GAM(\{f_{i,j}^l | j = 1, \dots, k\}))), \quad (9)$$

where S_z refers to serialization according to the z-order.

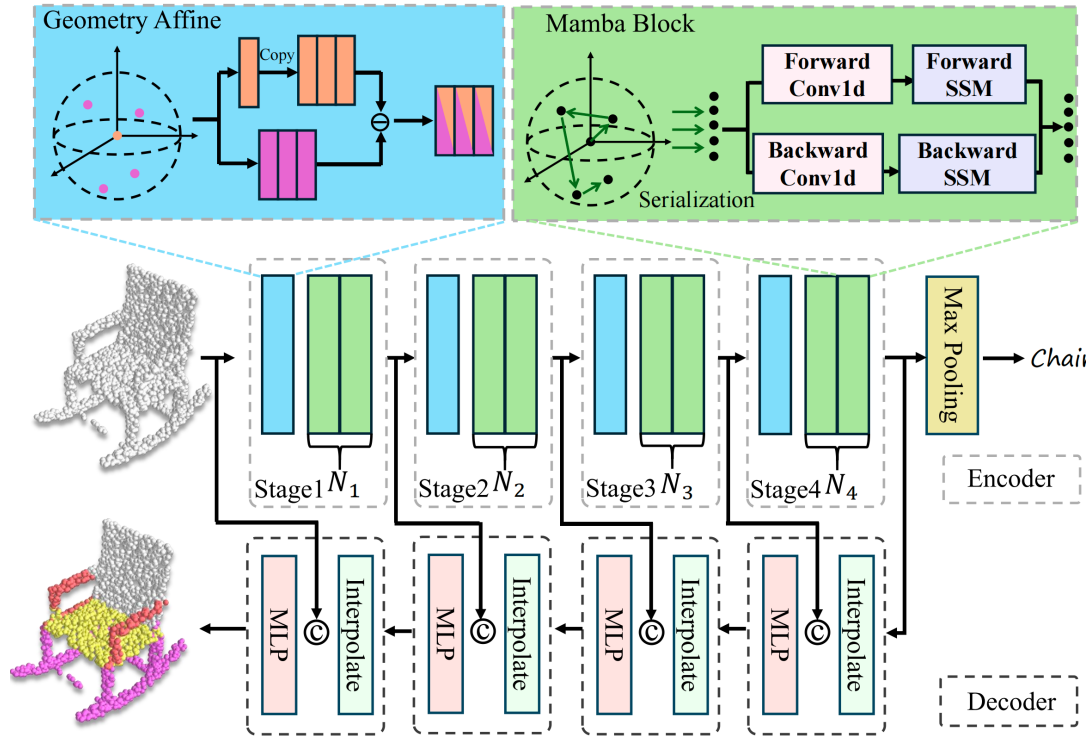


Figure 2: **The architecture of our proposed Point Cloud Mamba.** PCM encoder consists of four stages, each comprising a geometric affine module and several mamba layers. Point downsampling is performed between stages. The decoder only consists of point interpolation, feature concatenation, and MLP.

Shortcomings. The naive Mamba-based network still has some shortcomings, only achieving 84.1 OA on ScanObjectNN, significantly underperforming modern point-based methods like PointNeXt. There are some reasons: 1) Using a single serialization method to convert 3-D point cloud data into 1-D point sequences significantly loses the spatial relational information between points. Specifically, even if a point is adjacent to many others in space, it can only be adjacent to adjacent points in the point sequence. Therefore, employing multiple serialization methods in combination to alleviate this loss of spatial relational information is necessary. 2) The architecture design of the naive Mamba-based network follows the point-based method PointMLP, which may not be optimal or even reasonable for Mamba-based architecture. Exploring what benefits Mamba in modeling point clouds is important and necessary.

Point Cloud Mamba

In the last section, we obtained a naive mamba-based point cloud network; however, there is still plenty of room for optimization. Next, we will elaborate on improving this naive architecture to Point Cloud Mamba (PCM) and achieving performance beyond PointNeXt (Qian et al. 2022) and PTv3 (Wu et al. 2024).

Firstly, PCM provides various point cloud sequences through the consistent traverse serialization strategy and its variants, thereby preserving the relational information between points to the fullest extent by traversing multiple point sequences. Secondly, PCM introduces order prompts to assist

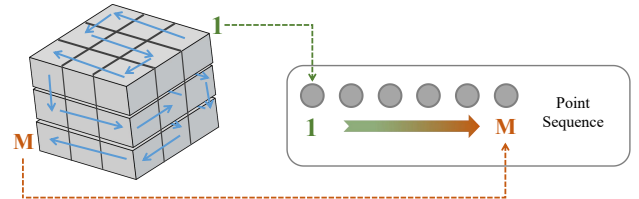


Figure 3: **The consistent traverse serialization strategy.** The 3-D point cloud data is voxelized and then serialized into a 1-D point sequence according to a predefined order. M represents the total number of points in the point cloud. With the permutation of x , y , and z coordinates, consistent traverse serialization has six variants.

Mamba layers in better handling point sequences generated by different serialization methods and strengthens the spatial position information of points through positional encoding. Finally, we have designed a more reasonable overall architecture.

Serialization strategy. How to convert 3-D point cloud data into a 1-D sequence that Mamba can handle is crucial. We find that Mamba can better process point cloud sequences arranged according to specific rules than disordered point cloud sequences. For example, randomly flattening point cloud data into a 1-D point sequence and feeding it into Mamba for modeling will significantly underperform compared to using an ordered point sequence (84.1 vs. 86.7). Based on this, we

propose the Consistent Traverse Serialization (CTS) strategy, which ensures that adjacent points in the sequence are also adjacent in spatial position. Figure 3 illustrates how CTS converts 3-D point clouds into a 1-D sequence.

Firstly, the point cloud is grid sampled to transform continuous spatial coordinates into discrete grid coordinates:

$$\{c_1^g, c_2^g, c_3^g\} = \text{int}(\{c_1^s, c_2^s, c_3^s\} \times N), \quad (10)$$

where $\{c_1^s, c_2^s, c_3^s\}$ and $\{c_1^g, c_2^g, c_3^g\}$ are the spatial and grid coordinates of the points, and N is the grid number. We design an encoding function that, given the coordinates of two dimensions, maps the coordinates to a code. Sorting the sequence according to the code ensures that adjacent points are contiguous in space.

$$\text{Code_func}(n_1, n_2) = \begin{cases} n_2 \times N + n_1, n_2 \% 2 = 0 \\ (n_2 + 1) \times N - n_1, n_2 \% 2 \neq 0 \end{cases} \quad (11)$$

Then, we can compute a code for each point based on its grid coordinates:

$$\text{code} = \text{Code_func}(\text{Code_func}(c_1^g, c_2^g), c_3^g). \quad (12)$$

Sorting the point cloud according to the **code** allows it to be flattened into a 1-D sequence. This simple serialization strategy performs comparably to carefully designed z-order (Morton 1966) and Hilbert-order (Hilbert and Hilbert 1935) serialization strategies (Wang 2023; Wu et al. 2024). Additionally, by exchanging the order of the x, y, and z axes, six different serialization methods can be derived, which we call "xyz", "xzy", "yxz", "yzx", "zxy", and "zyx". These different serialization methods can be viewed as various point cloud observations from different spatial perspectives.

Additionally, combining multiple serialization strategies can effectively assist Mamba in better modeling point cloud features, as shown in Table ???. Specifically, we adopt different serialization strategies for the inputs of different Mamba layers, allowing Mamba to perceive the point cloud more comprehensively, thus significantly surpassing a single serialization strategy alone.

Order prompt. When multiple serialization strategies are combined, assigning an identifier to each serialization is necessary. The identifier can help the Mamba layers recognize the arrangement of point cloud sequences and better capture point cloud features. We propose a simple but efficient order prompt mechanism to achieve this goal, which resembles the system messages in large language models.

As shown in Figure 4, we assigned N_p learnable embeddings as order prompts for each serialization order. Before being processed by the Mamba layer, the point cloud sequence has the corresponding order prompts added to both the beginning and end of the sequence. Considering that the input feature dimensions of the Mamba layers at different stages may vary, we also allocate a shared Linear layer for all Mamba layers within each stage to map the order prompts to the required channel size.

Positional embedding. In sequence modeling, positional encoding is crucial and widely applied in text and image patch sequence modeling. For example, RoPE (Su et al. 2024) and learnable positional embedding are widely used in text

sequence modeling and image patch sequence modeling, respectively. However, we find that these positional encoding methods are unsuitable for point clouds due to their sparsity and irregular shapes. The spatial distances between any two adjacent points in a point cloud sequence vary significantly, making it difficult to use the sequence positional information to represent the spatial gaps between adjacent points accurately. We employ a shared positional mapping function to address this challenge. It maps the point coordinates to positional embeddings with the same channel size as the features.

$$\text{Emd}_{pos} = \text{Linear}(\{c_1^s, c_2^s, c_3^s\}). \quad (13)$$

Emd_{pos} refers to the positional embedding projected from the spatial coordinates of the point. This simple positional mapping function accurately encodes the spatial information of points into features, and the positional embeddings corresponding to adjacent points in the sequence are similarly similar. Since the channel size of features varies across different stages of the network, we need to learn a private positional embedding function for each stage, and different Mamba layers within the same stage share the same positional encoding function.

Architecture settings. As shown in Figure 2, our network architecture consists of four stages, each incorporating a geometric affine module and several Mamba layers. We employ a simple decoder without Mamba layers for point cloud segmentation. The decoder solely conducts point cloud interpolation, concatenation with multi-stage encoder features, and channel transformation through MLP. The number of Mamba layers, serialization strategies, channel sizes, and order prompt counts in each stage of PCM-Tiny are illustrated in Table 1. Furthermore, we obtain PCM by increasing the number of Mamba layers and channel sizes and adopting new serialization methods accordingly.

Experiments

We conduct experiments on four datasets: ScanObjectNN (Uy et al. 2019) and ModelNet40 (Wu et al. 2015) classification datasets, ShapeNetPart (Chang et al. 2015) part segmentation dataset, and S3DIS (Armeni et al. 2016) semantic segmentation dataset. For the detailed experiment settings, please refer to the supplementary materials.

Main Results

3D object classification in ScanObjectNN dataset. ScanObjectNet (Uy et al. 2019) is a challenging point cloud classification dataset containing 15,000 real scanned objects categorized into 13 classes. It is known for its noise and occlusion challenges. Following PointMLP (Ma et al. 2022) and PointNeXt (Qian et al. 2022), we conducted experiments on PB_T50_RS, the most challenging and commonly used ScanObjectNN variant. As shown in Table 2, PCM achieved an OA of 88.1 and a mAcc of 86.6 on ScanObjectNN, surpassing the SOTA method PointNeXt and PTv3 by 0.4 and 1.7 in OA, respectively. Compared to PointMLP, Mamba layers demonstrated significantly stronger modeling capabilities than MLP, with PointNeXt surpassing PointMLP by 2.7 in OA and 2.7 in mAcc. By reducing the number of Mamba

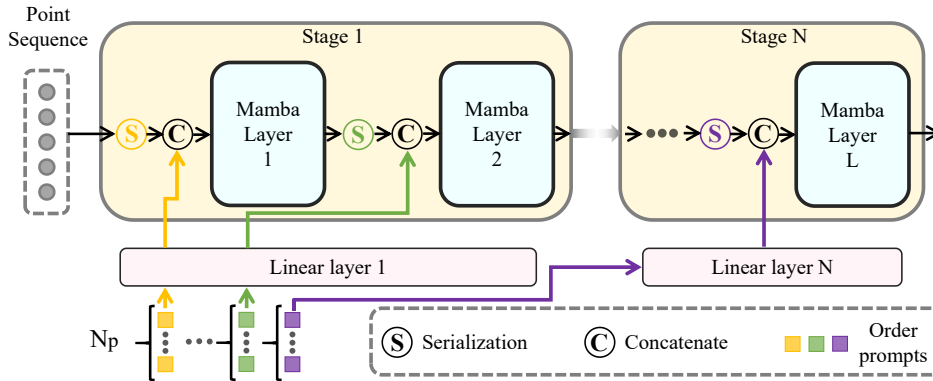


Figure 4: **The order prompts.** Different colors represent different serialization orders. N_p order prompts are mapped to the same channel size as the features and then concatenated to the beginning and end of the input point sequence.

Architecture	PCM-Tiny	PCM
Mamba layers	{1, 1, 2, 2}	{1, 2, 2, 4}
Serialization	{[xyz]-[xzy]-[yxz, yzx]-[zxy-zyx]}	{[xyz]-[xzy, yxz]-[yzx, zxy]-[zyx, H, z, z-trans]}
Channels	{192, 192, 384, 384}	{384, 384, 768, 768}
Order Prompts	6	6

Table 1: **Architecture settings.** The parameters of the four stages are enclosed in curly braces { }, while the parameters corresponding to each Mamba layer within a stage are enclosed in square brackets [].

layers and channel size, PCM-Tiny achieved an OA of 86.9 and a mAcc of 85.0 with only 20% of the parameters of PCM. It is worth noting that PCM-Tiny, with only 52% of the parameters of PointMLP (6.9 M vs. 13.2 M), still outperformed PointMLP by 1.5 in OA and 1.1 in mAcc.

The superior performance of PCM compared to PointMLP demonstrates the importance of Mamba’s global modeling capability for point cloud analysis.

3D object classification in ModelNet40 dataset. ModelNet40 (Wu et al. 2015) is a widely used synthetic 3D object classification dataset consisting of 40 categories, each with 100 unique CAD models. As shown in Table 2, PCM achieved an OA of 93.4 and a mAcc of 90.7, reaching a performance comparable to PointNeXt (Qian et al. 2022). PCM-Tiny achieved an OA of 93.1 and a mAcc of 90.6 with approximately 20% of the parameters of PCM. However, due to the smaller scale and less challenging nature of ModelNet40, performance on this dataset is difficult to differentiate between different methods’ modeling capabilities significantly, with most methods’ OA concentrated between 93 and 94. We have reproduced the experiments of PointMLP (Ma et al. 2022) and obtained an OA of 93.6, indicating that the high accuracy of 94.1 requires multiple repetitions and selection of the best.

3D semantic segmentation in S3DIS dataset. S3DIS (Armeni et al. 2016) is a large-scale indoor point cloud benchmark containing 6 large indoor areas, 271 rooms, and 13 semantic categories. PCM achieved 74.1 mIoU and 92.9 OA, surpassing PointNext (Qian et al. 2022) by 3.0 mIoU and 1.9 OA and exceeding PTv2 (Wu et al. 2022) by 1.4 mIoU and 1.3 OA. Moreover, PCM attained performance comparable to the current SOTA point-based method DeLA (Chen et al.

2023a) and transformer-based method PTv3 (Wu et al. 2024).

For better extraction of local point features, we cascade 4 DeLA blocks before PCM as an additional local feature extractor; please refer to the supplementary for details. When combining with the more powerful local feature extractor (Chen et al. 2023a), PCM-Tiny achieved 95.1 OA, 82.8 mAcc, and 79.6 mIoU, significantly surpassing the previous SOTA models DeLA (Chen et al. 2023a) and PTv3 (Wu et al. 2024) by 5.5 mIoU and 4.9 mIoU, respectively.

We also evaluate the performance of PTv3 with the same additional DeLA local feature extractor. However, the additional local feature extractor does not bring performance improvements to PTv3. This might be due to PTv3 performing attention within a window size of 1024, a limitation imposed by the quadratic computational complexity of transformers, which results in its limited global modeling capability.

Ablation Analysis and Visualization

Effect of serialization strategies. The key to applying Mamba for point cloud modeling is transforming point clouds into point sequences. As shown in Table 4, we conduct the ablation experiment with different serialization strategies. Similar performance is achieved when all Mamba layers’ inputs are serialized using a single order, whether it’s z-order, Hilbert-order, or our proposed xyz-order. However, significant performance gains are observed when more serialization strategies are employed. When the three variants of our proposed consistent traverse serialization, namely "xyz", "yxz", and "zxy", are used together, PCM demonstrates a performance improvement of 0.17 OA and 0.11 mAcc compared to using only the "xyz" variant. When all six variants of con-

Method	ScanObjectNN (PB_T50_RS)		ModelNet40		Params.
	OA (%)	mAcc (%)	OA (%)	mAcc (%)	M
PointNet (Qi et al. 2017a)	68.2	63.4	89.2	86.2	3.5
PointCNN (Li et al. 2018)	78.5	75.1	92.2	88.1	0.6
KPConv (Thomas et al. 2019)	-	-	92.9	-	14.3
ASSANet-L (Qian et al. 2021)	-	-	92.9	-	118.4
CurveNet (Xiang et al. 2021)	-	-	93.8	-	2.0
PointMLP (Ma et al. 2022)	85.4±1.3	83.9±1.5	94.1	91.3	13.2
PointNet++ (Qi et al. 2017b)	77.9	75.4	91.9	-	1.5
PointNeXt (Qian et al. 2022)	87.7±0.4	85.8±0.6	93.2±0.1	90.8±0.2	1.4
Transformer-based					
PCT (Guo et al. 2021)	-	-	93.2	-	2.9
Point-BERT (Yu et al. 2022)	83.1	-	93.2	-	22.1
Point-MAE (Pang et al. 2022)	85.2	-	93.8	-	22.1
PTv3 (Wu et al. 2024)	86.4	83.9	-	-	-
Mamba-based					
PointMamba (Liang et al. 2024)	84.9	-	-	-	12.3
PCM-Tiny (ours)	86.9±0.4	85.0±0.3	93.1±0.1	90.6±0.3	6.9
PCM (ours)	88.1±0.3	86.6±0.2	93.4±0.2	90.7±0.6	34.2

Table 2: **3D object classification in ScanObjectNN and ModelNet40.** Averaged results in three random runs using 1024 points as input without voting are reported.

Method	OA	mAcc	mIOU
PointNet (Qi et al. 2017a)	-	49.0	41.1
PointCNN (Li et al. 2018)	85.9	63.9	57.3
PointNeXt (Qian et al. 2022)	91.0	77.2	71.1
Strat. Trans. (Lai et al. 2022)	91.5	78.1	72.0
SPT (Robert, Raguét, and Landrieu 2023)	-	-	68.9
SpoTr (Park et al. 2023)	90.7	76.4	70.8
PTv1 (Zhao et al. 2021)	90.8	76.5	70.4
PTv2 (Wu et al. 2022)	91.6	78.0	72.7
DeLA (Chen et al. 2023a)	92.2	80.1	74.1
DeLA+X-3D (Sun et al. 2024)	92.2	80.1	74.3
KPConvX-L (Thomas et al. 2024)	91.7	78.7	73.5
PTv3 (Wu et al. 2024)	-	-	74.7
PTv3† (Wu et al. 2024)	91.4	78.4	72.3
PCM-Tiny (ours)	92.9	81.6	74.1
PCM-Tiny† (ours)	95.1	82.8	79.6

Table 3: **3D semantic segmentation in S3DIS.** † indicates using DeLA (Chen et al. 2023a) blocks as the additional local feature extractor.

sistent traverse serialization are combined, PCM shows a performance improvement of 0.39 OA and 0.51 mAcc. When all six variants of consistent traverse serialization, as well as "Hilbert," "z," and "z-trans" serialization strategies, are combined, PCM achieves a performance improvement of 0.49 OA and 0.54 mAcc. Different serialization strategies allow different Mamba layers to observe point clouds from different perspectives, resulting in more robust modeling of point cloud features.

Strategy	OA	mAcc
{"Z"} × 9	86.78	84.67
{"H"} × 9	86.78	84.68
{"xyz"} × 9	86.71	85.00
{"xyz", "yzx", "zxy"} × 3	86.88	85.11
{"xyz", "xzy", "yxz", "yzx", "zxy", "zyx", "xyz", "yzx", "zxy"}	87.10	85.51
{"xyz", "xzy", "yxz", "yzx", "zxy", "zyx", "H", "Z", "Z-trans"}	87.20	85.54

Table 4: **Ablation studies on serialization strategies.** Each mamba layer is assigned a serialization order and listed inside {}. "xyz", "xzy", "yxz", "yzx", "zxy", and "zyx" represent different variants of our proposed Consistent Traverse Serialization strategy.

Conclusion

This paper introduces a Mamba-based point cloud network named Point Cloud Mamba, which, for the first time, outperforms the SOTA point-based method PointNeXt and transformer-based method PTv3. Point Cloud Mamba incorporates several novel techniques to help Mamba better model point cloud data. Firstly, we propose Consistent Traverse Serialization to convert 3D point cloud data into 1D point sequences that Mamba can handle, ensuring that neighboring points in the sequence are also spatially adjacent. Secondly, we aid Mamba in handling point sequences serialized in different orders by introducing order prompts containing sequence arrangement rules. Finally, we propose a simple yet effective positional encoding based on spatial coordinate mapping. Our Point Cloud Mamba achieves SOTA performance on the ScanObjectNN, ModelNet40, ShapeNetPart, and S3DIS datasets. Adding a stronger local feature extractor, our method also outperforms previous STOA methods by a large margin on the S3DIS dataset.

References

- Armeni, I.; Sener, O.; Zamir, A. R.; Jiang, H.; Brilakis, I.; Fischer, M.; and Savarese, S. 2016. 3d semantic parsing of large-scale indoor spaces. In *CVPR*, 1534–1543.
- Behrouz, A.; and Hashemi, F. 2024. Graph Mamba: Towards Learning on Graphs with State Space Models.
- Boulch, A. 2020. ConvPoint: Continuous convolutions for point cloud processing. *Computers & Graphics*, 88: 24–34.
- Carion, N.; Massa, F.; Synnaeve, G.; Usunier, N.; Kirillov, A.; and Zagoruyko, S. 2020. End-to-end object detection with transformers. In *ECCV*.
- Chang, A. X.; Funkhouser, T.; Guibas, L.; Hanrahan, P.; Huang, Q.; Li, Z.; Savarese, S.; Savva, M.; Song, S.; Su, H.; et al. 2015. Shapenet: An information-rich 3d model repository. *arXiv:1512.03012*.
- Chen, B.; Xia, Y.; Zang, Y.; Wang, C.; and Li, J. 2023a. Decoupled local aggregation for point cloud learning. *arXiv preprint arXiv:2308.16532*.
- Chen, G.; Wang, M.; Yang, Y.; Yu, K.; Yuan, L.; and Yue, Y. 2023b. PointGPT: Auto-regressively Generative Pre-training from Point Clouds. *arXiv:2305.11487*.
- Choy, C.; Gwak, J.; and Savarese, S. 2019. 4d spatio-temporal convnets: Minkowski convolutional neural networks. In *CVPR*, 3075–3084.
- Dai, A.; and Nießner, M. 2018. 3dmv: Joint 3d-multi-view prediction for 3d semantic scene segmentation. In *ECCV*, 452–468.
- Dosovitskiy, A.; Beyer, L.; Kolesnikov, A.; Weissenborn, D.; Zhai, X.; Unterthiner, T.; Dehghani, M.; Minderer, M.; Heigold, G.; Gelly, S.; et al. 2021. An image is worth 16x16 words: Transformers for image recognition at scale. In *ICLR*.
- Fang, Z.; Li, X.; Li, X.; Buhmann, J. M.; Loy, C. C.; and Liu, M. 2023. Explore In-Context Learning for 3D Point Cloud Understanding. *NeurIPS*.
- Graham, B.; Engelcke, M.; and Van Der Maaten, L. 2018. 3d semantic segmentation with submanifold sparse convolutional networks. In *CVPR*, 9224–9232.
- Gu, A.; and Dao, T. 2023. Mamba: Linear-time sequence modeling with selective state spaces. *arXiv preprint arXiv:2312.00752*.
- Gu, A.; Goel, K.; and Ré, C. 2022. Efficiently Modeling Long Sequences with Structured State Spaces. In *ICLR*.
- Guo, M.-H.; Cai, J.-X.; Liu, Z.-N.; Mu, T.-J.; Martin, R. R.; and Hu, S.-M. 2021. Pct: Point cloud transformer. In *CVM*.
- He, X.; Cao, K.; Yan, K.; Li, R.; Xie, C.; Zhang, J.; and Zhou, M. 2024. Pan-Mamba: Effective pan-sharpening with State Space Model. *arXiv preprint arXiv:2402.12192*.
- Hilbert, D.; and Hilbert, D. 1935. Über die stetige Abbildung einer Linie auf ein Flächenstück. *Dritter Band: Analysis-Grundlagen der Mathematik- Physik Verschiedenes: Nebst Einer Lebensgeschichte*, 1–2.
- Hou, Y.; Zhu, X.; Ma, Y.; Loy, C. C.; and Li, Y. 2022. Point-to-voxel knowledge distillation for lidar semantic segmentation. In *CVPR*, 8479–8488.
- Jiang, J.; Lu, X.; Zhao, L.; Dazeley, R.; and Wang, M. 2022. Masked autoencoders in 3D point cloud representation learning. *arXiv:2207.01545*.
- Jiang, L.; Yang, Z.; Shi, S.; Golyanik, V.; Dai, D.; and Schiele, B. 2023. Self-supervised Pre-training with Masked Shape Prediction for 3D Scene Understanding. In *CVPR*, 1168–1178.
- Komarichev, A.; Zhong, Z.; and Hua, J. 2019. A-cnn: Annularly convolutional neural networks on point clouds. In *CVPR*.
- Lahoud, J.; Cao, J.; Khan, F. S.; Cholakkal, H.; Anwer, R. M.; Khan, S.; and Yang, M.-H. 2022. 3D Vision with Transformers: A Survey. *arXiv:2208.04309*.
- Lai, X.; Liu, J.; Jiang, L.; Wang, L.; Zhao, H.; Liu, S.; Qi, X.; and Jia, J. 2022. Stratified Transformer for 3D Point Cloud Segmentation. In *CVPR*.
- Lang, A. H.; Vora, S.; Caesar, H.; Zhou, L.; Yang, J.; and Beijbom, O. 2019. Pointpillars: Fast encoders for object detection from point clouds. In *CVPR*, 12697–12705.
- Li, S.; Singh, H.; and Grover, A. 2024. Mamba-ND: Selective State Space Modeling for Multi-Dimensional Data. *arXiv preprint arXiv:2402.05892*.
- Li, X.; Ding, H.; Zhang, W.; Yuan, H.; Cheng, G.; Jiangmiao, P.; Chen, K.; Liu, Z.; and Loy, C. C. 2023. Transformer-Based Visual Segmentation: A Survey. *arXiv pre-print*.
- Li, Y.; Bu, R.; Sun, M.; Wu, W.; Di, X.; and Chen, B. 2018. Pointcnn: Convolution on x-transformed points. In *NeurIPS*.
- Liang, D.; Zhou, X.; Wang, X.; Zhu, X.; Xu, W.; Zou, Z.; Ye, X.; and Bai, X. 2024. PointMamba: A Simple State Space Model for Point Cloud Analysis. *arXiv preprint arXiv:2402.10739*.
- Liu, H.; Cai, M.; and Lee, Y. J. 2022. Masked discrimination for self-supervised learning on point clouds. In *ECCV*.
- Liu, Y.; Fan, B.; Xiang, S.; and Pan, C. 2019. Relation-shape convolutional neural network for point cloud analysis. In *CVPR*.
- Liu, Y.; Tian, Y.; Zhao, Y.; Yu, H.; Xie, L.; Wang, Y.; Ye, Q.; and Liu, Y. 2024. Vmamba: Visual state space model. *arXiv preprint arXiv:2401.10166*.
- Liu, Z.; Yang, X.; Tang, H.; Yang, S.; and Han, S. 2023. FlatFormer: Flattened Window Attention for Efficient Point Cloud Transformer. In *CVPR*, 1200–1211.
- Ma, J.; Li, F.; and Wang, B. 2024. U-Mamba: Enhancing Long-range Dependency for Biomedical Image Segmentation. *arXiv preprint arXiv:2401.04722*.
- Ma, X.; Qin, C.; You, H.; Ran, H.; and Fu, Y. 2022. Rethinking network design and local geometry in point cloud: A simple residual MLP framework. In *ICLR*.
- Morton, G. M. 1966. A computer oriented geodetic data base and a new technique in file sequencing.
- Pang, Y.; Wang, W.; Tay, F. E.; Liu, W.; Tian, Y.; and Yuan, L. 2022. Masked autoencoders for point cloud self-supervised learning. In *ECCV*.
- Park, J.; Lee, S.; Kim, S.; Xiong, Y.; and Kim, H. J. 2023. Self-positioning point-based transformer for point cloud understanding. In *ICCV*, 21814–21823.

- Qi, C. R.; Su, H.; Mo, K.; and Guibas, L. J. 2017a. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *CVPR*.
- Qi, C. R.; Yi, L.; Su, H.; and Guibas, L. J. 2017b. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. In *NeurIPS*.
- Qian, G.; Hammoud, H.; Li, G.; Thabet, A.; and Ghanem, B. 2021. ASSANet: An Anisotropic Separable Set Abstraction for Efficient Point Cloud Representation Learning. In *NeurIPS*.
- Qian, G.; Li, Y.; Peng, H.; Mai, J.; Hammoud, H.; Elhoseiny, M.; and Ghanem, B. 2022. Pointnext: Revisiting pointnet++ with improved training and scaling strategies. *NeurIPS*.
- Ran, H.; Liu, J.; and Wang, C. 2022. Surface representation for point clouds. In *CVPR*.
- Robert, D.; Raguette, H.; and Landrieu, L. 2023. Efficient 3D semantic segmentation with superpoint transformer. In *ICCV*, 17195–17204.
- Ruan, J.; and Xiang, S. 2024. Vm-unet: Vision mamba unet for medical image segmentation. *arXiv preprint arXiv:2402.02491*.
- Sanghi, A. 2020. Info3d: Representation learning on 3d objects using mutual information maximization and contrastive learning. In *ECCV*, 626–642. Springer.
- Sauder, J.; and Sievers, B. 2019. Self-supervised deep learning on point clouds by reconstructing space. *NIPS*, 32.
- Schult, J.; Engelmann, F.; Hermans, A.; Litany, O.; Tang, S.; and Leibe, B. 2023. Mask3D: Mask Transformer for 3D Semantic Instance Segmentation.
- Shen, Y.; Feng, C.; Yang, Y.; and Tian, D. 2018. Mining point cloud local structures by kernel correlation and graph pooling. In *CVPR*.
- Shi, W.; and Rajkumar, R. 2020. Point-gnn: Graph neural network for 3d object detection in a point cloud. In *CVPR*, 1711–1719.
- Su, J.; Ahmed, M.; Lu, Y.; Pan, S.; Bo, W.; and Liu, Y. 2024. Roformer: Enhanced transformer with rotary position embedding. *Neurocomputing*, 568: 127063.
- Sun, J.; Qing, C.; Tan, J.; and Xu, X. 2023. Superpoint Transformer for 3D Scene Instance Segmentation. *AAAI*.
- Sun, S.; Rao, Y.; Lu, J.; and Yan, H. 2024. X-3D: Explicit 3D Structure Modeling for Point Cloud Recognition. In *CVPR*, 5074–5083.
- Thomas, H.; Qi, C. R.; Deschaud, J.-E.; Marcotegui, B.; Goulette, F.; and Guibas, L. J. 2019. Kpconv: Flexible and deformable convolution for point clouds. In *ICCV*.
- Thomas, H.; Tsai, Y.-H. H.; Barfoot, T. D.; and Zhang, J. 2024. KPConvX: Modernizing Kernel Point Convolution with Kernel Attention. In *CVPR*, 5525–5535.
- Uy, M. A.; Pham, Q.-H.; Hua, B.-S.; Nguyen, T.; and Yeung, S.-K. 2019. Revisiting point cloud classification: A new benchmark dataset and classification model on real-world data. In *ICCV*.
- Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, Ł.; and Polosukhin, I. 2017. Attention is all you need. In *NeurIPS*.
- Wang, L.; Huang, Y.; Hou, Y.; Zhang, S.; and Shan, J. 2019. Graph attention convolution for point cloud semantic segmentation. In *CVPR*, 10296–10305.
- Wang, P.-S. 2023. OctFormer: Octree-based Transformers for 3D Point Clouds. *SIGGRAPH*.
- Wang, X.; Fang, Z.; Li, X.; Li, X.; Chen, C.; and Liu, M. 2024. Skeleton-in-Context: Unified Skeleton Sequence Modeling with In-Context Learning. *CVPR*.
- Wang, Y.; and Solomon, J. M. 2019. Deep closest point: Learning representations for point cloud registration. In *ICCV*, 3523–3532.
- Wu, W.; Qi, Z.; and Fuxin, L. 2019. Pointconv: Deep convolutional networks on 3d point clouds. In *CVPR*.
- Wu, X.; Jiang, L.; Wang, P.-S.; Liu, Z.; Liu, X.; Qiao, Y.; Ouyang, W.; He, T.; and Zhao, H. 2024. Point Transformer V3: Simpler, Faster, Stronger. In *CVPR*.
- Wu, X.; Lao, Y.; Jiang, L.; Liu, X.; and Zhao, H. 2022. Point transformer V2: Grouped Vector Attention and Partition-based Pooling. In *NeurIPS*.
- Wu, X.; Tian, Z.; Wen, X.; Peng, B.; Liu, X.; Yu, K.; and Zhao, H. 2023a. Towards large-scale 3d representation learning with multi-dataset point prompt training. *arXiv preprint arXiv:2308.09718*.
- Wu, X.; Wen, X.; Liu, X.; and Zhao, H. 2023b. Masked scene contrast: A scalable framework for unsupervised 3d representation learning. In *CVPR*, 9415–9424.
- Wu, Z.; Song, S.; Khosla, A.; Yu, F.; Zhang, L.; Tang, X.; and Xiao, J. 2015. 3d shapenets: A deep representation for volumetric shapes. In *CVPR*.
- Xiang, T.; Zhang, C.; Song, Y.; Yu, J.; and Cai, W. 2021. Walk in the cloud: Learning curves for point clouds shape analysis. In *ICCV*.
- Xie, S.; Gu, J.; Guo, D.; Qi, C. R.; Guibas, L.; and Litany, O. 2020. Pointcontrast: Unsupervised pre-training for 3d point cloud understanding. In *ECCV*.
- Xing, Z.; Ye, T.; Yang, Y.; Liu, G.; and Zhu, L. 2024. Segmamba: Long-range sequential modeling mamba for 3d medical image segmentation. *arXiv preprint arXiv:2401.13560*.
- Xu, M.; Ding, R.; Zhao, H.; and Qi, X. 2021a. Paconv: Position adaptive convolution with dynamic kernel assembling on point clouds. In *CVPR*.
- Xu, M.; Zhang, J.; Zhou, Z.; Xu, M.; Qi, X.; and Qiao, Y. 2021b. Learning geometry-disentangled representation for complementary understanding of 3d object point cloud. In *AAAI*.
- Yan, X.; Gao, J.; Zheng, C.; Zheng, C.; Zhang, R.; Cui, S.; and Li, Z. 2022. 2dpass: 2d priors assisted semantic segmentation on lidar point clouds. In *ECCV*. Springer.
- Yang, Y.; Xing, Z.; and Zhu, L. 2024. Vivim: a video vision mamba for medical video object segmentation. *arXiv preprint arXiv:2401.14168*.
- Yang, Y.-Q.; Guo, Y.-X.; Xiong, J.-Y.; Liu, Y.; Pan, H.; Wang, P.-S.; Tong, X.; and Guo, B. 2023. Swin3D: A Pretrained Transformer Backbone for 3D Indoor Scene Understanding. *arXiv preprint arXiv:2304.06906*.

Yi, L.; Kim, V. G.; Ceylan, D.; Shen, I.-C.; Yan, M.; Su, H.; Lu, C.; Huang, Q.; Sheffer, A.; and Guibas, L. 2016. A scalable active framework for region annotation in 3d shape collections. In *TOG*.

Yu, X.; Tang, L.; Rao, Y.; Huang, T.; Zhou, J.; and Lu, J. 2022. Point-BERT: Pre-Training 3D Point Cloud Transformers with Masked Point Modeling. In *CVPR*.

Zhang, Z.; Girdhar, R.; Joulin, A.; and Misra, I. 2021. Self-supervised pretraining of 3d features on any point-cloud. In *ICCV*.

Zhao, H.; Jiang, L.; Jia, J.; Torr, P. H.; and Koltun, V. 2021. Point Transformer. In *ICCV*.

Zhu, H.; Yang, H.; Wu, X.; Huang, D.; Zhang, S.; He, X.; He, T.; Zhao, H.; Shen, C.; Qiao, Y.; et al. 2023. Ponderv2: Pave the way for 3d foundataion model with a universal pre-training paradigm. *arXiv preprint arXiv:2310.08586*.

Zhu, L.; Liao, B.; Zhang, Q.; Wang, X.; Liu, W.; and Wang, X. 2024. Vision mamba: Efficient visual representation learning with bidirectional state space model. *arXiv preprint arXiv:2401.09417*.

Zhu, X.; Zhou, H.; Wang, T.; Hong, F.; Ma, Y.; Li, W.; Li, H.; and Lin, D. 2021. Cylindrical and asymmetrical 3d convolution networks for lidar segmentation. In *CVPR*, 9939–9948.