PointMamba: A Simple State Space Model for Point Cloud Analysis

Dingkang Liang[1*], Xin Zhou[1*], Wei Xu[1], Xingkui Zhu[1], Zhikang Zou[2],

Xiaoqing Ye[2], Xiao Tan[2], Xiang Bai[1†]

[1]Huazhong University of Science & Technology, [2]Baidu Inc.
{dkliang, xzhou03, xbai}@hust.edu.cn
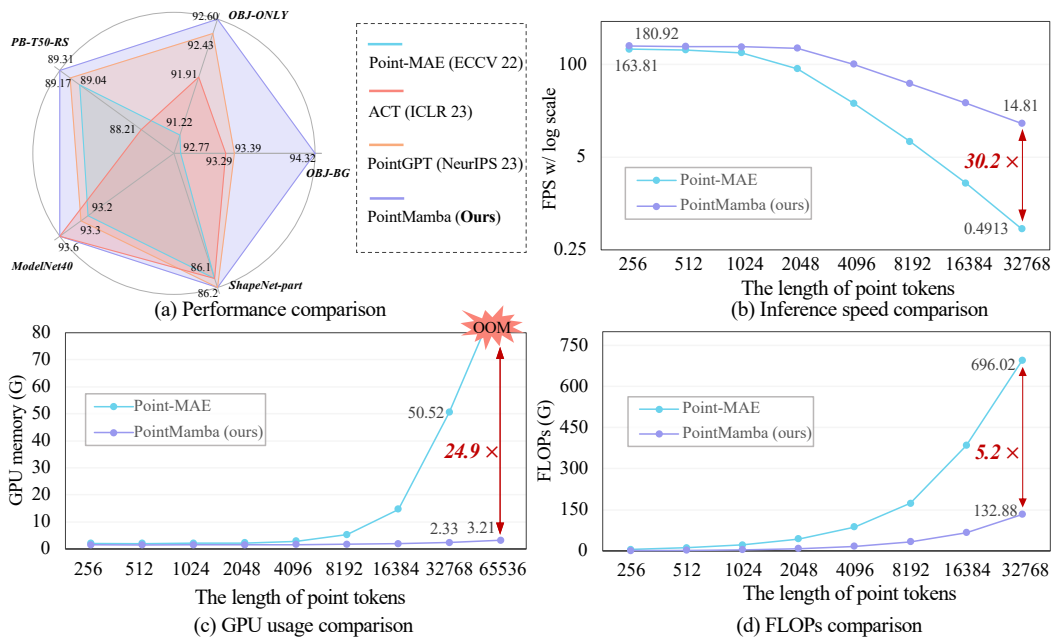
Figure 1: Comprehensive comparisons between our PointMamba and its Transformer-based counterparts [33, 6, 11]. (a) Without bells and whistles, our PointMamba achieve better performance than the representative Transformer-based methods on the various point cloud analysis datasets. (b)-(d) The Transformer presents quadratic complexity, while our PointMamba has linear complexity. For example, with the length of point tokens increasing, we significantly reduce GPU memory usage and FLOPs and have a faster inference speed compared to the most convincing Transformer-based method, i.e., PointMAE [33].

Abstract

Transformers have become one of the foundational architectures in point cloud analysis tasks due to their excellent global modeling ability. However, the attention mechanism has quadratic complexity, making the design of a linear complexity method with global modeling appealing. In this paper, we propose **PointMamba** 🐼, transferring the success of Mamba, a recent representative state space model (SSM), from NLP to point cloud analysis tasks. Unlike traditional Transformers, PointMamba employs a linear complexity algorithm, presenting global modeling capacity while significantly reducing computational costs. Specifically, our

---

*Equal contribution. † Corresponding author.

38th Conference on Neural Information Processing Systems (NeurIPS 2024).

method leverages space-filling curves for effective point tokenization and adopts an extremely simple, non-hierarchical Mamba encoder as the backbone. Comprehensive evaluations demonstrate that PointMamba achieves superior performance across multiple datasets while significantly reducing GPU memory usage and FLOPs. This work underscores the potential of SSMs in 3D vision-related tasks and presents a simple yet effective Mamba-based baseline for future research. The code is available at https://github.com/LMD0311/PointMamba.

1 Introduction

Point cloud analysis is one of the fundamental tasks in computer vision and has a wide range of real-world applications [47, 55, 8], including robotics, autonomous driving, and augmented reality. It is a challenging task due to the intrinsic irregularity and sparsity of point clouds. To address the issues, there has been rapid progress in deep learning-based methods [38, 33, 41, 44], consistently pushing the performance to the new record.

Recently, Transformers have achieved remarkable progress in point cloud analysis. The key to the Transformer is the attention mechanism, which can effectively capture the relationship of a set of points. By integrating self-supervised learning paradigms with fine-tuning for downstream tasks, these Transformer-based methods have achieved superior performance [60, 33, 42]. However, the complexity of attention mechanisms is quadratic, bringing significant computational cost, which is not friendly to low-resource devices. Thus, this naturally raises a question: *how to design a simple, elegant method that operates with linear complexity, thereby retaining the benefits of global modeling for point cloud analysis?*

We note the recent advance of the State Space Models (SSMs). As a pioneer, the Structured State Space Sequence Model [17] (S4) has emerged as a promising class of architectures for sequence modeling thanks to its strong representation ability and linear-time complexity (achieved by eliminating the need to store the complete context). Another pioneer, Mamba [14], adopts time-varying parameters to the SSM based on S4, proposing an efficient hardware-aware algorithm to enable highly efficient training and inference with dynamic modeling. Recently, a few concurrent methods [71, 35] successfully transfer the 1D-sequence Mamba from NLP to 2D vision tasks (e.g., image classification and segmentation), achieving similar or surpass the Transformer counterpart [12] while significantly reducing memory usage. However, regarding the more complex, unstructured data, e.g., 3D point cloud, the effectiveness of Mamba remains unclear. The lack of early exploration of Mamba's potential for point cloud-related tasks hinders further development of its capabilities across the diverse range of applications in this domain.

Inspired by this, this paper aims to unlock the potential of SSM in point cloud analysis tasks, discussing whether it can be a viable alternative to Transformers in this domain. Through a series of pilot experiments, we find that directly using the pioneering SSM, Mamba [14], can not achieve ideal performance. We argue that the main inherent limitation comes from the unidirectional modeling employed by the default Mamba, as the context is obtained by compressing the historical hidden state instead of through the interaction between each element. In contrast, the self-attention of the Transformer is invariant to the permutation of the input elements. Given the three-dimensional nature (e.g., unstructured and disordered) of point clouds, using a single scanning process often struggles to concurrently capture dependency information across various directions, which makes it difficult to construct global modeling for the RNN-like modes (e.g., Mamba).

Therefore, we introduce a simple yet effective Point Cloud State Space Model (denoted as **PointMamba** 🐍) with global modeling and linear complexity. Specifically, to enable Mamba to capture the point cloud structure causally, we first use a point tokenizer to generate two types of point tokens via a point scanning strategy, employing two space-filling curves to scan key points from different directions. As a result, the unstructured 3D point clouds can be transformed into a regular sequence. The first type of token has local modeling capabilities through sequential encoding, with the latest token holding global sequence information. Consequently, the second type of token can achieve global modeling by containing global information that comes from the first type. Besides, we propose an extremely simple order indicator to maintain the distinct spatial characteristics of different scanning when training, preserving the integrity of the spatial information.

2

To make the model as simple as possible, PointMamba only employs plain and non-hierarchical Mamba as the backbone to extract features for given serialized point tokens without bells and whistles. We demonstrate that PointMamba is very flexible in the pre-training paradigm, where we customize an MAE-like pertaining strategy to provide a good prior, which chooses a random serialization strategy from a pre-defined serialization bank to perform mask modeling. It facilitates the model to exact the general local relationships from different scanning perspectives, better matching the requirement of indirection modeling of mamba.

Despite no elaborate or complex designs in the model, our PointMamba achieves superior performance on various point cloud analysis datasets (Fig. 1(a)). Besides the superior performance, thanks to the linear complexity of Mamba, we show the surprisingly low computational cost[2], as shown in Fig. 1(b)-(c). These notable results underscore the potential of SSM in 3D vision-related tasks.

In conclusion, the contributions of this paper are twofold. **1)** We introduce the first state space model for point cloud analysis, named **PointMamba** , which features global modeling with linear complexity. Despite the absence of elaborate or complex structural designs, PointMamba demonstrates its potential as an optional model for 3D vision applications. **2)** Our PointMamba exhibits impressive capabilities, including structural simplicity (e.g., vanilla Mamba), low computational cost, and knowledge transferability (e.g., support for self-supervised learning).

2 Related work

2.1 Point Cloud Transformers

Vision Transformer (ViT) [12] has become one of the mainstream architectures in point cloud analysis tasks due to its excellent global modeling ability. Specifically, Point-BERT [60] and Point-MAE [38] introduce a standard Transformer architecture for self-supervised learning and is applicable to different downstream tasks. Several works further introduce GPT scheme [6], multi-scale [64, 61], and multi-modal [11, 42, 43] to guide 3D representation learning. On the other hand, some researchers [20, 68, 51] focus on modifying the Transformers for point clouds. The PCT [20] conducts global attention directly on the point cloud. Point Transformer [68] applies vector attention [67] to perform local attention between each point and its adjacent points. The later Point Transformer series [56, 55] further extends the performance and efficiency of the Transformer for different tasks. OctFormer [51] leverages sorted shuffled keys of octrees to partition point clouds and significantly improve efficiency and effectiveness.

Standard Transformers can be smoothly integrated into autoencoders using an encoder-decoder design [25], which makes this structure ideal for pre-training and leads to significant performance improvements in downstream point cloud analysis tasks [60, 38, 6, 62, 63]. However, the attention mechanism has a time complexity of $O(n^2d)$, where n represents the length of the input token sequence and d represents the dimension of the Transformer. This implies that as the input sequence grows, the operational efficiency of the Transformer is significantly constrained.

In this work, we focus on designing a simple State Space Model (SSM) for point cloud analysis without attention while maintaining the global modeling advantages of the Transformer.

2.2 State Space Models

Linear state space equations [15, 18], combined with deep learning, offer a compelling approach for modeling sequential data, presenting an alternative to CNNs or Transformers. The Structured State Space Sequence Model [17] (S4) leverages a linear state space for contextualization and shows strong performance on various sequence modeling tasks, especially with lengthy sequences. To alleviate computational burden, HTTYH [19], DSS [22], and S4D [16] propose employing a diagonal matrix within S4, maintaining performance without excessive computational costs. The S5 [48] proposes a parallel scan and the MIMO SSM, enabling the state space model to be efficiently utilized and widely implemented. Recently, Mamba [14] introduced the selective SSM mechanism, a breakthrough achieving linear-time inference and effective training using a hardware-aware algorithm, garnering considerable attention. In the vision domain, Vision Mamba [35] compresses the visual

---

[2]Note that we removed the tokenizer of both Point-MAE and PointMamba , directly fed with a predefined sequence, to better illustrate the structural efficiency.

3

representation through bidirectional state space models. VMamba [71] introduces the Cross-Scan Module, enabling 1D selective scanning in 2D images with global receptive fields. Besides, the great potential of Mamba motivates a series of work across diverse domains, including graph [50, 3], medical segmentation [36, 34], video understanding [30, 7] and generative models [29, 31].

To the best of our knowledge, there are limited works that study SSMs for point cloud analysis. In this work, we delve into the potential of Mamba in point cloud analysis and propose PointMamba, which achieves superior performance and significantly reduces computational costs.

3 Preliminaries

State Space Model. Drawing inspiration from control theory, the State Space Model (SSM) represents a continuous system that maps a state x_t to y_t through an implicit latent state $h_t \in \mathbb{R}^N$. To integrate SSMs into deep models, S4 [17] defines the system with four parameters ($\boldsymbol{A}, \boldsymbol{B}, \boldsymbol{C}$, and sampling step size $\boldsymbol{\Delta}$). The sequence-to-sequence transformation is defined as:

$$h_t = \overline{\boldsymbol{A}} h_{t-1} + \overline{\boldsymbol{B}} x_t, \qquad y_t = \boldsymbol{C} h_t + \boldsymbol{D} x_t, \tag{1}$$

where $\boldsymbol{C} \in \mathbb{R}^{1 \times N}$ is a project parameter, and $\boldsymbol{D} \in \mathbb{R}^{1 \times N}$ represents a residual connection. The parameters $\overline{\boldsymbol{A}}, \overline{\boldsymbol{B}}$ are defined using the zero-order hold (ZOH) discretization rule:

$$\overline{\boldsymbol{A}} \in \mathbb{R}^{N \times N} = \exp(\boldsymbol{A}\boldsymbol{\Delta}), \qquad \overline{\boldsymbol{B}} \in \mathbb{R}^{N \times 1} = (\boldsymbol{A}\boldsymbol{\Delta})^{-1} \left(\exp(\boldsymbol{A}\boldsymbol{\Delta}) - \boldsymbol{I}\right) \cdot \boldsymbol{\Delta}\boldsymbol{B}. \tag{2}$$

However, the parameter ($\overline{\boldsymbol{A}}, \overline{\boldsymbol{B}}, \boldsymbol{C}, \boldsymbol{\Delta}$) are fixed across all time steps due to the Linear Time-Invariant (LTI) property of SSMs, which limits their capacity to handle varied input sequences.

Recently, Selective SSM (S6) considers parameters $\boldsymbol{B}, \boldsymbol{C}, \boldsymbol{\Delta}$ as functions of the input, effectively transforming the SSM into a time-variant model. Our PointMamba adopts a hardware-aware implementation [14] of S6, showing linear complexity and strong sequence modeling capability.

Space-filling curve. Space-filling curves are paths that traverse every point within a higher-dimensional discrete space while maintaining spatial proximity to a certain degree. Mathematically, they can be defined as a bijective function $\Phi : \mathbb{Z} \to \mathbb{Z}^3$ for point clouds. Our PointMamba focuses on the Hilbert space-filling curve [27] and its transposed variant (called Trans-Hilbert), both of which are recognized for effectively preserving locality, ensuring that data points close in $\mathbb{Z}$ space remain close after transformation to $\mathbb{Z}^3$. We note that some methods [55, 51] utilize space-filling curves to partition the point cloud for capturing spatial contexts, whereas our work mainly focuses on transferring the point clouds to serialization-based sequences and combine with Mamba to implement global modeling. The motivation and objective are different.

4 PointMamba

This paper aims to design a simple yet solid Mamba-based Point cloud analysis method. The pipeline of our method is shown in Fig. 2. Starting with an input point cloud, we first sample the key points via Farthest Point Sampling (FPS). Then, a simple space-scanning strategy is applied to reorganize these points, resulting in serialized key points. Under a KNN and lightweight PointNet [40], we obtain the serialized point tokens. Finally, the entire sequence is subsequently processed by a plain, non-hierarchical encoder structure composed of several stacked Mamba blocks. Besides, to provide a good prior for PointMamba, we propose a serialization-based mask modeling paradigm, which randomly chooses a space-filling curve for serialization and mask, as shown in Fig. 4.

4.1 The structure of PointMamba

In this section, we introduce the structure of our PointMamba. The goal of this paper is to provide a simple yet solid Mamba baseline for point cloud analysis tasks and explore the potential of plain and non-hierarchical Mamba. Thus, in the spirit of *Occam's razor*, we make the structure as simple as possible without any complex or elaborate design.

Point scanning strategy. Building on the pioneer works [60, 38], we first utilize the Farthest Point Sampling (FPS) to select the key points. Specifically, given an input point cloud $\boldsymbol{P} \in \mathbb{R}^{M \times 3}$, where M is the number of points, the FPS is applied to sample n key points from the original point cloud

4
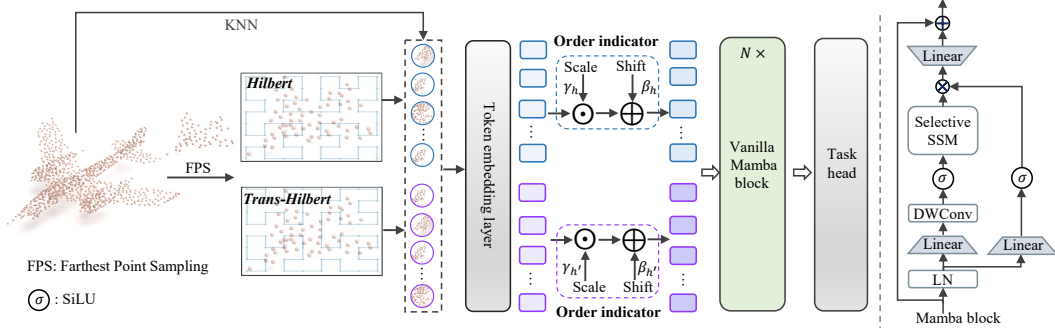
Figure 2: The pipeline of our PointMamba. It is simple and elegant, without bells and whistles. We first utilize Farthest Point Sampling (FPS) to select the key points. Then, we propose to utilize two types of space-filling curves, including Hilbert and Trans-Hilbert, to generate the serialized key points. Based on these, the KNN is used to form point patches, which will be fed to the token embedding layer to generate the serialized point tokens. To indicate the tokens generated from which space-filling curve, the order indicator is proposed. The encoder is extremely simple, consisting of $N\times$ plain and non-hierarchical Mamba blocks.

P, resulting in $p \in \mathbb{R}^{n\times 3}$. In general, the order of the sampling key points p is random, without specific order. This is not a significant problem for the previous Transformer-based methods, as the Transformer is order-invariant when processing sequence data: in the self-attention mechanism, each element at a given position can interact with all other elements in the sequence through attention weights. However, for the selective state space model, i.e., Mamba, we argue that it is hard to model the unstructured point clouds due to the unidirectional modeling. Thus, we propose to leverage the space-filling curves to transform the unstructured point clouds into a regular sequence. Specifically, we choose two representative space-filling curves to scan the key points: the Hilbert curve [27] and its transposed variant, denoted as Trans-Hilbert. Compared with the random sequence, space-filling curves like the Hilbert curve can preserve spatial locality well, i.e., along the scanned 1D serialized point sequence, adjacent key points often have geometrically close positions in 3D space. We argue that this property ensures that the spatial relationships between points are largely maintained, which is crucial for accurate feature representation and analysis in point cloud data. As a complementary, the Trans-Hilbert performs similarly but scans from different clues, which can provide diverse perspectives on spatial locality. By applying Hilbert and Trans-Hilbert to the key points, we obtain two different point serializations, p_h and $p_{h'}$, which will be used to construct point tokens.

Point tokenizer . After obtaining the two serialized key points p_h and $p_{h'}$, we then utilize the KNN algorithm to select k nearest neighbors for each key point, forming n token patches $T_h \in \mathbb{R}^{n\times k\times 3}$ and $T_{h'} \in \mathbb{R}^{n\times k\times 3}$ with patch size k. To aggregate local information, points within each patch are normalized by subtracting the key point to obtain relative coordinates. We map the unbiased local patches to feature space using a lightweight PointNet [40] (point embedding layer), obtaining serialized point tokens $E_0^h \in \mathbb{R}^{n\times C}$ and $E_0^{h'} \in \mathbb{R}^{n\times C}$, where the former is the Hilbert-based and the latter is Trans-Hilbert-based.

Order indicator. Directly fed the two type serialized point tokens $E_0^h \in \mathbb{R}^{n\times C}$ and $E_0^{h'} \in \mathbb{R}^{n\times C}$ into Mamba encoder might cause confusion as E_0^h and $E_0^{h'}$ actually share the same center but with different order. Maintaining the distinct characteristics of these different scanning strategies is important for preserving the integrity of the spatial information. Thus, we propose an extremely simple order indicator to indicate the scanning strategy used. Specifically, the proposed order indicator performs the linear transformation to transfer features into different latent spaces. The formulation can be written as follows:

$$Z_0^h = E_0^h \odot \gamma_h + \beta_h, \quad Z_0^{h'} = E_0^{h'} \odot \gamma_{h'} + \beta_{h'}, \tag{3}$$

where $\gamma_h/\gamma_{h'} \in \mathbb{R}^C$ and $\beta_h/\beta_{h'} \in \mathbb{R}^C$ refer to the scale and shift factors, respectively. $\odot$ is the Hadamard product and is implemented by the broadcast mechanism. We then concat Z_0^h and $Z_0^{h'}$, resulting in $Z_0 \in \mathbb{R}^{2n\times C}$.

5

Mamba encoder. After obtaining the token $\boldsymbol{Z}_0$, we will feed it into the encoder, containing N × Mamba block, to extract the feature. Specifically, for each Mamba block, layer normalization (LN), Selective SSM, depth-wise convolution [10] (DW), and residual connections are employed. A standard Mamba layer is shown in Fig. 2, and the output can be summarized as follows:

$$\boldsymbol{Z}_{l-1}' = \text{LN}\left(\boldsymbol{Z}_{l-1}\right), \qquad \boldsymbol{Z}_l' = \sigma\left(\text{DW}\left(\text{Linear}\left(\boldsymbol{Z}_{l-1}'\right)\right)\right)$$
$$\boldsymbol{Z}_l'' = \sigma\left(\text{Linear}\left(\boldsymbol{Z}_{l-1}'\right)\right), \quad \boldsymbol{Z}_l = \text{Linear}\left(\text{SelectiveSSM}\left(\boldsymbol{Z}_l'\right) \times \boldsymbol{Z}_l''\right) + \boldsymbol{Z}_{l-1} \tag{4}$$

$\boldsymbol{Z}_l \in \mathbb{R}^{2n \times C}$ is the output of the l-th block, and σ indicates SiLU activation [26]. The SelectiveSSM is the key to the Mamba block, with a detailed description in Sec. 3. To better understand why the proposed PointMamba has global modeling capacity, we provide an intuitive visualization. As shown in Fig. 3, after modeling the first group of point tokens (i.e., Hilbert-based), the accumulated global information can improve the serialization process for the next set of tokens (i.e., Trans-Hilbert-based). This mechanism ensures that each serialized point in the Trans-Hilbert sequence is informed by the

Figure 3: An intuitive illustration of global modeling from PointMamba.

entire history of the previously processed Hilbert sequence, thereby enabling a more contextually rich and globally aware modeling process. More discussions can be found in Appendix A.2.

In our study, we show that even a very simple Mamba block without specific designs, our Point-Mamba can surpass the various Transformer-based point cloud analysis methods.

4.2 The serialization-based mask modeling

One intriguing characteristic of Transformers-based methods [33, 60, 6] is their improved performance using the pre-training scheme, especially mask modeling [25]. In this paper, considering the unidirectional modeling of Mamba, we customize a simple yet effective serialization-based mask modeling paradigm, as shown in Fig. 4.

Specifically, after obtaining the key points, we randomly choose Hilbert or Trans-Hilbert curve to implement serialization in each iteration, resulting in serialization-based key points, i.e., $\boldsymbol{p}_h$ and $\boldsymbol{p}_{h'}$, are obtained from Hilbert and Trans-Hilbert, respectively. Such a scheme allows the model to exact the local relationships from different scanning clues. Then, the KNN and the token embedding layer are used to generate the point tokens. To discriminate the point tokens serialized from which space-filling curves, we apply the order indicator to the point tokens, where different serialized point tokens have different order indicators, which are similar to the mentioned Eq. 3. Next, we randomly mask the serialization-based point tokens with a high ratio of 60%. Then, an asymmetric autoencoder, consisting of several vanilla Mamba blocks, is employed to extract the point feature, and the final layer of the autoencoder utilizes a simple prediction head for reconstruction. To reconstruct masked point patches in coordinate space, we employ a linear head to project the masked token to the shape of the masked input points. The Chamfer Distance [13] is then used as the reconstruction loss to recover the coordinates of the points in each masked point patch.

We demonstrate that with such a simple serialization-based mask modeling paradigm, Point-Mamba can easily achieve superior performance.

5 Experiments

5.1 Implementation details

SSM is new to 3D point cloud analysis, with no existing works detailing the specific implementation. To handle different resolutions of the input point cloud, we divide them into different numbers of patches with a linear scaling (e.g., $M = 1024$ input points are divided into $n = 64$ point patches), with each patch containing $k = 32$ points determined by the KNN algorithm. The PointMamba encoder has $N = 12$ vanilla Mamba blocks, each Mamba block featuring $C = 384$ hidden dimensions. For the pre-training, we utilize ShapeNetCore [5] as the dataset, following previous methods [60, 38, 6]. In addition, we utilize $4 \times$ Mamba blocks as the decoder to reconstruct the masked point clouds.

6
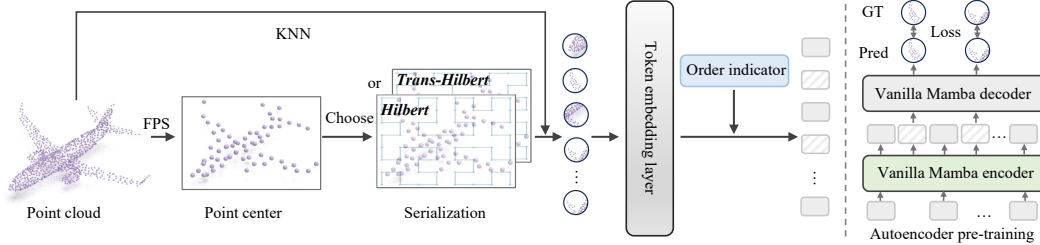
Figure 4: The details of our proposed serialization-based mask modeling. During the pre-training, we randomly choose one space-filling curve to generate the serialized point tokens for mask modeling, and different serialized point tokens have different order indicators.

Table 1: Object classification on the ScanObjectNN dataset [49]. We evaluate PointMamba on three variants, with PB-T50-RS being the most challenging. Overall accuracy (%) is reported. Param. denotes the number of tunable parameters during training. † indicates that using simple rotational augmentation [11] for training.

Methods	Reference	Backbone	Param. (M) ↓	FLOPs (G) ↓	OBJ-BG ↑	OBJ-ONLY ↑	PB-T50-RS ↑
Supervised Learning Only							
PointNet [40]	CVPR 17	-	3.5	0.5	73.3	79.2	68.0
PointNet++ [41]	NeurIPS 17	-	1.5	1.7	82.3	84.3	77.9
PointCNN [32]	NeurIPS 18	-	0.6	0.9	86.1	85.5	78.5
DGCNN [52]	TOG 19	-	1.8	2.4	82.8	86.2	78.1
PRANet [9]	TIP 21	-	-	-	-	-	81.0
MVTN [23]	ICCV 21	-	11.2	43.7	-	-	82.8
PointNeXt [44]	NeurIPS 22	-	1.4	1.6	-	-	87.7
PointMLP [37]	ICLR 22	-	13.2	31.4	-	-	85.4
RepSurf-U [45]	CVPR 22	-	1.5	0.8	-	-	84.3
ADS [28]	ICCV 23	-	-	-	-	-	87.5
Training from pre-training (Single-Modal)							
Point-BERT [60]	CVPR 22	Transformer	22.1	4.8	87.43	88.12	83.07
MaskPoint [33]	CVPR 22	Transformer	22.1	4.8	89.30	88.10	84.30
Point-MAE [38]	ECCV 22	Transformer	22.1	4.8	90.02	88.29	85.18
Point-M2AE [64]	NeurIPS 22	Transformer	15.3	3.6	91.22	88.81	86.43
PointDif [69]	CVPR 24	Transformer	-	-	93.29	91.91	87.61
Point-MAE+IDPT [63]	ICCV 23	Transformer	1.7	7.2	91.22	90.02	84.94
Point-MAE+DAPT [70]	CVPR 24	Transformer	1.1	5.0	90.88	90.19	85.08
Point-MAE† [38]	ECCV 22	Transformer	22.1	4.8	92.77	91.22	89.04
PointGPT-S† [6]	NeurIPS 23	Transformer	29.2	5.7	93.39	92.43	89.17
PointMamba† (**ours**)	-	**Mamba**	**12.3**	**3.1**	**94.32**	**92.60**	**89.31**
Training from pre-training (Cross-Modal)							
ACT† [11]	ICLR 23	Transformer	22.1	4.8	93.29	91.91	88.21
Joint-MAE [21]	IJCAI 23	Transformer	-	-	90.94	88.86	86.07
I2P-MAE† [65]	CVPR 23	Transformer	15.3	-	94.15	91.57	90.11
RECON† [42]	ICML 23	Transformer	43.6	5.3	**95.18**	**93.29**	**90.63**

5.2 Compared with Transformer-based counterparts

This paper aims to unlock the potential of Mamba in point cloud tasks, discussing whether it can be a viable alternative to Transformers. Thus, in the following experiments, **we mainly compare with the state-of-the-art vanilla Transformer-based point cloud analysis methods.**

Real-world object classification on ScanObjectNN. ScanObjectNN [49] is a challenging 3D dataset comprising about 15,000 objects across 15 categories, scanned from real-world indoor scenes with cluttered complexity backgrounds. As shown in Tab. 1, we conduct experiments on three versions of ScanObjectNN (i.e., OBJ-BG, OBJ-ONLY, and PB-T50-RS), each with increasing complexity. When compared with the most convincing Transformer-based method, i.e., Point-MAE [33], PointMamba surpasses it by 1.55%, 1.38% and 0.27% on OBJ-BG, OBJ-ONLY, and PB-T50-RS respectively while using less computational costs. Besides, we also outperform the SOTA PointGPT-S [6] by 0.93%, 0.17%, 0.14% across three variants on a comparable scale setting. Note that our method follows *Occam's Razor*, without auxiliary tasks like generation during fine-tuning

7

Table 2: Classification on ModelNet40 [57]. Overall accuracy (%) is reported. The results are obtained from 1024 points without voting.

Methods	Param. (M) ↓	FLOPs (G) ↓	OA (%) ↑
Supervised Learning Only			
PointNet [40]	3.5	0.5	89.2
PointNet++ [41]	1.5	1.7	90.7
PointCNN [32]	0.6	-	92.2
DGCNN [39]	1.8	2.4	92.9
PointNeXt [44]	1.4	1.6	92.9
PCT [20]	2.9	2.3	93.2
OctFormer [51]	3.98	31.3	92.7
with Self-supervised pre-training			
Point-BERT [60]	22.1	2.3	92.7
MaskPoint [33]	22.1	2.3	92.6
Point-M2AE [64]	12.8	4.6	93.4
Point-MAE [38]	22.1	2.4	93.2
PointGPT-S [6]	29.2	2.9	93.3
ACT [11]	22.1	2.4	**93.6**
PointMamba (**ours**)	**12.3**	**1.5**	**93.6**

Table 3: Few-shot learning on ModelNet40 [57]. Overall accuracy (%)±the standard deviation (%) without voting is reported.

Methods	5-way		10-way	
	10-shot	20-shot	10-shot	20-shot
Supervised Learning Only				
PointNet [40]	52.0±3.8	57.8±4.9	46.6±4.3	35.2±4.8
PointNet-CrossPoint [1]	90.9±1.9	93.5±4.4	84.6±4.7	90.2±2.2
DGCNN [52]	31.6±2.8	40.8±4.6	19.9±2.1	16.9±1.5
DGCNN-CrossPoint [1]	92.5±3.0	94.9±2.1	83.6±5.3	87.9±4.2
with Self-supervised pre-training				
Point-BERT [60]	94.6±3.1	96.3±2.7	91.0±5.4	92.7±5.1
MaskPoint [33]	95.0±3.7	97.2±1.7	91.4±4.0	93.4±3.5
Point-MAE [38]	96.3±2.5	97.8±1.8	92.6±4.1	95.0±3.0
Point-M2AE [64]	96.8±1.8	98.3±1.4	92.3±4.5	95.0±3.0
PointGPT-S [6]	96.8±2.0	98.6±1.1	92.6±4.6	95.2±3.4
ACT [11]	96.8±2.3	98.0±1.4	**93.3**±4.0	**95.6**±2.8
PointMamba (**ours**)	**96.9**±2.0	**99.0**±1.1	93.0±4.4	**95.6**±3.2

used in PointGPT [6]. Furthermore, compared to cross-modal learning methods [11, 42] that use additional training data (cross-modal information) or teacher models, which is not a fair comparison, our PointMamba still maintains highly competitive. We mainly want to introduce a new Mamba-based point cloud analysis methods paradigm. Although using some complex designs can bring improvement, they might be heuristics. More importantly, these heuristic designs will decrease the objectivity of the evaluation of our method.

Synthetic object classification on ModelNet40. ModelNet40 [57] is a pristine 3D CAD dataset consisting of 12,311 clean samples across 40 categories. As shown in Tab. 2, we report the overall accuracy without adopting the voting strategy. The proposed PointMamba achieves the best results compared with various self-supervised Transformer-based methods [33, 60, 6]. In particular, PointMamba surpasses Point-MAE [33] and PointGPT-S [6] by 0.4% and 0.3% respectively. It is worth noting that the single-modal-learned PointMamba achieves comparable results with the cross-modal-based ACT [11] while significantly reducing parameters and FLOPs about 44% and 38%, respectively. Additionally, PointMamba demonstrates competitive performance against elaborately designed Transformer models like OctFormer [51].

Few-shot learning. We further conduct few-shot experiments on ModelNet40 [57] to demonstrate our few-shot transfer ability. Consistent with prior studies [60], we utilize the "n-way, m-shot" setup, where $n \in \{5, 10\}$ denotes the category count and $m \in \{10, 20\}$ represents the samples per category. Following standard procedure, we carry out 10 separate experiments for each setting and reported mean accuracy along with the standard deviation. As indicated in Tab. 3, our PointMamba shows competitive results with limited data, e.g., +1.0% mean accuracy compared to the cross-modal method ACT [11] on the 5-way 20-shot split.

Part segmentation on ShapeNetPart. Part segmentation on ShapeNetPart [58] is a challenging task that aims to predict a more detailed label for each point within a sample. As shown in Tab. 4, we report mean IoU (mIoU) for all classes (Cls.) and all instances (Inst.). Our PointMamba model demonstrates highly competitive performance compared to the Transformer-based counterparts [33, 6, 60]. These impressive results further prove the potential of SSM in the point cloud analysis tasks.

Table 4: Part segmentation on the ShapeNet-Part [58]. The mIoU for all classes (Cls.) and for all instances (Inst.) are reported.

Methods	Cls. mIoU (%) ↑	Inst. mIoU (%) ↑
Supervised Learing Only		
PointNet [40]	80.39	83.7
PointNet++ [41]	81.85	85.1
DGCNN [52]	82.33	85.2
APES [54]	83.67	85.8
with Self-supervised pre-training		
Transformer [60]	83.4	85.1
OcCo [60]	83.4	85.1
MaskPoint [33]	84.6	86.0
Point-BERT [60]	84.1	85.6
Point-MAE [38]	84.2	86.1
PointGPT-S [6]	84.1	**86.2**
ACT [11]	**84.7**	86.1
PointMamba (**ours**)	84.4	**86.2**

8

(a) w/ Identity   (b) w/ Attention   (c) w/ MLP   (d) w/ Selective SSM (**ours**)
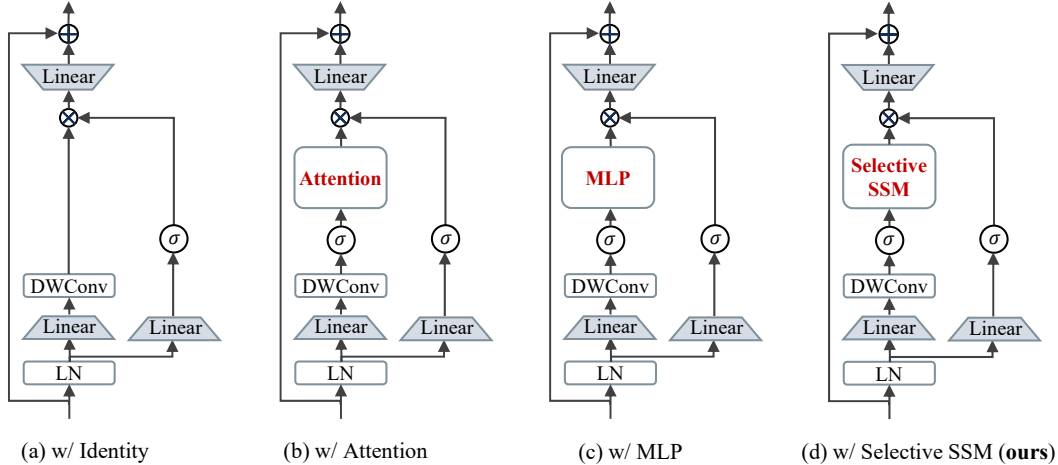
Figure 5: Different variant of PointMamba. (a) Directly removing the SSM part. (b) Replacing SSM with attention. (c) Replacing SSM with MLP. (d) Ours PointMamba with Selective SSM.

Table 5: The effect of each component.

Hilbert	Trans-Hilbert	Order indicator	OBJ-BG	OBJ-ONLY
Random		-	92.60	90.18
✓	-	-	92.94	91.05
-	✓	-	93.46	91.74
✓	✓	-	93.80	91.91
✓	✓	✓	**94.32**	**92.60**

Table 6: The effect of different scanning curves.

Scanning curve	OBJ-BG	OBJ-ONLY
Random	92.60	90.18
Hilbert and Trans-Hilbert	**94.32**	**92.60**
Z-order and Trans-Z-zorder	93.29	90.36
Hilbert and Z-order	93.29	90.88
Trans-Hilbert and Trans-Z-order	93.29	91.91

Table 7: The effect of Selective SSM.

Setting	Param.	OBJ-BG	OBJ-ONLY
w/ Identity	11.4	93.80	91.57
w/ Attention	39.8	92.77	91.22
w/ MLP	18.5	93.29	91.22
w/ Selective SSM	12.3	**94.32**	**92.60**

Table 8: The effect of Order indicator.

Setting	OBJ-BG	OBJ-ONLY
None	93.80	91.91
Using the same indicator	93.29	90.19
Using different indicator	**94.32**	**92.60**

5.3 Analysis and ablation study

To investigate the architecture design, we conduct ablation studies on ScanObjectNN [49] with both pre-training and fine-tuning. Default settings are marked in `gray`.

The structural efficiency. We first discuss the efficiency of our method. To fully explore the potential of processing the long point tokens (sequence), we gradually increase the sequence length until the GPU (NVIDIA A800 80GB) memory explodes. The comprehensive efficiency comparisons are present in Fig. 1(b)-(d), where Compared with the most convincing Transformer-based method [33], our PointMamba demonstrates significantly improved inference speed and reduce the GPU usage and FLOPs, especially when facing the long sequence. For example, when the length increases to more than 32,768, we outperform PointMAE by $30.2\times$, $24.9\times$, and $5.2\times$ in terms of inference speed, GPU memory, and FLOPs, respectively. More importantly, even presenting impressive efficiency, we still achieve impressive performance on various point cloud analysis datasets.

The effect of each component. We then study the effectiveness of the proposed components of PointMamba as shown in Tab. 5. We can make the following observations: 1) Directly utilizing random serialization, PointMamba only achieves 92.26% and 90.18% overall accuracy on OBJ-BG and OBJ-ONLY, respectively. It is reasonable as Mamba is hard to model the unstructured point clouds due to its unidirectional modeling. 2) By introducing the locality-preserved Hilbert or Trans-Hilbert scanning, PointMamba's ability to capture sequence information is enhanced, leading to performance improvements compared to random serialization. Further applying both Hilbert and Trans-Hilbert scanning curves, PointMamba surpasses the random serialization by 1.20% and 1.73% on two datasets, respectively. 3) By using the order indicator to maintain the distinct characteristics

9

of the two different scanning strategies, we achieve notable improvement, resulting in 94.32% and 92.60% on OBJ-BG and OBJ-ONLY, respectively. Note that the order indicator is extremely light (only 1.5k parameters), which will not introduce additional computational costs.

The effect of different scanning curves. We further explore the effect of using different scanning curves to construct serialized point tokens. Specifically, we select two widely used space-filling curves, including Hilbert and Z-order, along with their transposed variants, i.e., Trans-Hilbert and Trans-Z-order. As listed in Tab. 6, we empirically find that serializing point clouds with space-filling curves scanning can achieve better performance compared to random sequences. We argue that scanning sequences along a specific pattern of spatial locations offers a more logical sequence modeling order for SSM. We choose the combination of Hilbert and Trans-Hilbert for PointMamba due to their superior locality-preserving properties.

The effect of Selective SSM. The key of S6 models or Mamba [14] is the SSM with the selective mechanism. We prove that as a unidirectional modeling method, SSM can be analogous to masked self-attention, ensuring each position can only attend to previous positions (the detailed proof can be found in the Appendix A.1). Thus, as shown in Tab. 7, we analyze the effect of selective SSM by removing it (i.e., identity setting) or replace with masked self-attention or MLP (an illustration is shown in Fig. 5). Compared with the identity setting, the selective SSM brings notable improvement, indicating the effectiveness of introducing global modeling from SSM. Note that while a very recent method, MambaOut [59], thinks the SSM of Mamba might negatively impact image classification tasks, our findings demonstrate that this is not the case for point cloud analysis tasks. Another interesting thing is that when Selective SSM is replaced with masked self-attention, the performance is even lower than that of the identity setting. We argue the main reason is that masked self-attention is hard to combine with the Gated MLP [46] used in default Mamba, leading to optimized difficulty, which might need to be explored in the future.

Analysis on order indicator. This part analyzes the effect of the order indicator. PointMamba applies Hilbert and Trans-Hilbert to recognize the key points, obtaining two types of serialized point tokens $\boldsymbol{E}_0^h$ and $\boldsymbol{E}_0^{h'}$. The order indicator is used to indicate the scanning strategy. As shown in Tab. 8, using two different order indicators can improve 1.20% and 1.03% compared to no indicator on OBJ-BG and OBJ-ONLY, respectively. However, using the same order indicator for both types of sequences without distinguishing between different scanning strategies does not yield positive results.

5.4 Limitation

Although PointMamba achieves promising results, there are some limitations: 1) We only focus on the point cloud analysis task in this paper while designing a unified Mamba-based foundation model for various 3D vision tasks (e.g., 3D object classification/detection/segmentation) is a more appealing direction. 2) We only use the point clouds as training data while combining them with 2D images or language knowledge to improve the performance, which is also worthy of exploration. We left these in our future work.

6 Conclusion

In this paper, we present an elegant, simple Mamba-based method named PointMamba for point cloud analysis. PointMamba utilizes a space-filling curve-based point tokenizer and a plain, non-hierarchical Mamba architecture to achieve global modeling with linear complexity. Despite its structural simplicity, PointMamba delivers state-of-the-art performance across various datasets, significantly reducing computational costs in terms of GPU memory and FLOPs. PointMamba success highlights the potential of SSMs, particularly Mamba, in handling the complexities of point cloud data. As a newcomer to point cloud analysis, PointMamba is a promising option for constructing 3D vision foundation models, and we hope it can offer a new perspective for the field.

Acknowledgements

This work is supported in part by the NSFC (Grant. No. 62225603 and 623B2038), in part by the Taihu Lake Innovation Fund for Future Technology (HUST: 2023-A-1), and in part by the Fundamental Research Funds for the Central Universities (HUST: 52400036).

10

References

[1] Mohamed Afham, Isuru Dissanayake, Dinithi Dissanayake, Amaya Dharmasiri, Kanchana Thilakarathna, and Ranga Rodrigo. Crosspoint: Self-supervised cross-modal contrastive learning for 3d point cloud understanding. In *Proc. of IEEE Intl. Conf. on Computer Vision and Pattern Recognition*, 2022.

[2] Ameen Ali, Itamar Zimerman, and Lior Wolf. The hidden attention of mamba models. *arXiv preprint arXiv:2403.01590*, 2024.

[3] Ali Behrouz and Farnoosh Hashemi. Graph mamba: Towards learning on graphs with state space models. In *Proc. of ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 119–130, 2024.

[4] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. In *Proc. of Advances in Neural Information Processing Systems*, 2020.

[5] Angel X Chang, Thomas Funkhouser, Leonidas Guibas, Pat Hanrahan, Qixing Huang, Zimo Li, Silvio Savarese, Manolis Savva, Shuran Song, Hao Su, et al. Shapenet: An information-rich 3d model repository. *arXiv preprint arXiv:1512.03012*, 2015.

[6] Guangyan Chen, Meiling Wang, Yi Yang, Kai Yu, Li Yuan, and Yufeng Yue. Pointgpt: Auto-regressively generative pre-training from point clouds. In *Proc. of Advances in Neural Information Processing Systems*, 2023.

[7] Guo Chen, Yifei Huang, Jilan Xu, Baoqi Pei, Zhe Chen, Zhiqi Li, Jiahao Wang, Kunchang Li, Tong Lu, and Limin Wang. Video mamba suite: State space model as a versatile alternative for video understanding. *arXiv preprint arXiv:2403.09626*, 2024.

[8] Yukang Chen, Jianhui Liu, Xiangyu Zhang, Xiaojuan Qi, and Jiaya Jia. Voxelnext: Fully sparse voxelnet for 3d object detection and tracking. In *Proc. of IEEE Intl. Conf. on Computer Vision and Pattern Recognition*, 2023.

[9] Silin Cheng, Xiwu Chen, Xinwei He, Zhe Liu, and Xiang Bai. Pra-net: Point relation-aware network for 3d point cloud analysis. *IEEE Transactions on Image Processing*, 30:4436–4448, 2021.

[10] François Chollet. Xception: Deep learning with depthwise separable convolutions. In *Proc. of IEEE Intl. Conf. on Computer Vision and Pattern Recognition*, 2017.

[11] Runpei Dong, Zekun Qi, Linfeng Zhang, Junbo Zhang, Jianjian Sun, Zheng Ge, Li Yi, and Kaisheng Ma. Autoencoders as cross-modal teachers: Can pretrained 2d image transformers help 3d representation learning? In *Proc. of Intl. Conf. on Learning Representations*, 2022.

[12] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. In *Proc. of Intl. Conf. on Learning Representations*, 2021.

[13] Haoqiang Fan, Hao Su, and Leonidas J Guibas. A point set generation network for 3d object reconstruction from a single image. In *Proc. of IEEE Intl. Conf. on Computer Vision and Pattern Recognition*, 2017.

[14] Albert Gu and Tri Dao. Mamba: Linear-time sequence modeling with selective state spaces. In *Conference on Language Modeling*, 2024.

[15] Albert Gu, Tri Dao, Stefano Ermon, Atri Rudra, and Christopher Ré. Hippo: Recurrent memory with optimal polynomial projections. In *Proc. of Advances in Neural Information Processing Systems*, 2020.

[16] Albert Gu, Karan Goel, Ankit Gupta, and Christopher Ré. On the parameterization and initialization of diagonal state space models. In *Proc. of Advances in Neural Information Processing Systems*, 2022.

[17] Albert Gu, Karan Goel, and Christopher Re. Efficiently modeling long sequences with structured state spaces. In *Proc. of Intl. Conf. on Learning Representations*, 2021.

[18] Albert Gu, Isys Johnson, Karan Goel, Khaled Saab, Tri Dao, Atri Rudra, and Christopher Ré. Combining recurrent, convolutional, and continuous-time models with linear state space layers. In *Proc. of Advances in Neural Information Processing Systems*, 2021.

[19] Albert Gu, Isys Johnson, Aman Timalsina, Atri Rudra, and Christopher Re. How to train your hippo: State space models with generalized orthogonal basis projections. In *Proc. of Intl. Conf. on Learning Representations*, 2022.

11

[20] Meng-Hao Guo, Jun-Xiong Cai, Zheng-Ning Liu, Tai-Jiang Mu, Ralph R Martin, and Shi-Min Hu. Pct: Point cloud transformer. *Computational Visual Media*, 2021.

[21] Ziyu Guo, Renrui Zhang, Longtian Qiu, Xianzhi Li, and Pheng-Ann Heng. Joint-mae: 2d-3d joint masked autoencoders for 3d point cloud pre-training. In *Proc. of Intl. Joint Conf. on Artificial Intelligence*, 2023.

[22] Ankit Gupta, Albert Gu, and Jonathan Berant. Diagonal state spaces are as effective as structured state spaces. In *Proc. of Advances in Neural Information Processing Systems*, 2022.

[23] Abdullah Hamdi, Silvio Giancola, and Bernard Ghanem. Mvtn: Multi-view transformation network for 3d shape recognition. In *Porc. of IEEE Intl. Conf. on Computer Vision*, 2021.

[24] Xu Han, Yuan Tang, Zhaoxuan Wang, and Xianzhi Li. Mamba3d: Enhancing local features for 3d point cloud analysis via state space model. In *Proc. of ACM Multimedia*, 2024.

[25] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. In *Proc. of IEEE Intl. Conf. on Computer Vision and Pattern Recognition*, 2022.

[26] Dan Hendrycks and Kevin Gimpel. Gaussian error linear units (gelus). *arXiv preprint arXiv:1606.08415*, 2016.

[27] David Hilbert. Über die stetige abbildung einer linie auf ein flächenstück. *Dritter Band: Analysis· Grundlagen der Mathematik· Physik Verschiedenes: Nebst Einer Lebensgeschichte*, 1935.

[28] Cheng-Yao Hong, Yu-Ying Chou, and Tyng-Luh Liu. Attention discriminant sampling for point clouds. In *Porc. of IEEE Intl. Conf. on Computer Vision*, 2023.

[29] Vincent Tao Hu, Stefan Andreas Baumann, Ming Gui, Olga Grebenkova, Pingchuan Ma, Johannes Fischer, and Bjorn Ommer. Zigma: Zigzag mamba diffusion model. In *Proc. of European Conference on Computer Vision*, 2024.

[30] Kunchang Li, Xinhao Li, Yi Wang, Yinan He, Yali Wang, Limin Wang, and Yu Qiao. Videomamba: State space model for efficient video understanding. In *Proc. of European Conference on Computer Vision*, 2024.

[31] Shufan Li, Harkanwar Singh, and Aditya Grover. Mamba-nd: Selective state space modeling for multi-dimensional data. *arXiv preprint arXiv:2402.05892*, 2024.

[32] Yangyan Li, Rui Bu, Mingchao Sun, Wei Wu, Xinhan Di, and Baoquan Chen. Pointcnn: Convolution on x-transformed points. In *Proc. of Advances in Neural Information Processing Systems*, 2018.

[33] Haotian Liu, Mu Cai, and Yong Jae Lee. Masked discrimination for self-supervised learning on point clouds. In *Proc. of European Conference on Computer Vision*, 2022.

[34] Jiarun Liu, Hao Yang, Hong-Yu Zhou, Yan Xi, Lequan Yu, Cheng Li, Yong Liang, Guangming Shi, Yizhou Yu, Shaoting Zhang, et al. Swin-umamba: Mamba-based unet with imagenet-based pretraining. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 615–625. Springer, 2024.

[35] Yue Liu, Yunjie Tian, Yuzhong Zhao, Hongtian Yu, Lingxi Xie, Yaowei Wang, Qixiang Ye, and Yunfan Liu. Vmamba: Visual state space model. In *Proc. of Advances in Neural Information Processing Systems*, 2024.

[36] Jun Ma, Feifei Li, and Bo Wang. U-mamba: Enhancing long-range dependency for biomedical image segmentation. *arXiv preprint arXiv:2401.04722*, 2024.

[37] Xu Ma, Can Qin, Haoxuan You, Haoxi Ran, and Yun Fu. Rethinking network design and local geometry in point cloud: A simple residual mlp framework. In *Proc. of Intl. Conf. on Learning Representations*, 2022.

[38] Yatian Pang, Wenxiao Wang, Francis EH Tay, Wei Liu, Yonghong Tian, and Li Yuan. Masked autoencoders for point cloud self-supervised learning. In *Proc. of European Conference on Computer Vision*, 2022.

[39] Anh Viet Phan, Minh Le Nguyen, Yen Lam Hoang Nguyen, and Lam Thu Bui. Dgcnn: A convolutional neural network over large-scale labeled graphs. *Neural Networks*, 2018.

[40] Charles R Qi, Hao Su, Kaichun Mo, and Leonidas J Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *Proc. of IEEE Intl. Conf. on Computer Vision and Pattern Recognition*, 2017.

12

[41] Charles Ruizhongtai Qi, Li Yi, Hao Su, and Leonidas J Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. In *Proc. of Advances in Neural Information Processing Systems*, 2017.

[42] Zekun Qi, Runpei Dong, Guofan Fan, Zheng Ge, Xiangyu Zhang, Kaisheng Ma, and Li Yi. Contrast with reconstruct: Contrastive 3d representation learning guided by generative pretraining. In *Proc. of Intl. Conf. on Machine Learning*, 2023.

[43] Zekun Qi, Runpei Dong, Shaochen Zhang, Haoran Geng, Chunrui Han, Zheng Ge, Li Yi, and Kaisheng Ma. Shapellm: Universal 3d object understanding for embodied interaction. In *ECCV*, 2024.

[44] Guocheng Qian, Yuchen Li, Houwen Peng, Jinjie Mai, Hasan Hammoud, Mohamed Elhoseiny, and Bernard Ghanem. Pointnext: Revisiting pointnet++ with improved training and scaling strategies. In *Proc. of Advances in Neural Information Processing Systems*, 2022.

[45] Haoxi Ran, Jun Liu, and Chengjie Wang. Surface representation for point clouds. In *Proc. of IEEE Intl. Conf. on Computer Vision and Pattern Recognition*, 2022.

[46] Noam Shazeer. Glu variants improve transformer. *arXiv preprint arXiv:2002.05202*, 2020.

[47] Shaoshuai Shi, Zhe Wang, Jianping Shi, Xiaogang Wang, and Hongsheng Li. From points to parts: 3d object detection from point cloud with part-aware and part-aggregation network. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2020.

[48] Jimmy TH Smith, Andrew Warrington, and Scott Linderman. Simplified state space layers for sequence modeling. In *Proc. of Intl. Conf. on Learning Representations*, 2022.

[49] Mikaela Angelina Uy, Quang-Hieu Pham, Binh-Son Hua, Thanh Nguyen, and Sai-Kit Yeung. Revisiting point cloud classification: A new benchmark dataset and classification model on real-world data. In *Porc. of IEEE Intl. Conf. on Computer Vision*, 2019.

[50] Chloe Wang, Oleksii Tsepa, Jun Ma, and Bo Wang. Graph-mamba: Towards long-range graph sequence modeling with selective state spaces. *arXiv preprint arXiv:2402.00789*, 2024.

[51] Peng-Shuai Wang. Octformer: Octree-based transformers for 3d point clouds. *ACM Transactions ON Graphics*, 2023.

[52] Yue Wang, Yongbin Sun, Ziwei Liu, Sanjay E Sarma, Michael M Bronstein, and Justin M Solomon. Dynamic graph cnn for learning on point clouds. *ACM Transactions ON Graphics*, 2019.

[53] Zicheng Wang, Zhenghao Chen, Yiming Wu, Zhen Zhao, Luping Zhou, and Dong Xu. Pointramba: A hybrid transformer-mamba framework for point cloud analysis. *arXiv preprint arXiv:2405.15463*, 2024.

[54] Chengzhi Wu, Junwei Zheng, Julius Pfrommer, and Jürgen Beyerer. Attention-based point cloud edge sampling. In *Proc. of IEEE Intl. Conf. on Computer Vision and Pattern Recognition*, 2023.

[55] Xiaoyang Wu, Li Jiang, Peng-Shuai Wang, Zhijian Liu, Xihui Liu, Yu Qiao, Wanli Ouyang, Tong He, and Hengshuang Zhao. Point transformer v3: Simpler, faster, stronger. In *Proc. of IEEE Intl. Conf. on Computer Vision and Pattern Recognition*, 2024.

[56] Xiaoyang Wu, Yixing Lao, Li Jiang, Xihui Liu, and Hengshuang Zhao. Point transformer v2: Grouped vector attention and partition-based pooling. In *Proc. of Advances in Neural Information Processing Systems*, 2022.

[57] Zhirong Wu, Shuran Song, Aditya Khosla, Fisher Yu, Linguang Zhang, Xiaoou Tang, and Jianxiong Xiao. 3d shapenets: A deep representation for volumetric shapes. In *Proc. of IEEE Intl. Conf. on Computer Vision and Pattern Recognition*, 2015.

[58] Li Yi, Vladimir G Kim, Duygu Ceylan, I-Chao Shen, Mengyan Yan, Hao Su, Cewu Lu, Qixing Huang, Alla Sheffer, and Leonidas Guibas. A scalable active framework for region annotation in 3d shape collections. *ACM Transactions ON Graphics*, 2016.

[59] Weihao Yu and Xinchao Wang. Mambaout: Do we really need mamba for vision? *arXiv preprint arXiv:2405.07992*, 2024.

[60] Xumin Yu, Lulu Tang, Yongming Rao, Tiejun Huang, Jie Zhou, and Jiwen Lu. Point-bert: Pre-training 3d point cloud transformers with masked point modeling. In *Proc. of IEEE Intl. Conf. on Computer Vision and Pattern Recognition*, 2022.

13

[61] Yaohua Zha, Huizhen Ji, Jinmin Li, Rongsheng Li, Tao Dai, Bin Chen, Zhi Wang, and Shu-Tao Xia. Towards compact 3d representations via point feature enhancement masked autoencoders. In *Proc. of the AAAI Conf. on Artificial Intelligence*, 2024.

[62] Yaohua Zha, Naiqi Li, Yanzi Wang, Tao Dai, Hang Guo, Bin Chen, Zhi Wang, Zhihao Ouyang, and Shu-Tao Xia. Lcm: Locally constrained compact point cloud model for masked point modeling. In *Proc. of Advances in Neural Information Processing Systems*, 2024.

[63] Yaohua Zha, Jinpeng Wang, Tao Dai, Bin Chen, Zhi Wang, and Shu-Tao Xia. Instance-aware dynamic prompt tuning for pre-trained point cloud models. In *Porc. of IEEE Intl. Conf. on Computer Vision*, 2023.

[64] Renrui Zhang, Ziyu Guo, Peng Gao, Rongyao Fang, Bin Zhao, Dong Wang, Yu Qiao, and Hongsheng Li. Point-m2ae: multi-scale masked autoencoders for hierarchical point cloud pre-training. In *Proc. of Advances in Neural Information Processing Systems*, 2022.

[65] Renrui Zhang, Liuhui Wang, Yu Qiao, Peng Gao, and Hongsheng Li. Learning 3d representations from 2d pre-trained models via image-to-point masked autoencoders. In *Proc. of IEEE Intl. Conf. on Computer Vision and Pattern Recognition*, 2023.

[66] Tao Zhang, Xiangtai Li, Haobo Yuan, Shunping Ji, and Shuicheng Yan. Point could mamba: Point cloud learning via state space model. *arXiv preprint arXiv:2403.00762*, 2024.

[67] Hengshuang Zhao, Jiaya Jia, and Vladlen Koltun. Exploring self-attention for image recognition. In *Proc. of IEEE Intl. Conf. on Computer Vision and Pattern Recognition*, 2020.

[68] Hengshuang Zhao, Li Jiang, Jiaya Jia, Philip HS Torr, and Vladlen Koltun. Point transformer. In *Porc. of IEEE Intl. Conf. on Computer Vision*, 2021.

[69] Xiao Zheng, Xiaoshui Huang, Guofeng Mei, Yuenan Hou, Zhaoyang Lyu, Bo Dai, Wanli Ouyang, and Yongshun Gong. Point cloud pre-training with diffusion models. In *Proc. of IEEE Intl. Conf. on Computer Vision and Pattern Recognition*, 2024.

[70] Xin Zhou, Dingkang Liang, Wei Xu, Xingkui Zhu, Yihan Xu, Zhikang Zou, and Xiang Bai. Dynamic adapter meets prompt tuning: Parameter-efficient transfer learning for point cloud analysis. In *Proc. of IEEE Intl. Conf. on Computer Vision and Pattern Recognition*, 2024.

[71] Lianghui Zhu, Bencheng Liao, Qian Zhang, Xinlong Wang, Wenyu Liu, and Xinggang Wang. Vision mamba: Efficient visual representation learning with bidirectional state space model. In *Proc. of Intl. Conf. on Machine Learning*, 2024.

14

Appendix A   Theoretically Analysis

A.1   Closer look at Selective SSM

As described in Sec. 3, Selective SSM [14] considers parameters $\boldsymbol{B}, \boldsymbol{C}, \boldsymbol{\Delta}$ in Eq. 2 as functions of the input. To be specific, given the input sequence $\hat{\boldsymbol{x}} = [x_1, \cdots, x_t, \cdots, x_L] \in \mathbb{R}^{L \times C}$, the per-time matrices $\boldsymbol{B}_t, \boldsymbol{C}_t, \boldsymbol{\Delta}_t$ can be computed as follows:

$$\boldsymbol{B}_t = L_B(x_t), \quad \boldsymbol{C}_t = L_C(x_t), \quad \boldsymbol{\Delta}_t = \text{softplus}\left(L_\Delta(x_t)\right), \tag{5}$$

where L_B, L_C, L_Δ are linear projection layers, and $\text{softplus}(x) = \log(1 + e^x)$. The matrices $\overline{\boldsymbol{A}}_t, \overline{\boldsymbol{B}}_t, \boldsymbol{C}_t, \boldsymbol{\Delta}_t$ can be obtained by taking Eq. 5 into Eq. 2.

To simplify, we ignore the residual connection $\boldsymbol{D}$ and expand Eq. 1, the output $\hat{\boldsymbol{y}} = [y_1, \cdots, y_t, \cdots, y_L] \in \mathbb{R}^{L \times C}$ can be computed below:

$$y_t = \boldsymbol{C}_t h_t, \quad h_t = \sum_{i=1}^{t} \left(\prod_{j=i+1}^{t} \overline{\boldsymbol{A}}_j \right) \overline{\boldsymbol{B}}_i x_i, \tag{6}$$

which can be further described in matrix form below:

$$\begin{bmatrix} h_1 \\ h_2 \\ \vdots \\ h_t \end{bmatrix} = \begin{bmatrix} \overline{\boldsymbol{B}}_1 & 0 & \cdots & 0 \\ \overline{\boldsymbol{A}}_2 \overline{\boldsymbol{B}}_1 & \overline{\boldsymbol{B}}_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ \prod_{i=2}^{t} \overline{\boldsymbol{A}}_i \overline{\boldsymbol{B}}_1 & \prod_{i=3}^{t} \overline{\boldsymbol{A}}_i \overline{\boldsymbol{B}}_2 & \cdots & \overline{\boldsymbol{B}}_t \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_t \end{bmatrix}. \tag{7}$$

Drawing inspiration from the previous work [2], we further provide an intuitive understanding. The Eq. 7 resembles the self-attention mechanism with a mask $\boldsymbol{M}$, specifically causal self-attention. In this context, $\boldsymbol{M}$ is a lower triangular matrix with elements set to 1. To further exam this, consider the transfer matrix $\boldsymbol{W}$ between $\hat{y}$ and $\hat{x}$, i.e., $(\hat{\boldsymbol{y}} = \boldsymbol{W}\hat{\boldsymbol{x}})$:

$$\boldsymbol{W}_{i,j} = \boldsymbol{C}_i \left(\prod_{k=j+1}^{i} \overline{\boldsymbol{A}}_k \right) \overline{\boldsymbol{B}}_j \tag{8}$$

$$= \boldsymbol{C}_i \left(\prod_{k=j+1}^{i} \exp\left(\boldsymbol{\Delta}_k \boldsymbol{A}\right) \right) \overline{\boldsymbol{B}}_j \tag{9}$$

$$= \boldsymbol{C}_i \exp \left(\sum_{k=j+1}^{i} \boldsymbol{\Delta}_k \boldsymbol{A} \right) \overline{\boldsymbol{B}}_j \tag{10}$$

$$\approx \boldsymbol{C}_i \exp \left(\sum_{\substack{k=j+1 \\ L_\Delta(x_k) > 0}}^{i} L_\Delta\left(x_k\right) \boldsymbol{A} \right) \overline{\boldsymbol{B}}_j, \tag{11}$$

where $\boldsymbol{W}_{i,j}$ represents the element in the i-th row and j-th column, the approximation in Eq. 11 is done using ReLU instead of softplus. Consider the notation below:

$$\boldsymbol{Q}_i := \boldsymbol{C}_i, \quad \boldsymbol{T}_{i,j} = \exp \left(\sum_{\substack{k=j+1 \\ L_\Delta(x_k) > 0}}^{i} \left(L_\Delta\left(x_k\right) \boldsymbol{A}\right) \right), \quad \boldsymbol{K}_j = \left(\overline{\boldsymbol{B}}_j\right)^T. \tag{12}$$

Thus, the Eq. 11 can be simplified to:

$$\boldsymbol{W}_{i,j} \approx \boldsymbol{Q}_i \boldsymbol{T}_{i,j} \boldsymbol{K}_j^T. \tag{13}$$

This shows that the Selective SSM captures the influence of x_i and x_j using $\boldsymbol{Q}_i$ and $\boldsymbol{K}_j$, respectively, while $\boldsymbol{T}_{i,j}$ molding the token significance from x_i to x_j. Note that $i \leq j$ because $\boldsymbol{W}$ is a lower triangular matrix, indicating a strong relationship with causal self-attention [4, 2].

15

A.2 Global modeling of PointMamba

In this subsection, we explain the global modeling of our PointMamba. Let's consider the total input sequence as $\left[\hat{l}_1; \hat{l}_2\right] = \left[x_1,, \cdots, x_{l/2}; x_{l/2+1}, \cdots, x_l\right]$ where the sequence has a length l and l is even. $\hat{l}_1$ comes from Hilbert serialization and the other half, $\hat{l}_2$, comes from Trans-Hilbert. The large matrix in Eq. 7 can be represented as a partitioned matrix $\begin{bmatrix} X & 0 \\ Y & Z \end{bmatrix}$ as below:

$$
\begin{bmatrix}
\overline{B}_1 & & & & & & \\
\vdots & \ddots & & & & & \\
\prod_{i=2}^{\frac{l}{2}}\overline{A}_i\overline{B}_1 & \cdots & \overline{B}_{\frac{l}{2}} & & & & \\
\prod_{i=2}^{\frac{l}{2}+1}\overline{A}_i\overline{B}_1 & \cdots & \overline{A}_{\frac{l}{2}+1}\overline{B}_{\frac{l}{2}} & & \overline{B}_{\frac{l}{2}+1} & & \\
\vdots & \ddots & \vdots & & \vdots & \ddots & \\
\prod_{i=2}^{l}\overline{A}_i\overline{B}_1 & \cdots & \prod_{i=\frac{l}{2}+1}^{l}\overline{A}_i\overline{B}_{\frac{l}{2}} & \prod_{i=\frac{l}{2}+2}^{l}\overline{A}_i\overline{B}_{\frac{l}{2}+1} & \cdots & \overline{B}_l
\end{bmatrix}
\tag{14}
$$

Note that the block Y, highlighted in gray , is associated with both $\hat{l}_1$ (from $\overline{B}$) and $\hat{l}_2$ (from $\overline{A}$), denoted as $Y(\hat{l}_1, \hat{l}_2)$. The blocks X and Z only relate to half of the sequence, denoted as $X(\hat{l}_1)$ and $Z(\hat{l}_2)$, respectively. Thus, the hidden space output $\left[\hat{h}_1; \hat{h}_2\right]^T = \left[h_1, \cdots, h_{l/2}; h_{l/2+1}, \cdots, h_l\right]^T$ can be compressed as below:

$$
\begin{bmatrix} \hat{h}_1 \\ \hat{h}_2 \end{bmatrix} = \begin{bmatrix} X(\hat{l}_1)\hat{l}_1 \\ Y(\hat{l}_1, \hat{l}_2)\hat{l}_2 + Z(\hat{l}_2)\hat{l}_2 \end{bmatrix}.
\tag{15}
$$

As in Fig. 3 and Eq. 15, the serialized points from Trans-Hilbert can receive global information from Hilbert serialization.

Appendix B Concurrent Related Works

Table 9: Classification performance comparisons with other state space model methods on three variants of the ScanObjectNN [49]. All results are reported without voting.

Method	Param. (M)	FLOPs (G)	ScanObjectNN		
			OBJ_BG	OBJ_ONLY	PB_T50_RS
Point Cloud Mamba [66]	34.2	45.0	-	-	88.10
Mamba3D [24]	16.9	3.9	93.12	92.08	88.20
PoinTramba [53]	19.5	-	92.30	91.30	89.10
PointMamba (**ours**)	**12.3**	**3.1**	**94.32**	**92.60**	**89.31**

Some works on state space models for point cloud analysis appeared recently. This section discusses the differences between these methods and our PointMamba.

Point Cloud Mamba (PCM) [66] combines an improved Mamba module, i.e., Vision Mamba [71], with PointMLP [37] (a strong point cloud analysis method), and incorporates consistent traverse serialization at each stage. To enhance Mamba's capability in managing point sequences with varying orders, PCM introduces point prompts that convey the sequence's arrangement rules. While these techniques improve the performance of state space models, they also introduce additional computational overhead and complexity in design.

Mamba3D [24] is another recently proposed method. To obtain better global features, Mamba3D introduces an enhanced Vision Mamba [71] block, which includes both a token forward SSM and a

16
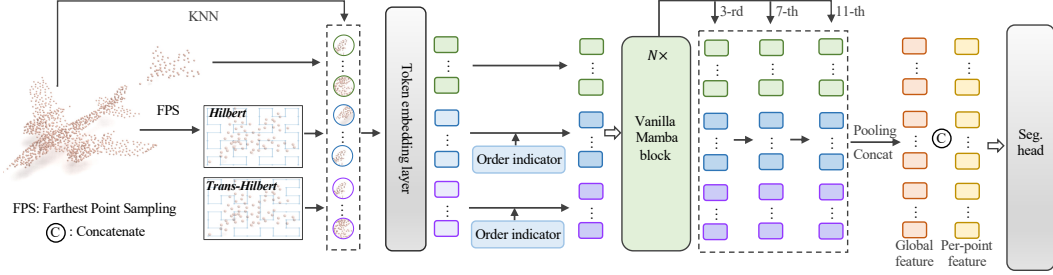
Figure 6: The details of our PointMamba for segmentation task.

backward SSM that operates on the feature channel. And it proposes a Local Norm Pooling block to extract local geometric features.

PointTramba [53] introduces a hybrid approach that integrates Transformers and Mamba. It segments point clouds into groups and utilizes Transformers to capture intra-group dependencies, while Mamba models inter-group relationships using a bi-directional, importance-aware ordering strategy.

As shown in Tab. 9, our PointMamba surpasses these concurrent methods by offering superior performance with reduced computational overhead. Note that our method is extremely simple and without complex design, which utilizes the vanilla Mamba block and abstains from incorporating modular designs from other baselines, thereby maintaining simplicity and efficiency in our approach. We believe such method can better illustrate the potential of SSM in point cloud analysis tasks.

Appendix C   More experimental resutls

C.1   Implement details

Table 10: Implementation details for pre-training and downstream tasks.

Configuration	Pre-training	Classification		Segmentation
	ShapeNetCore	ModelNet40	ScanObjectNN	ShapeNetPart
Optimizer	AdamW	AdamW	AdamW	AdamW
Learning rate	1e-3	3e-4	5e-4	2e-4
Weight decay	5e-2	5e-2	5e-2	5e-2
Learning rate scheduler	cosine	cosine	cosine	cosine
Training epochs	300	300	300	300
Warmup epochs	10	10	10	10
Batch size	128	32	32	16
Num. of encoder layers N	12	12	12	12
Num. of decoder layers	4	-	-	-
Input points M	1024	1024	2048	2048
Num. of patches n	64	64	128	128
Patch size k	32	32	32	32
Augmentation	Scale&Trans	Scale&Trans	Rotation	-

Pre-training Details. The ShapeNetCore dataset [5] is used for pre-training, including ∼51K clean 3D sample across 55 categories. The 1,024 input points are divided into 64 point patches, with each patch consisting of 32 points. The pre-training process includes 300 epochs, with a batch size of 128. More detail can be found in Tab. 10.

Downstream tasks Details. Fig 2 shows the pipeline of PointMamba for classification tasks. We report the overall accuracy without voting on the challenging ScanObjectNN [49] using 2,048 input points, and on ModelNet40 [57] using 1,024 input points. For segmentation on ShapeNetPart [58], as shown in Fig. 6, we use random, Hilbert, and Trans-Hilbert serializations, with order indicators applied on Hilbert/Trans-Hilbert serializations. Features from the 3-rd, 7-th, and last layer are pooled as global features after a simple feature fusion. These global features are then concatenated with per-point features and sent to the segmentation head. More detail can be found in Tab. 10.

17

Table 11: The effect of masking strategy. The pre-training loss (× 1000) along with fine-tuning accuracy (%) are reported.

Masking ratio	Loss	OBJ-BG	OBJ-ONLY
0.4	2.01	92.60	90.70
0.6	1.97	**94.32**	**92.60**
0.8	2.33	93.46	90.17
0.9	2.00	92.43	91.05

Table 12: The effect of classification token. Fine-tuning accuracy (%) are reported.

Methods	OBJ-BG	OBJ-ONLY
Before the sequence	93.63	91.05
After the sequence	**94.32**	90.19
Middle the sequence	93.98	90.71
Max Pool	93.39	91.36
Average Pool	**94.32**	**92.60**

| Input | Masking | Reconstruction | Input | Masking | Reconstruction |
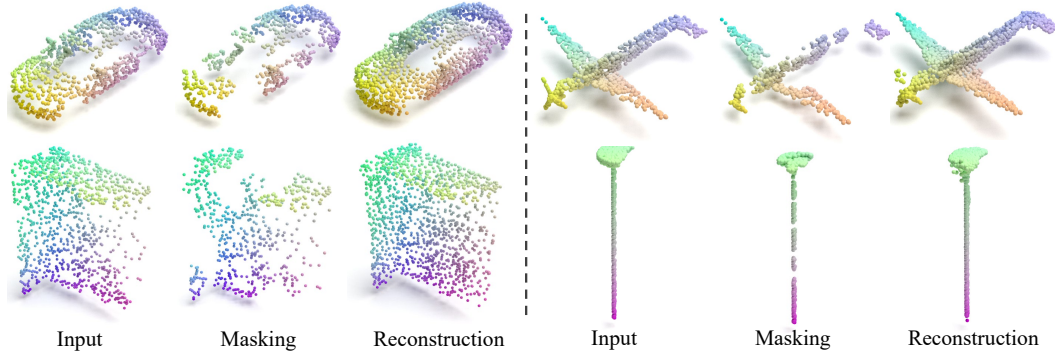
Figure 7: The qualitative results of mask predictions of our PointMamba on ShapeNet validation set.

C.2 Additional ablation study

In this section, we do additional ablation studies on several hyper-parameters.

Masking strategy for pre-training. By employing a serialization-based mask modeling paradigm, our PointMamba achieves superior performance. To find a proper masking strategy for our method, we compare two types of masking with varying ratios. The block masking [60] masks geometrically proximate point cloud patches, leading to a more challenging reconstruction target. In Tab. 11, we experimentally find that masking 60% of point patches by randomly choosing can achieve good performance.

Usage of classification token. Previous works [12, 60, 38] often use a classification token [CLS] as a global token for classification. As in Tab. 12, we find that without [CLS] and utilizing only the average pooling of the final block's output yields the best results for PointMamba.

Appendix D   Qualitative Analysis

D.1   Mask modeling visualization

As in Sec. 4.2, we customize a simple yet effective serialization-based mask modeling paradigm. By randomly masking about 60% of serialization-based point tokens, an asymmetric vanilla Mamba autoencoder is utilized to extract the point feature, with a simple prediction head for reconstruction. In Fig. 7, we present qualitative results of mask modeling on ShapeNet validation set. Despite a 60% masking ratio, our PointMamba effectively reconstructs the masked patches, providing a strong self-supervised knowledge for downstream tasks.

D.2   Part segmentation visualization

In this subsection, we present the qualitative results for part segmentation on the ShapeNetPart validation set, including both the ground truth and the predicted results. As in Fig. 8, our PointMamba shows highly competitive results on part segmentation.

18
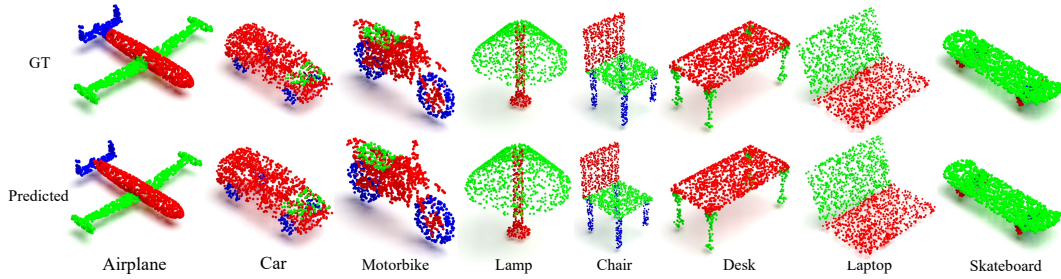
Figure 8: The qualitative results of part segmentation of our PointMamba on ShapeNetPart.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: See abstract and introduction.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: See limitation part.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.

19

- While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory Assumptions and Proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes]

   Justification: See experiments.

   Guidelines:

   - The answer NA means that the paper does not include theoretical results.
   - All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
   - All assumptions should be clearly stated or referenced in the statement of any theorems.
   - The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
   - Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
   - Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental Result Reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: See experiments part.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.

20

(c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).

(d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [NA]

Justification: The code will be made available.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental Setting/Details**

Question: Does the paper specify all the training and test details (e.g., data splits, hyper-parameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

Answer: [Yes]

Justification: See experiments part.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
- The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment Statistical Significance**

Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

Answer: [Yes]

Justification: See experiments part.

21

Guidelines:

- The answer NA means that the paper does not include experiments.
- The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments Compute Resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: See experiments part.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code Of Ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics https://neurips.cc/public/EthicsGuidelines?

   Answer: [Yes]

   Justification: We follow the NeurIPS Code of Ethic.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader Impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [NA]

22

Justification: no societal impacts.

Guidelines:

- The answer NA means that there is no societal impact of the work performed.
- If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

Answer: [NA]

Justification: No such risks.

Guidelines:

- The answer NA means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [Yes]

Justification: We will release the code.

Guidelines:

- The answer NA means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.

23

- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New Assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [NA]

    Justification: We use the public assets.

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and Research with Human Subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

    Justification: [NA]

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional Review Board (IRB) Approvals or Equivalent for Research with Human Subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

    Justification: [NA]

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.

24

- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

25