CloudMamba: Grouped Selective State Spaces for Point Cloud Analysis

Kanglin Qu[1], Pan Gao[1*], Qun Dai[1*], Zhanzhi Ye[1], Rui Ye[2], Yuanhao Sun[3]

[1]Nanjing University of Aeronautics and Astronautics
[2]Nanjing Agricultural University
[3]Beijing University of Posts and Telecommunications
klinqu@163.com, {Pan.Gao, daiqun, yezhanzhi}@nuaa.edu.cn, yerui@njau.edu.cn, sunyh@bupt.edu.cn

Abstract

Due to the long-range modeling ability and linear complexity property, Mamba has attracted considerable attention in point cloud analysis. Despite some interesting progress, related work still suffers from imperfect point cloud serialization, insufficient high-level geometric perception, and overfitting of the selective state space model (S6) at the core of Mamba. To this end, we resort to an SSM-based point cloud network termed CloudMamba to address the above challenges. Specifically, we propose sequence expanding and sequence merging, where the former serializes points along each axis separately and the latter serves to fuse the corresponding higher-order features causally inferred from different sequences, enabling unordered point sets to adapt more stably to the causal nature of Mamba without parameters. Meanwhile, we design chainedMamba that chains the forward and backward processes in the parallel bidirectional Mamba, capturing high-level geometric information during scanning. In addition, we propose a grouped selective state space model (GS6) via parameter sharing on S6, alleviating the overfitting problem caused by the computational mode in S6. Experiments on various point cloud tasks validate CloudMamba's ability to achieve state-of-the-art results with significantly less complexity.

Code — https://github.com/Point-Cloud-Learning/CloudMamba

Extended version — https://arxiv.org/abs/2511.07823

Introduction

The attention mechanism (Zhang et al. 2023, 2024a; Zhou et al. 2025) is gradually replacing the convolution as a dominant operator in point cloud analysis with its ability to achieve global modeling, but its quadratic complexity causes unbearable computational overheads. To handle this challenge, existing attention-based networks (Nie et al. 2022; Fan et al. 2022; Park et al. 2022; Liu et al. 2023) perform attention interactions in local neighborhoods or within windows, which, however, impose constraints on receptive fields. Recently, state space models (SSMs) (Gu et al. 2021; Gu, Goel, and Ré 2022; Mehta et al. 2022; Gupta, Gu, and Berant 2022) have shown great potential in natural language processing (NLP). Mamba (Gu and Dao 2023) achieves flexible selection of relevant information in a data-dependent manner by the selective state space model (S6), further enhancing long-range modeling capability. Moreover, Mamba with linear complexity adopts a hardware-aware algorithm inspired by FlashAttention (Dao et al. 2022), significantly improving training and inference efficiencies.

Serialization	Grid size	OA (%)
Hilbert + Trans-Hilbert	0.010	90.84
Hilbert + Trans-Hilbert	0.015	92.82
Hilbert + Trans-Hilbert	0.020	89.13
Sequence expanding & merging	/	**93.65**

Table 1: Experimental results of different serialization methods in our network on ModelNet40 dataset.

Given Mamba's success in NLP, some works (Han et al. 2024; Liang et al. 2024; Liu et al. 2024; Zhang et al. 2024b) attempt to transplant this success from language modeling to point cloud analysis. In this process, despite some significant progress, these efforts still struggle to achieve satisfactory results due to the following three aspects:

- **Point cloud serialization.** It is crucial to build inter-point structure dependencies in a point sequence by serialization for Mamba's causal inference, and existing strategies are mostly based on space-filling curves. However, the reliability of the structural dependencies built by space-filling curves is highly sensitive to the setting of the grid size, as shown in Tab. 1. Besides, as a pioneering study based on Mamba, PointMamba (Liang et al. 2024) concatenates the serialization results of the Hilbert curve and its variant Trans-Hilbert (Hilbert 1935). However, this practice has the following shortcomings: (1) a longer sequence induced by the concatenation introduces redundancy and negatively affects efficiency, and (2) concatenating serialized sequences with different spatial relationships tends to cause confusion. Hence, existing efforts are imperfect in point cloud serialization.
- **High-level geometric perception.** While Mamba achieves superior long-range modeling via the selection mechanism, it is only unidirectional and not applicable to visual data requiring global learning. Therefore, some

*Corresponding author.

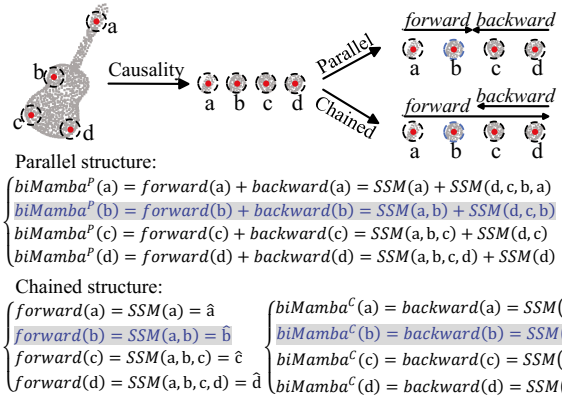


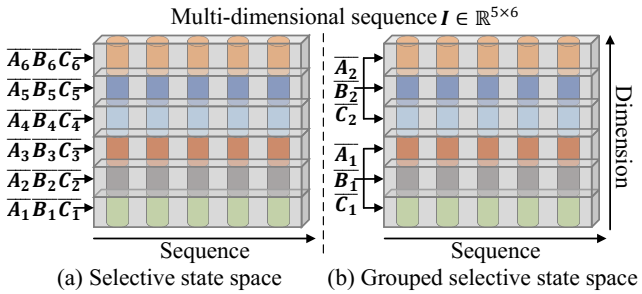
Figure 1: Inference process of the bidirectional Mamba with different structures, where the blue equations denote the inference processes of both bidirectional Mamba for the point *b*, respectively. In the backward inference of the chained structure, the previous points perceive the high-level structural semantics that are inferred from the forward Mamba.

works (Liu et al. 2024; Zhang et al. 2024b) introduce the bidirectional Mamba with a parallel structure from Vision Mamba (Zhu et al. 2024). However, this structure limits the expressivity of networks. As illustrated in Fig. 1, the parallel bidirectional Mamba provides each point with a global receptive field through the forward and backward inference, *e.g.*, the point b can interact with a, c, and d. Nevertheless, we argue that this type of inference on primitive low-order geometric features can result in insufficient high-level geometric perception.

- **Computational mode in S6.** Existing Mamba-based point cloud works (Liang et al. 2024; Liu et al. 2024; Zhang et al. 2024b) take S6 as the core component, where each dimension is learned by a separate set of parameters when dealing with multi-dimensional sequences, as shown in Fig. 2(a). This computational mode results in overfitting in causal inference due to excessive parameters in each dimension.

According to the above analyses, we propose a novel SSM-based point cloud network termed CloudMamba, which obeys the following designs to address the above challenges encountered in related works in order to greatly promote the development of Mamba in the point cloud domain:

- **Sequence expanding & merging.** We propose sequence expanding, which serializes points along each axis separately, and sequence merging, which fuses the corresponding higher-order features causally inferred from different sequences. The sequence expanding directly builds structural dependencies in multiple perspectives from different axes, which not only overcomes the parameter sensitivity in space-filling curves, but also avoids the inefficiency and causal confusion caused by concatenating different serialization results. Furthermore, it captures rich geometric information by integrating with the sequence merging. Together, these enable unordered point sets to adapt more stably to the causal nature of Mamba without parameters, as shown in Tab. 1.



(a) Selective state space (b) Grouped selective state space

Figure 2: Computational modes of S6 and GS6, where GS6's grouping rate is 3. A multi-dimensional sequence $\boldsymbol{I} \in \mathbb{R}^{5\times6}$ is used as an example, with the subscript denoting the parameter to be used for a dimension or a set of dimensions.

- **ChainedMamba.** We propose a chained bidirectional Mamba termed chainedMamba, which chains the forward and backward processes in the parallel bidirectional Mamba. In this case, due to the chaining property, the point b could interact with the high-level structural semantics of d, c, b, *i.e.*, $\hat{d}$, $\hat{c}$, and $\hat{b}$ as shown in Fig. 1. By utilizing these high-level structural semantics inferred from low-order geometric features in the forward Mamba during backward inference, it can achieve superior high-level geometric perception, as demonstrated in Tab. 7.
- **GS6.** We propose a grouped selective state space model (GS6) by sharing a same set of parameters across several dimensions, which alleviates the overfitting caused by the computational mode in S6, as illustrated in Fig. 2(b).

Finally, we conduct experiments on point cloud recognition ModelNet40 (Wu et al. 2015) and ScanObjectNN (Uy et al. 2019), part segmentation ShapeNet (Yi et al. 2016), as well as semantic segmentation S3DIS (Armeni et al. 2016). Experimental comparisons show CloudMamba is able to obtain state-of-the-art accuracy with linear complexity.

In summary, the contributions of this paper are fourfold:
(1) A novel SSM-based point cloud network, CloudMamba, is constructed upon Mamba, which achieves state-of-the-art results in various point cloud tasks with linear complexity.
(2) A serialization method consisting of sequence expanding and sequence merging is proposed, which builds structural dependencies directly from the coordinate axes defining spatial locations and captures rich geometric information by merging the sequences generated along different axes, making unordered point sets better adapted to the causal nature of Mamba without parameters.
(3) A chained bidirectional Mamba termed chainedMamba is designed by chaining the forward and backward processes in the parallel bidirectional Mamba. This simple yet effective refinement achieves excellent high-level geometric perception.
(4) A variant GS6 on S6 is proposed by adopting the parameter sharing, which mitigates overfitting in S6. To our best knowledge, all existing Mamba-based methods are based on

S6, and GS6 is the first exploration of optimizing the computational mode in S6.

Related Work

SSMs and Mamba

SSMs are used in control theory to describe dynamic systems, which have recently been introduced to deep learning. Initially, Gu *et al.* (Gu et al. 2021) designed a linear state space layer (LSSL) based on linear continuous-time state space representations and demonstrated its potential to deal with long-range dependencies in sequential data by combining it with the HiPPO initialization (Gu et al. 2020). However, LSSL requires excessive computational resources and cannot be used as a generic sequential modeling method. Hence, the structured state space sequence model (S4) (Gu, Goel, and Ré 2022) normalizes the parameters into a diagonal structure to solve the computational bottleneck. Subsequently, many efforts have been made to improve SSMs (Mehta et al. 2022; Smith, Warrington, and Linderman 2022; Gupta, Gu, and Berant 2022), where S6 is a landmark work. To enable flexible integration into neural networks, S6 is combined with the simplified H3 architecture (Gu and Dao 2023) to build Mamba, which achieves impressive results in NLP. Later, several works extend Mamba to other research directions, including image recognition (Liu et al. 2025), medical image segmentation (Ruan and Xiang 2024), and graph sequence modeling (Wang et al. 2024a).

Recently, Mamba has also begun to be applied to point cloud analysis. PointMamba (Liang et al. 2024) and Oct-Mamba (Liu et al. 2024) employ space-filling curves to form causal sequences for Mamba. Different from them, our network directly adopts coordinate axes to build structural dependencies in multiple perspectives without parameters, overcoming the parameter sensitivity in space-filling curves. PCM (Zhang et al. 2024b) employs the parallel bidirectional Mamba, resulting in insufficient high-level geometric perception. In our work, chainedMamba achieves excellent high-level geometric perception by chaining the forward and backward processes in the parallel bidirectional Mamba. Finally, all the previous works only apply directly Mamba to point cloud analysis, while we improve its S6 to mitigate potential overfitting.

Preliminaries

SSMs

SSMs are cyclic processes with latent states, which map a 1-D equation or sequence $x(t) \in \mathbb{R}^N$ to $y(t) \in \mathbb{R}^N$ by a latent state $h(t) \in \mathbb{R}^N$. The process is mathematically denoted as a linear ordinary differential equation as follows

$$y(t) = \boldsymbol{C}h(t), h'(t) = \boldsymbol{A}h(t) + \boldsymbol{B}x(t), \quad (1)$$

where the three parameters $\boldsymbol{A} \in \mathbb{R}^{N\times N}$, $\boldsymbol{B} \in \mathbb{R}^N$, and $\boldsymbol{C} \in \mathbb{R}^N$ represent the state matrix, input matrix, and output matrix, respectively. Since the above SSMs run on continuous inputs and are not applicable to discrete inputs such as images and text, they cannot be introduced into deep models. Thus, it is necessary to discretize them, and the zero-order hold is commonly used as a discretization method (see Appendix A). The discretized formulas are as follows

$$y_t = \bar{\boldsymbol{C}}h_t, \quad h_t = \bar{\boldsymbol{A}}h_{t-1} + \bar{\boldsymbol{B}}x_t, \quad (2)$$

where $\bar{\boldsymbol{A}}$ and $\bar{\boldsymbol{B}}$ are the results of discretizing the continuous parameters $\boldsymbol{A}$ and $\boldsymbol{B}$ by a time scale $\boldsymbol{\Delta}$, denoted as

$$\bar{\boldsymbol{A}} = e^{\boldsymbol{\Delta A}}, \ \bar{\boldsymbol{C}} = \boldsymbol{C}, \quad \bar{\boldsymbol{B}} = (\boldsymbol{\Delta A})^{-1}\Big(e^{\boldsymbol{\Delta A}} - \boldsymbol{I}\Big)(\boldsymbol{\Delta B}). \quad (3)$$

Mamba

Since processing the input and latent state equally, previous approaches focusing on linear time-invariant SSMs (where $\bar{\boldsymbol{A}}$ and $\bar{\boldsymbol{B}}$ are invariant) may fail to capture critical information from context. Therefore, Mamba proposes a novel SSM (termed S6) by integrating an input-dependent selective mechanism into SSMs, where $\bar{\boldsymbol{A}}$ and $\bar{\boldsymbol{B}}$ are the functions of inputs, indicating Mamba is linear time-variant.

However, Mamba's linear time-variant parameters result in dynamic weights, preventing it from being computed as efficiently as linear time-invariant SSMs using the convolution. Thus, Mamba uses a parallel scanning algorithm with linear complexity (Smith, Warrington, and Linderman 2022) to maintain efficient computation. In addition, it designs a hardware-aware algorithm by utilizing the fast SRAM in GPUs to improve efficiency.

Methodology

Overview

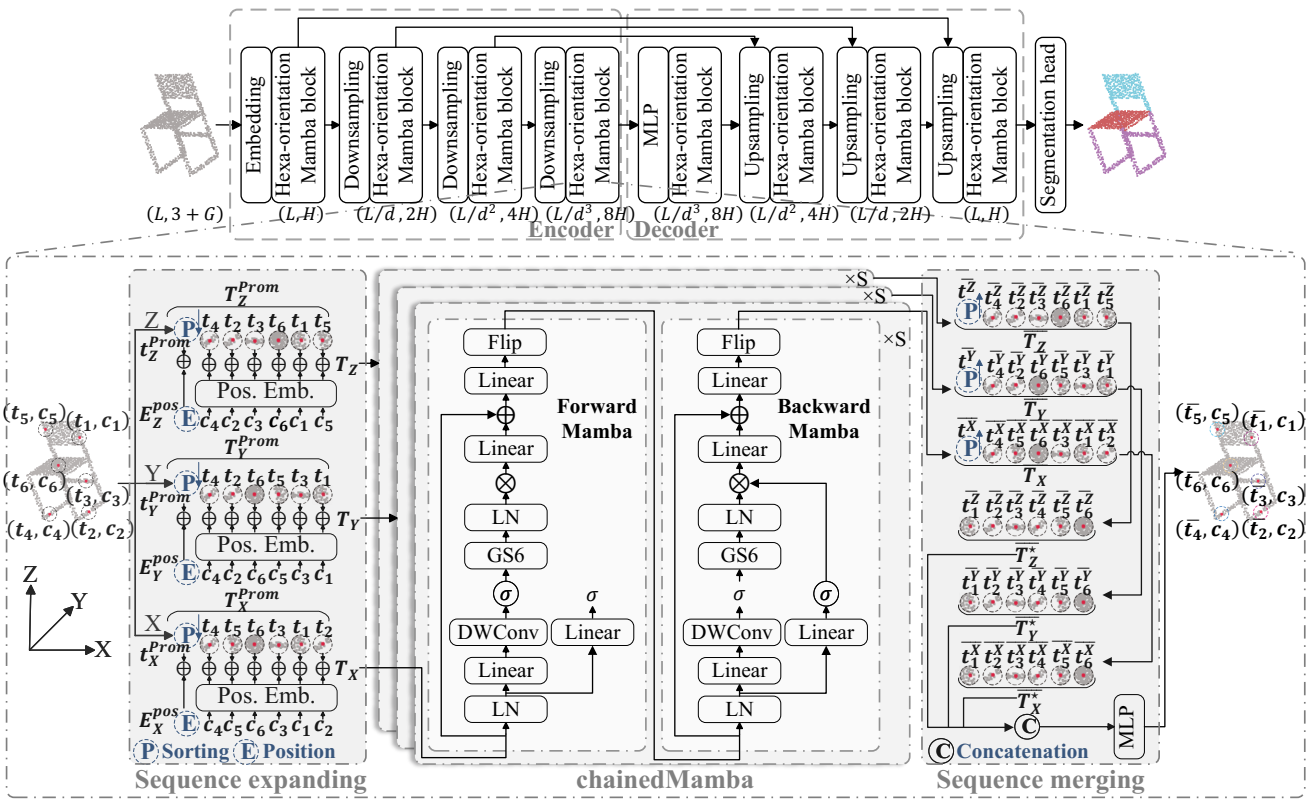
Figure 3 presents the overview of our network, which takes as input a point set $\boldsymbol{P} = \{\boldsymbol{p_i} = (\boldsymbol{c_i}, \ \boldsymbol{f_i})\}_{i=1}^{L}$, where $\boldsymbol{c_i} \in \mathbb{R}^3$ and $\boldsymbol{f_i} \in \mathbb{R}^G$ denote the 3D coordinates of $\boldsymbol{p_i}$ and other relevant features, respectively. In the initial stage, $\boldsymbol{P}$ is transformed into a high-dimensional space with H dimensions through an embedding layer consisting of a multi-layer perceptron machine (MLP). Then, an encoder-decoder architecture built on up- and down-sampling layers and hexa-orientation Mamba blocks is used for hierarchical feature aggregation. Finally, a corresponding task head is applied. In this paper, our network is validated by recognition and segmentation results on point clouds. The recognition head processes the output of the encoder through an average pooling layer and an MLP, and the segmentation head employs an MLP to process the output of the decoder. Next, we describe components in the encoder-decoder architecture in detail.

Hexa-orientation Mamba Block

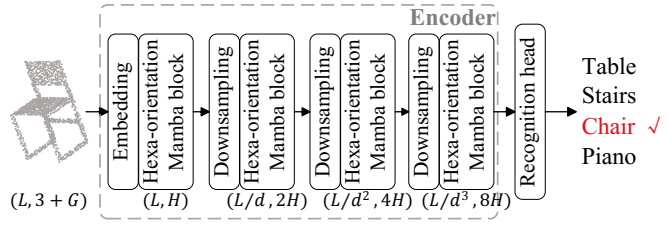
The hexa-orientation Mamba block is the main feature aggregation module of our network, capable of facilitating geometry perception from multiple perspectives, and is used for features at each level to capture fine-grained structures, specifically consisting of sequence expanding, chainedMamba, and sequence merging.

Sequence Expanding

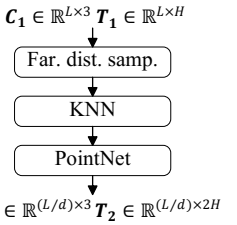
Mamba tailored for causal sequences requires causal dependencies between elements, but visual data has a non-causal nature, such that applying Mamba directly to point



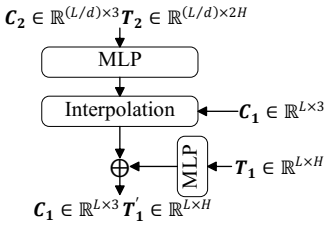
(a) CloudMamba for point cloud segmentation



(b) CloudMamba for point cloud recognition



(c) Downsampling



(d) Upsampling

Figure 3: Pipeline of our proposed network. The Flip layer in the chainedMamba indicates a flip operation on the sequence for global modelling. S6 in Mamba is replaced by GS6. Since each point sequence is processed with forward and backward directions, there are hexa orientations for causal modeling.

clouds does not achieve expected results (see Tab. 6). Existing strategies mostly build structural dependencies based on the spatial proximity of space-filling curves, yet their reliability is highly sensitive to the setting of the grid size. Hence, we build structural dependencies directly from the coordinate axis defining spatial locations. Since structural dependencies on only one axis cannot reflect rich geometric information, we construct causal sequences based on the ascending order of point coordinates $\boldsymbol{C} = \{\boldsymbol{c_1},\ \boldsymbol{c_2},\ \cdots,\ \boldsymbol{c_j}\}$ corresponding to input point features $\boldsymbol{T} = \{\boldsymbol{t_1},\ \boldsymbol{t_2},\ \cdots,\ \boldsymbol{t_j}\}$ along the Z, Y, and X axes, respectively

$$\boldsymbol{T}_{\boldsymbol{A}}^{\boldsymbol{Cau}} = \left\{\boldsymbol{t}_{\boldsymbol{A}_1},\ \boldsymbol{t}_{\boldsymbol{A}_2},\ \cdots,\ \boldsymbol{t}_{\boldsymbol{A}_j}\right\}\ A \in \{\mathrm{Z},\ \mathrm{Y},\ \mathrm{X}\}, \quad (4)$$

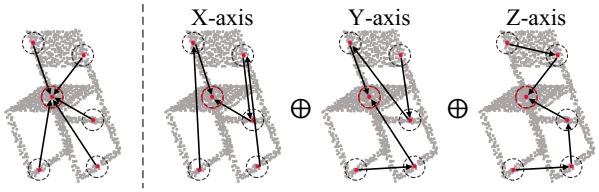
where $\boldsymbol{T}_{\boldsymbol{A}}^{\boldsymbol{Cau}}$ denotes a causal sequence constructed on the A-axis, $\boldsymbol{t}_{\boldsymbol{A}_j}$ represents the j-th point feature sorted on the A-axis, and A_j corresponds to a subscript in $\boldsymbol{T}$.

Prompt. Since the network involves the structural dependencies arising from three different directions (these causal sequences are essentially shared, differing only in sorting), a learnable sorting prompt $\boldsymbol{t}_{\boldsymbol{A}}^{\boldsymbol{Prom}}$ is added for each sequence to avoid confusing the network about these dependencies

$$\boldsymbol{T}_{\boldsymbol{A}}^{\boldsymbol{Prom}} = \left\{\boldsymbol{t}_{\boldsymbol{A}}^{\boldsymbol{Prom}},\ \boldsymbol{t}_{\boldsymbol{A}_1},\ \boldsymbol{t}_{\boldsymbol{A}_2},\ \cdots,\ \boldsymbol{t}_{\boldsymbol{A}_j}\right\}, \quad (5)$$

where $\boldsymbol{t}_{\boldsymbol{A}}^{\boldsymbol{Prom}}$ is a sorting prompt added for the A axis.

Position-aware. Visual data is location-sensitive (Zhao et al. 2021; Xu et al. 2024; Liang et al. 2025b). While point coordinates contain position information, fine-grained position information may be lost in higher-level features as the network deepens. Additionally, unlike the convolution, SSMs cannot implicitly endow the network with the spatial inductive bias. Thus, we explicitly provide position information to corresponding point features by position embeddings on



(a) Attention (b) Hexa-orientation Mamba block

Figure 4: Illustration of the global receptive fields of the attention mechanism and hexa-orientation Mamba block, with the point in the red box as an example.

point coordinates as follows

$$\boldsymbol{T_A} = \left\{ \boldsymbol{t_A^{Prom}} + \boldsymbol{E_A^{pos}}, \boldsymbol{t_{A_1}} + \boldsymbol{\rho}\left(\boldsymbol{c_{A_1}}\right), \cdots, \boldsymbol{t_{A_j}} + \boldsymbol{\rho}\left(\boldsymbol{c_{A_j}}\right) \right\}, \quad (6)$$

where $\boldsymbol{E_A^{pos}}$ is a learnable position embedding we add for each sorting prompt, $\boldsymbol{c_{A_j}}$ is a point coordinate corresponding to the point feature $\boldsymbol{t_{A_j}}$, and ρ is an MLP used to map point coordinates to position embeddings.

ChainedMamba

Although the mainstream parallel bidirectional Mamba overcomes the limitation of Mamba for visual data requiring global learning, its inference on primitive low-order geometric features leads to insufficient high-level geometric perception. Therefore, we propose chainedMamba by chaining the forward and backward processes in the parallel bidirectional Mamba, ingeniously realizing superior high-level geometric perception, as shown in Fig. 3. Besides, by combining the causal sequences formed by the sequence expanding, Fig. 4 demonstrates that the hexa-orientation Mamba block is able to facilitate geometric learning from multiple perspectives with a global receptive field, in which the global receptive field is obtained through compressing the historical hidden state in both forward and backward directions, unlike the attention mechanism whereas the global receptive field is obtained through interactions with all elements. Although the way to obtain both global receptive fields is different, there is a close correlation between Mamba and self-attention, *i.e.*, S6 is a special causal self-attention, see Appendix B.

GS6. As shown in Eq. 1, SSMs are sequence mappings from $\mathbb{R}^N \rightarrow \mathbb{R}^N$ on a 1-D sequence, thus S6 uses a separate set of parameters for each dimension when the input sequence has multiple dimensions, which leads to overfitting in causal inference due to excessive parameters in each dimension (see Tab. 5). To improve the generalizability of the network, we propose a grouped selective state space model (GS6) through parameter sharing on S6, as shown in Algorithm 1, where Linear_N denotes a linear layer projected to dimension N and $\mathrm{Repeat}\left(\boldsymbol{V},\ g\right)$ denotes repeating an element in tensor $\boldsymbol{V}$ g times. GS6 uses a grouping ratio g to make every g dimensions in $\boldsymbol{x}$ using the same set of parameters, alleviating the overfitting caused by S6's computational mode, while the repeating function ensures the least modification on S6's coding.

Sequence Merging

After the chainedMamba process different causal sequences $\{\boldsymbol{T_A},\ A \in \{\mathrm{Z},\ \mathrm{Y},\ \mathrm{X}\}\}$, these sequences

Algorithm 1. S6 + Grouping (GS6)

Input: $\boldsymbol{x}$: $(B,\ L,\ D)$ and a grouping rate g
Output: $\boldsymbol{y}$: $(B,\ L,\ D)$
1. $\boldsymbol{B}$: $(B,\ L,\ N) \leftarrow \mathrm{Linear}_N\left(\boldsymbol{x}\right)$
2. $\boldsymbol{C}$: $(B,\ L,\ N) \leftarrow \mathrm{Linear}_N\left(\boldsymbol{x}\right)$
/* shape of $\mathbf{Parameter_{\Delta}}$ is (D/g), and the softplus ensures positive $\boldsymbol{\Delta}$ */
3. $\boldsymbol{\Delta}$: $(B, L, D/g) \leftarrow \log\left(1 + e^{\left(\mathrm{Linear}_{D/g}(\boldsymbol{x}) + \mathbf{Parameter_{\Delta}}\right)}\right)$
/* shape of $\mathbf{Parameter_A}$ is $(D/g, N)$. Each dimension in $\boldsymbol{A}$ represents a structured diagonal $N \times N$ matrix */
4. $\bar{\boldsymbol{A}}$: $(B, L, D/g, N) \leftarrow \boldsymbol{\Delta} \otimes \mathbf{Parameter_A}$
5. $\bar{\boldsymbol{B}}$: $(B, L, D/g, N) \leftarrow \boldsymbol{\Delta} \otimes \boldsymbol{B}$
6. $\bar{\boldsymbol{A}},\ \bar{\boldsymbol{B}}$: $(B, L, D, N) \leftarrow \mathrm{Repeat}(\bar{\boldsymbol{A}},\ g), \mathrm{Repeat}(\bar{\boldsymbol{B}},\ g)$
/* time-variant: calculated by the parallel scanning */
7. $\boldsymbol{y}$: $(B, L, D) \leftarrow \mathrm{SSM}\left(\bar{\boldsymbol{A}},\ \bar{\boldsymbol{B}},\ \boldsymbol{C}\right)\left(\boldsymbol{x}\right)$
return $\boldsymbol{y}$

$\left(\left\{\overline{\boldsymbol{T_A}} = \left\{\overline{\boldsymbol{t^A}},\ \overline{\boldsymbol{t^A_{A_1}}},\ \cdots,\ \overline{\boldsymbol{t^A_{A_j}}}\right\}\right\}\right)$ contain the higher-order interaction features causally inferred from different perspectives, respectively, and the sequence merging is responsible for integrating them together to captures rich geometric information. Specifically, the sorting prompts (the first term) in the sequences are first removed and the sequences are returned to their original order $\left(\left\{\overline{\boldsymbol{T^\star_A}} = \left\{\overline{\boldsymbol{t^A_1}},\ \overline{\boldsymbol{t^A_2}},\ \cdots,\ \overline{\boldsymbol{t^A_j}}\right\}\right\}\right)$, and then the sequences are merged by Eq. 7, where Concat denotes channel-wise concatenation on the corresponding element and γ is an MLP scaled down by three times, keeping the output dimension of the hexa-orientation Mamba block the same as the input dimension.

$$\boldsymbol{Output} = \gamma\left(\mathrm{Concat}\left(\left\{\overline{\boldsymbol{T^\star_A}},\ A \in \{\mathrm{Z},\ \mathrm{Y},\ \mathrm{X}\}\right\}\right)\right) \quad (7)$$

Downsampling and Upsampling Layers

Downsampling. The downsampling layer is to reduce the cardinality of point sets at a downsampling rate in the encoder. Assume that the input point coordinate set and its corresponding point feature set of the downsampling layer are $\boldsymbol{C_1} \in \mathbb{R}^{L\times 3}$ and $\boldsymbol{T_1} \in \mathbb{R}^{L\times H}$, respectively, and the downsampling rate is d. First, the farthest distance sampling (Qi et al. 2017b) is performed in $\boldsymbol{C_1}$ to determine a point coordinate subset $\boldsymbol{C_2} \in \mathbb{R}^{(L/d)\times 3}$ with L/d sampled points. Then, to pool the point features on $\boldsymbol{C_1}$ to $\boldsymbol{C_2}$, we perform the K-nearest neighbor on $\boldsymbol{C_1}$ for each point coordinate in $\boldsymbol{C_2}$ and pass the neighboring point features of each point in $\boldsymbol{C_2}$ through PointNet (Qi et al. 2017a) to construct a point feature set $\boldsymbol{T_2} \in \mathbb{R}^{(L/d)\times 2H}$ corresponding to $\boldsymbol{C_2}$. The overview of the downsampling layer is shown in Fig. 3(c).

Upsampling. For dense tasks such as point cloud segmentation, we follow the U-Net design (Ronneberger, Fischer, and Brox 2015), where the above encoder is coupled with a symmetric decoder. The upsampling layer serves as a connection between successive stages in the decoder, and it is to recover the cardinality of point sets according to the corresponding encoder stage. Suppose that the input point coordinate set and its corresponding point feature set of the upsampling layer are $\boldsymbol{C_2} \in \mathbb{R}^{(L/d)\times 3}$ and $\boldsymbol{T_2} \in \mathbb{R}^{(L/d)\times 2H}$, respectively, and the point coordinate set and its point feature set

of the corresponding encoder stage are $\boldsymbol{C_1} \supset \boldsymbol{C_2}$ ($\boldsymbol{C_1} \in \mathbb{R}^{L\times 3}$) and $\boldsymbol{T_1} \in \mathbb{R}^{L\times H}$, respectively. First, the dimension of $\boldsymbol{T_2}$ is aligned to $\boldsymbol{T_1}$ by an MLP (*i.e.*, $\boldsymbol{T_2} \in \mathbb{R}^{(L/d)\times H}$), and the tri-linear interpolation method is performed on $\boldsymbol{C_2}$ for each point in $\boldsymbol{C_1}$ to obtain the interpolated point features corresponding to $\boldsymbol{C_1}$ from $\boldsymbol{T_2}$. Then, $\boldsymbol{T_1}$ passing through an MLP is summed with these interpolated point features by the skip connection to obtain a point feature set $\boldsymbol{T_1'} \in \mathbb{R}^{L\times H}$ corresponding to $\boldsymbol{C_1}$ at the decoder stage. Fig. 3(d) illustrates the structure of the upsampling layer.

Experiment

To validate CloudMamba's potential in point cloud analysis, it is compared with different types of networks on ModelNet40, ScanObjectNN, ShapeNet, and S3DIS datasets. For a detailed introduction on datasets and evaluation metrics, see Appendix C. Moreover, we explore the network property and the impact of the components on the performance by extensive ablation comparisons. Also, the FLOPs (floating point operations) and Params reported, reflecting computation and space complexity, respectively, are measured on the same RTX 4090 GPU to ensure a fair comparison.

Point Cloud Recognition

Table 2 compares the experimental results of our network and different operators-based mainstream works on ModelNet40 dataset. As shown in the fifth column in Tab. 2, CloudMamba achieves a higher OA compared to the concurrent SSM-based works, which indicates that our network is able to take full advantage of Mamba's superior long-range modeling. Besides, CloudMamba achieves competitive accuracy with the state-of-the-art Point Transformer but consuming less computational resources, which not only stresses the strong geometric representation capability of our network, but also proves it improves computational efficiency based on the linear complexity of Mamba. To further validate the robustness of the network to complex scenarios, the sixth column in Tab. 2 lists the experimental results of our network and different operators-based works on challenging ScanObjectNN (PB_T50_RS) dataset, where CloudMamba outperforms the state-of-the-art PointConT and PCM, which strongly proves its robustness.

Point Cloud Part Segmentation

Table 3 compares the experimental results of our network with different operators-based mainstream works on ShapeNet dataset. As shown in Tab. 3, our network achieves a higher Instance mIoU, and significantly improves the accuracy compared to the concurrent SSM-based works. It is worth noting that CloudMamba achieves the same Instance mIoU as the state-of-the-art Point Transformer with fewer FLOPs and Params, which fully validates our network is capable of adapting Mamba's superior long-range modeling to point cloud analysis. Also, we observe CloudMamba requires only 1.234G FLOPs, significantly lower than the attention networks and other SSM networks, which demonstrates our network enables a lightweight architecture.

Networks	Operators	FLOPs	Params	OA	
				Model.	Scan.
PointNet++ (2017b)	MLP	4.1	1.7	91.9	77.9
PointMLP (2022)	MLP	15.7	13.2	93.6	85.4
PointNext (2022)	MLP	1.6	1.4	93.2	87.7
DGCNN (2019)	CNN	-	-	92.9	78.1
3D-GCN (2022)	CNN	-	-	92.1	-
Point Transformer (2021)	Attention	18.4	9.6	**93.7**	-
Point-BERT‡ (2022)	Attention	2.3	22.1	92.7	83.1
Point-MAE‡ (2022)	Attention	2.4	22.1	93.2	85.2
OctFormer (2023)	Attention	0.6	3.4	92.7	-
IDPT‡† (2023)	Attention	7.1	1.7	92.6	83.7
PointGST‡† (2025a)	Attention	4.8	0.6	93.4	85.6
DAPT‡† (2024)	Attention	5.0	1.1	93.1	85.4
LCM‡ (2024)	Attention	1.3	2.7	93.6	87.8
PointTramba (2024b)	SSM & Attention	-	-	92.7	-
Mamba3D‡ (2024)	SSM	3.9	16.9	93.4	87.6
PointMamba‡ (2024)	SSM	3.6	12.3	93.6	**89.3**
OctMamba (2024)	SSM	1.3	3.1	92.7	-
SI-Mamba (2025)	SSM	3.6	12.3	92.7	87.3
CloudMamba	SSM	1.150	9.95	**93.7**	88.3

Table 2: Experimental results of our network and different operators-based mainstream works on ModelNet40 and ScanObjectNN datasets. ‡: Pre-training strategy. †: Point-BERT as a baseline.

Point Cloud Semantic Segmentation

Table 4 compares the experimental results of our network and different operators-based mainstream works on S3DIS dataset. As shown in Tab. 4, fewer SSM-based networks are validated on difficult scene-level S3DIS dataset, while CloudMamba makes up for this deficiency, and further demonstrates our network's exceptional long-range modeling capability and efficient linear complexity property by outperforming the state-of-the-art Point Transformer V3 with fewer FLOPs and params.

Ablation Studies

Validity Checking

In the proposed network, we design the sorting prompt, position embedding, and GS6 to further enhance the network performance. To demonstrate their validity, we compare the experimental results with and without these components, where the sorting prompt and position embedding are directly removed from the network when they are not used, and S6 is used as a replacement when GS6 is not used (*i.e.*, the grouping rate is set to 1 in GS6).

Sorting prompt. As shown in Group I in Tab. 5, the sorting prompt improves 0.46% OA, indicating that as the network continuously learns, the causal dependencies generated from different directions are confused, leading to incorrect shape understanding, and the sorting prompt, as a widget, can help the network clearly infer geometric structures from different dependencies with little effect on the FLOPs and Params.

Position embedding. Group II in Tab. 5 shows that the absence of the position embedding decreases 0.5% OA, which indicates that as the network deepens, higher-level features

Networks	Operator	FLOPs	Params	Ins. mIoU
PointNet++ (2017b)	MLP	4.8	1.7	85.1
PointMLP (2022)	MLP	6.3	16.8	86.1
ReCon‡ (2023)	MLP	-	-	86.4
DGCNN (2019)	CNN	-	-	85.2
3D-GCN (2022)	CNN	-	-	85.3
Point Transformer (2021)	Attention	36.7	19.4	**86.6**
PCT (2021)	Attention	4.4	2.9	86.4
Point-MAE‡ (2022)	Attention	4.8	22.1	86.1
MaskPoint‡ (2022)	Attention	4.8	22.1	86.0
IDPT‡† (2023)	Attention	4.8	5.7	85.3
LCM‡ (2024)	Attention	-	-	86.3
PointGST‡† (2025a)	Attention	4.8	5.6	85.7
DAPT‡† (2024)	Attention	5.0	5.7	85.5
Mamba3D‡ (2024)	SSM	11.8	23.0	85.6
PointMamba‡ (2024)	SSM	14.3	17.4	86.2
PCM (2024b)	SSM	52.1	40.6	84.3
PMA‡† (2025)	SSM	-	-	86.1
CloudMamba	SSM	1.234	16.57	**86.6**

Table 3: Experimental results of our network and different operators-based mainstream works on ShapeNet dataset. ‡: Pre-training strategy. †: Point-BERT as a baseline.

Networks	Operator	FLOPs	Params	mIoU
RandLA-Net (2022)	MLP	-	-	63.2
PointMeta (2023)	MLP	-	-	69.5
DGCNN (2019)	CNN	-	-	56.5
3D-GCN (2022)	CNN	-	-	58.6
Point Transformer (2021)	Attention	147.2	19.4	70.4
Point Transformer v2 (2022)	Attention	88.3	12.9	71.6
SuperpointTransformer (2023)	Attention	-	-	68.9
Point Transformer v3 (2024)	Attention	26.5	46.2	73.4
Grid Mamba (2025)	SSM	-	-	71.8
Pamba (2025)	SSM	-	-	73.5
PCM (2024b)	SSM	72.1	40.6	63.4
CloudMamba	SSM	4.922	16.56	**73.6**

Table 4: Experimental results of our network and different operators-based mainstream works on S3DIS dataset.

lose fine position information. Hence, it is necessary to facilitate geometric learning by continuously maintaining position information through the position embedding.

GS6. Group III in Tab. 5 shows GS6 improves 1.54% OA while reducing 0.4M params, revealing the computational mode in S6 causes a certain degree of overfitting, and GS6 is able to alleviate this problem via the parameter sharing.

Network Property

Causal dependency. Our network forms three causal sequences, providing multiple perspectives for geometric perception. To explore the impact of these sequences, we perform separate experiments in combinations of the sequences from Z, Y, and X axes, as shown in Tab. 6, where more causal sequences involved are able to obtain a higher OA, which reveals more causal dependencies from multiple perspectives can help the network to learn stereoscopic geometry. Notably, no causal sorting of point sets (input directly) achieves a low OA, which reflects the causal nature of Mamba.

	FLOPs (G)	Params (M)	OA (%)
I		Sorting prompt	
w/	1.150	9.95	**93.65**
w/o	1.150	9.95	93.19
II		Position embedding	
w/	1.150	9.95	**93.65**
w/o	1.141	9.70	93.15
III		GS6	
w/	1.150	9.95	**93.65**
w/o	1.150	10.35	92.11

Table 5: Experimental results of our network with and without the relevant components on ModelNet40 dataset.

Combination	FLOPs (G)	Params (M)	OA (%)
None	1.123	4.99	88.98
X	1.123	4.99	91.37
Y	1.123	4.99	90.96
Z	1.123	4.99	91.09
X, Y	1.136	7.47	92.38
X, Z	1.136	7.47	92.71
Y, Z	1.136	7.47	92.59
X, Y, Z	1.150	9.95	**93.65**

Table 6: Experimental results in combinations of the sequences from Z, Y, and X axes on ModelNet40 dataset, where None denotes point sets are not causally sorted.

Structure	FLOPs (G)	Params (M)	OA (%)
Parallel	1.150	9.95	92.69
Chained	1.150	9.95	**93.65**

Table 7: Experimental results with different structures of the bidirectional Mamba on ModelNet40 dataset.

Structure of the bidirectional Mamba. Existing Mamba-based visual networks mostly follow the idea of Zhu *et al.* (Zhu et al. 2024) to adapt Mamba's unidirectional modeling to visual data, *i.e.*, the parallel bidirectional Mamba, but it results in insufficient high-level geometric perception compared to our chainedMamba. We compare the experimental results of this parallel structure with the chained structure adopted by our network. For a fair comparison, the experimental configurations are identical (including the number of the bidirectional Mamba, and the use of GS6) except for employing different structures of the bidirectional Mamba. Tab. 7 shows our chained structure obtains a higher OA, which means our structure is able to achieve more superior geometric perception and yield better expressivity.

Conclusion

In this paper, we propose an SSM-based point cloud network termed CloudMamba, which overcomes the shortcomings of existing related works, including imperfect point cloud serialization, insufficient high-level geometric perception, and overfitting caused by the computational mode in S6, through the sequence expanding & merging, chainedMamba, and GS6. Extensive experiments demonstrate CloudMamba is able to obtain state-of-the-art results with linear complexity.

Acknowledgments

This work is supported in part by the National Natural Science Foundation of China under Grant (62476126, 62272227).

References

Armeni, I.; Sener, O.; Zamir, A. R.; Jiang, H. L.; Brilakis, I.; Fischer, M.; and Savarese, S. 2016. 3D semantic parsing of large-scale indoor spaces. In *CVPR*, 1534–1543. Las Vegas, NV, USA.

Bahri, A.; Yazdanpanah, M.; Noori, M.; Dastani, S.; Cheraghalikhani, M.; Osowiechi, D.; Hakim, G.; Beizaee, F.; Ayed, I. B.; and Desrosiers, C. 2025. Spectral Informed Mamba for Robust Point Cloud Processing. In *CVPR*, 11799–11809. Nashville, TN, USA.

Dao, T.; Fu, D. Y.; Ermon, S.; Rudra, A.; and Ré, C. 2022. FlashAttention: Fast and Memory-Efficient Exact Attention with IO-Awareness. In *NIPS*, 1–16. New Orleans, LA, USA.

Fan, L.; Pang, Z. Q.; Zhang, T. Y.; Wang, Y. X.; Zhao, H.; Wang, F.; Wang, N. Y.; and Zhang, Z. X. 2022. Embracing single stride 3D object detector with sparse transformer. In *CVPR*, 8448–8458. New Orleans, LA, USA.

Gu, A.; and Dao, T. 2023. Mamba: Linear-time sequence modeling with selective state spaces. arXiv:2312.00752.

Gu, A.; Dao, T.; Ermon, S.; Rudra, A.; and Ré, C. 2020. Hippo: Recurrent memory with optimal polynomial projections. In *NIPS*, 1474–1487. Virtual Event.

Gu, A.; Goel, K.; and Ré, C. 2022. Efficiently modeling long sequences with structured state spaces. In *ICLR*, 1–15. Virtual Event.

Gu, A.; Johnson, I.; Goel, K.; Saab, K.; Dao, T.; Rudra, A.; and Ré, C. 2021. Combining recurrent, convolutional, and continuous-time models with linear state space layers. In *NIPS*, 572–585. Virtual Event.

Guo, M. H.; Cai, J. X.; Liu, Z. N.; Mu, T. J.; Martin, R. R.; and Hu, S. M. 2021. PCT: Point cloud transformer. *Computational Visual Media*, 7(2): 187–199.

Gupta, A.; Gu, A.; and Berant, J. 2022. Diagonal State Spaces are as Effective as Structured State Spaces. In *NIPS*, 1–12. New Orleans, LA, USA.

Han, X.; Tang, Y.; Wang, Z. X.; and Li, X. Z. 2024. Mamba3D: Enhancing Local Features for 3D Point Cloud Analysis via State Space Model. In *ACM MM*, 1–10. Melbourne, Australia.

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

Hu, Q. Y.; Yang, B.; Xie, L. H.; Rosa, S.; Guo, Y. L.; Wang, Z. H.; Trigoni, N.; and Markham, A. 2022. Learning Semantic Segmentation of Large-Scale Point Clouds With Random Sampling. *PAMI*, 44(11): 8338–8354.

Li, Z. Y.; Ai, Y. B.; Lu, J. H.; Wang, C. X.; C., J.; Deng; Chang, H. Z.; Liang, Y. Z.; Yang, W. F.; Zhang, S. F.; and Zhang, T. Z. 2025. Pamba: Enhancing Global Interaction in Point Clouds via State Space Model. In *AAAI*, 5092–5100. Philadelphia, PA, USA.

Liang, D. K.; Feng, T. R.; Zhou, X.; Zhang, Y. M.; Zou, Z. K.; and Bai, X. 2025a. Parameter-Efficient Fine-Tuning in Spectral Domain for Point Cloud Learning. *PAMI*, 47(12): 10949–10966.

Liang, D. K.; Hua, W.; Shi, C. S.; Zou, Z. K.; Ye, X. Q.; and Bai, X. 2025b. SOOD++: Leveraging Unlabeled Data to Boost Oriented Object Detection. *PAMI*, 1–18.

Liang, D. K.; Zhou, X.; Xu, W.; Zhu, X. K.; Zou, Z. K.; Ye, X. Q.; Tan, X.; and Bai, X. 2024. PointMamba: A Simple State Space Model for Point Cloud Analysis. In *NIPS*, 1–14. Vancouver, BC, Canada.

Lin, H. J.; Zheng, X. W.; Li, L. J.; Chao, F.; Wang, S. S.; Wang, Y.; Tian, Y. H.; and Ji, R. R. 2023. Meta Architecture for Point Cloud Analysis. In *CVPR*, 17682–17691. Vancouver, BC, Canada.

Lin, Z. H.; Huang, S. Y.; and Wang, Y. F. 2022. Learning of 3D Graph Convolution Networks for Point Cloud Analysis. *PAMI*, 44(8): 4212–4224.

Liu, H. T.; Cai, M.; and Lee, Y. J. 2022. Masked discrimination for self-supervised learning on point clouds. In *ECCV*, 657–675. Tel Aviv, Israel.

Liu, J. M.; Han, J. R.; Liu, L. H.; Aviles-Rivero, A. I.; Jiang, C. K.; Liu, Z.; and Wang, H. S. 2025. Mamba4D: Efficient 4D Point Cloud Video Understanding with Disentangled Spatial-Temporal State Space Models. In *CVPR*, 17626–17636. Nashville, TN, USA.

Liu, J. M.; Yu, R. J.; Wang, Y.; Zheng, Y.; Deng, T. C.; Ye, W. C.; and Wang, H. S. 2024. Point Mamba: A Novel Point Cloud Backbone Based on State Space Model with Octree-Based Ordering Strategy. arXiv:2403.06467.

Liu, Z. J.; Yang, X. Y.; Tang, H. T.; Yang, S.; and Han, S. 2023. FlatFormer: Flattened Window Attention for Efficient Point Cloud Transformer. In *CVPR*, 1200–1211. Vancouver, BC, Canada.

Ma, X.; Qin, C.; You, H. X.; Ran, H. X.; and Fu, Y. 2022. Rethinking Network Design and Local Geometry in Point Cloud: A Simple Residual MLP Framework. In *ICLR*, 1–14. Virtual Event.

Mehta, H.; Gupta, A.; Cutkosky, A.; and Neyshabur, B. 2022. Long range language modeling via gated state spaces. arXiv:2206.13947.

Nie, D.; Lan, R.; Wang, L.; and Ren, X. F. 2022. Pyramid Architecture for Multi-Scale Processing in Point Cloud Segmentation. In *CVPR*, 17263–17273. New Orleans, LA, USA.

Pang, Y. T.; Wang, W. X.; Tay, F. E. H.; Liu, W.; Tian, Y. H.; and Yuan, L. 2022. Masked Autoencoders for Point Cloud Self-supervised Learning. In *ECCV*, 604–621. Tel Aviv, ISRAEL.

Park, C.; Jeong, Y.; Cho, M. S.; and Park, J. 2022. Fast point transformer. In *CVPR*, 16928–16937. New Orleans, LA, USA.

Qi, C. R.; Su, H.; Mo, K. C.; and Guibas, L. J. 2017a. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *CVPR*, 77–85. Honolulu, HI, USA.

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 *NIPS*, 5099–5108. Long Beach, CA, USA.

Qi, Z. K.; Dong, R. P.; Fan, G. F.; Ge, Z.; Zhang, X. Y.; Ma, K. S.; and Yi, L. 2023. Contrast with Reconstruct: Contrastive 3D Representation Learning Guided by Generative Pretraining. In *ICML*, 28223–28243. Honolulu, HI, USA.

Qian, G. C.; Li, Y. C.; Peng, H. W.; Mai, J. J.; Hammoud, H. A. A. K.; Elhoseiny, M.; and Ghanem, B. 2022. PointNeXt: Revisiting PointNet++ with Improved Training and Scaling Strategies. In *NIPS*, 1–13. New Orleans, LA, USA.

Robert, D.; Raguet, H.; and Landrieu, L. 2023. Efficient 3d semantic segmentation with superpoint transformer. In *ICCV*, 17149–17158. Paris, France.

Ronneberger, O.; Fischer, P.; and Brox, T. 2015. U-Net: Convolutional Networks for Biomedical Image Segmentation. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, 234–241. Munich, Germany.

Ruan, J. C.; and Xiang, S. C. 2024. VM-UNet: Vision Mamba UNet for Medical Image Segmentation. arXiv:2402.02491.

Smith, J. T. H.; Warrington, A.; and Linderman, S. W. 2022. Simplified state space layers for sequence modeling. arXiv:2208.04933.

Uy, M. A.; Pham, Q. H.; Hua, B. S.; Nguyen, D. T.; and Yeung, S. K. 2019. Revisiting Point Cloud Classification: A New Benchmark Dataset and Classification Model on Real-World Data. In *ICCV*, 1588–1597. Seoul, South Korea.

Wang, C.; Tsepa, O.; Ma, J.; and Wang, B. 2024a. Graph-Mamba: Towards Long-Range Graph Sequence Modeling with Selective State Spaces. arXiv:2402.00789.

Wang, P. S. 2023. OctFormer: Octree-based Transformers for 3D Point Clouds. *TOG*, 42(4): 155.

Wang, Y.; Sun, Y. B.; Liu, Z. W.; Sarma, S. E.; Bronstein, M. M.; and Solomon, J. M. 2019. Dynamic Graph CNN for Learning on Point Clouds. *TOG*, 38(5): 146.

Wang, Z. C.; Chen, Z. H.; Wu, Y. M.; Zhao, Z.; Zhou, L. P.; and Xu, D. 2024b. PoinTramba: A Hybrid Transformer-Mamba Framework for Point Cloud Analysis. arXiv:2405.15463.

Wu, X. Y.; Jiang, L.; Wang, P. S.; Liu, Z. J.; Liu, X. H.; Qiao, Y.; Ouyang, W. L.; He, T.; and Zhao, H. S. 2024. Point Transformer V3: Simpler, Faster, Stronger. In *CVPR*, 1–15. Seattle, WA, USA.

Wu, X. Y.; Lao, Y. X.; Jiang, L.; Liu, X. H.; and Zhao, H. S. 2022. Point transformer V2: Grouped vector attention and partition-based pooling. In *NIPS*, 1–13. New Orleans, LA, USA.

Wu, Z. R.; Song, S. R.; Khosla, A.; Yu, F.; Zhang, L. G.; Tang, X. O.; and Xiao, J. X. 2015. 3d shapenets: A deep representation for volumetric shapes. In *CVPR*, 1912–1920. Boston, MA, USA.

Xu, W.; Shi, C. S.; Tu, S. F.; Zhou, X.; Liang, D. K.; and Bai, X. 2024. A Unified Framework for 3D Scene Understanding. In *NIPS*, 1–14. Vancouver, BC, Canada.

Yang, Y. L.; Xun, T. Z.; Hao, K. R.; Wei, B.; and Tang, X. S. 2025. Grid Mamba:Grid State Space Model for Large-Scale Point Cloud Analysis. *Neurocomputing*, 636: 129985.

Yi, L.; Kim, V. G.; Ceylan, D.; Shen, I. C.; Yan, M. Y.; Su, H.; Lu, C.; Huang, Q. X.; Sheffer, A.; and Guibas, L. 2016. A scalable active framework for region annotation in 3d shape collections. *TOG*, 35(6): 210.

Yu, X. M.; Tang, L. L.; Rao, Y. M.; Huang, T. J.; Zhou, J.; and Lu, J. W. 2022. Point-BERT: Pre-training 3D Point Cloud Transformers with Masked Point Modeling. In *CVPR*, 19291–19300. New Orleans, LA, USA.

Zha, Y. H.; Li, N. Q.; Wang, Y. Z.; Dai, T.; Guo, H.; Chen, B.; Wang, Z.; Ouyang, Z. H.; and Xia, S. T. 2024. LCM: Locally Constrained Compact Point Cloud Model for Masked Point Modeling. In *NIPS*, 1–14. Vancouver, BC, Canada.

Zha, Y. H.; Wang, J. P.; Dai, T.; Chen, B.; Wang, Z.; and Xia, S. T. 2023. Instance-aware Dynamic Prompt Tuning for Pre-trained Point Cloud Models. In *ICCV*, 14115–14124. Paris, France.

Zha, Y. H.; Wang, Y. Z.; Guo, H.; Wang, J. P.; Dai, T.; Chen, B.; Ouyang, Z. H.; Xue, Y. R.; Chen, K.; and Xia, S. T. 2025. PMA: Towards Parameter-Efficient Point Cloud Understanding via Point Mamba Adapter. In *CVPR*, 16976–16986. Nashville, TN, USA.

Zhang, D. Y.; Liang, D. K.; Tan, Z. C.; Ye, X. Q.; Zhang, C.; Wang, J. D.; and Bai, X. 2024a. Make Your ViT-Based Multi-view 3D Detectors Faster via Token Compression. In *ECCV*, 56–72. Milan, Italy.

Zhang, D. Y.; Liang, D. K.; Zou, Z. K.; Li, J. Y.; Ye, X. Q.; Liu, Z.; Tan, X.; and Bai, X. 2023. A Simple Vision Transformer for Weakly Semi-supervised 3D Object Detection. In *ICCV*, 8339–8349. Paris, France.

Zhang, T.; Li, X. T.; Yuan, H. B.; Ji, S. P.; and Yan, S. C. 2024b. Point Cloud Mamba: Point Cloud Learning via State Space Model. arXiv:2403.00762.

Zhao, H. S.; Jiang, L.; Jia, J. Y.; Torr, P.; and Koltun, V. 2021. Point transformer. In *ICCV*, 16239–16248. Montreal, BC, Canada.

Zhou, X.; Liang, D. K.; Tu, S. F.; Chen, X. W.; Ding, Y. K.; Zhang, D. Y.; Tan, F. Y.; Zhao, H. S.; and Bai, X. 2025. HERMES: A Unified Self-Driving World Model for Simultaneous 3D Scene Understanding and Generation. In *ICCV*. Honolulu, HI, USA.

Zhou, X.; Liang, D. K.; Xu, W.; Zhu, X. K.; Xu, Y. H.; Zou, Z. K.; and Bai, X. 2024. Dynamic Adapter Meets Prompt Tuning: Parameter-Efficient Transfer Learning for Point Cloud Analysis. In *CVPR*, 14707–14717. Seattle, WA, USA.

Zhu, L. H.; Liao, B. C.; Zhang, Q.; Wang, X. L.; Liu, W. Y.; and Wang, X. G. 2024. Vision Mamba: Efficient Visual Representation Learning with Bidirectional State Space Model. In *ICML*, 1–10. Vienna, Austria.