

Pamba: Enhancing Global Interaction in Point Clouds via State Space Model

Zhuoyuan Li^{1*}, Yubo Ai^{1*}, Jiahao Lu¹, ChuXin Wang¹, Jiacheng Deng¹,
Hanzhi Chang¹, Yanzhe Liang¹, Wenfei Yang¹, Shifeng Zhang², Tianzhu Zhang^{1†}

¹Deep Space Exploration Laboratory/School of Information Science and Technology,
University of Science and Technology of China
²Sangfor Technologies Inc.

Abstract

Transformers have demonstrated impressive results for 3D point cloud semantic segmentation. However, the quadratic complexity of transformer makes computation costs high, limiting the number of points that can be processed simultaneously and impeding the modeling of long-range dependencies between objects in a single scene. Drawing inspiration from the great potential of recent state space models (SSM) for long sequence modeling, we introduce Mamba, an SSM-based architecture, to the point cloud domain and propose Pamba, a novel architecture with strong global modeling capability under linear complexity. Specifically, to make the disorderness of point clouds fit in with the causal nature of Mamba, we propose a multi-path serialization strategy applicable to point clouds. Besides, we propose the ConvMamba block to compensate for the shortcomings of Mamba in modeling local geometries and in unidirectional modeling. Pamba obtains state-of-the-art results on several 3D point cloud segmentation tasks, including ScanNet v2, ScanNet200, S3DIS and nuScenes, while its effectiveness is validated by extensive experiments.

1 Introduction

3D point cloud semantic segmentation is a fundamental task in 3D scene understanding, which aims to predict the semantic labels for all points in the scene. As a critical technique for understanding realistic scenes, 3D point cloud semantic segmentation has various applications, including robotics (Seita et al. 2023), automatic driving (Geiger et al. 2013; Caesar et al. 2019; Fong et al. 2021; Ettinger et al. 2021) and AR/VR (Park, Kim, and Lee 2020). However, the interaction between different points at different scales in the scene poses challenges to precise 3D point cloud semantic segmentation.

To overcome the above challenges, a variety of 3D semantic segmentation methods have been proposed, which mainly fall into two categories: voxel-based methods and point-based methods. Voxel-based methods first quantize irregular point clouds into regular voxel representations and then perform 3D convolutions on the voxels (Maturana and Scherer 2015; Deng et al. 2021). The cubic growth in the number of voxels

*These authors contributed equally.

†Corresponding author: Tianzhu Zhang.

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

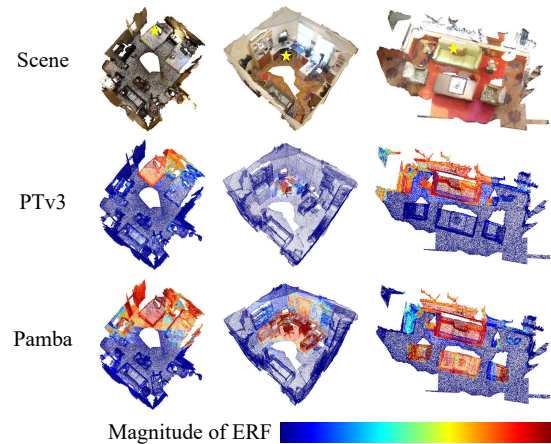


Figure 1: Visualization of effective receptive fields (ERF) of the point of interest on ScanNet200 dataset. The yellow star represents the position of the point of interest. Pamba shows larger ERF and the ability of handling long-range interactions between different objects in a scene. More illustrations are provided in Appendix.

as a function of resolution leads to significant inefficiency, which is not solved until the proposal of sparse convolutions (Graham, Engelcke, and Van Der Maaten 2018; Choy and Savarese 2019). However, the quantization loss during voxelization always exists. Therefore, some point-based methods that directly handle the points are proposed (Qi et al. 2017a,b; Thomas et al. 2019; Wang et al. 2019). The pioneering work PointNet (Qi et al. 2017a) adopts permutation-invariant operators to aggregate features across the whole point cloud. Recently, inspired by the big success of transformer in the field of vision (Dosovitskiy et al. 2020; Wu and Zhang 2021; Liu et al. 2021) and natural language processing (Devlin et al. 2018; Radford et al. 2018; Touvron et al. 2023), many works (Zhao and Koltun 2021; Wu and Zhao 2022; Wang 2023; Liu et al. 2023; Lai et al. 2023; Yang et al. 2023), which are categorized into point-based methods, incorporate transformer into point cloud analysis and achieve exceptional performance. Point Transformer (Zhao and Koltun 2021) utilizes KNN (Cover and Hart 1967) to construct the neighbourhood, in which local attention is performed. PTv2 (Wu

and Zhao 2022) adopts grids to partition the point cloud into non-overlapping patches and performs attention mechanism per patch. Further, FlatFormer (Liu et al. 2023), OctFormer (Wang 2023), and PTV3 (Wu et al. 2024) adopt serialization-based methods to partition point clouds. The superior performance of transformer-based methods is attributed to transformer’s strong ability of modeling long-range dependencies in large reception fields (Vaswani et al. 2017; Raghu et al. 2021). However, transformer-based approaches are flawed in terms of scalability. Specifically, the quadratic complexity of transformer makes computation costs high, limiting the number of points that can be processed simultaneously and impeding the modeling of long-range interactions between objects in a single scene. For ease of illustration, we visualize the effective receptive fields (ERF) of a previous method, PTV3, in Fig. 1. The visualization demonstrates that previous methods focus more on semantically consistent neighbouring points but fail to capture interactions between objects due to limited ERF. As an example, in this context, a bed is more likely to appear against a wall and another bed, a desk is more likely to come in a set with a chair, a sofa is more likely to have a table and other sofas next to it. This prior knowledge matches human visual perception and can make models perform better.

Recent research advancements have sparked considerable interest in state space models (SSM) (Gu, Goel, and Ré 2021a; Gu et al. 2021; Gupta, Gu, and Berant 2022; Gu, Goel, and Ré 2021b), which excel at capturing long-range dependencies under linear complexity while benefiting from parallel training. In particular, an SSM-based architecture, Mamba, demonstrates superior performance for NLP tasks to rival transformer (Gu and Dao 2023). This leads us to think: Is it possible to introduce Mamba into the point cloud scene understanding tasks to solve the scalability problem of existing transformer-based methods? However, we find the direct application of Mamba into 3D scene understanding tasks results in poor performance. After analysis, we point out three main problems with processing point clouds using Mamba. **1) Permutation sensitivity:** Mamba is designed to process the causal sequence (Gu and Dao 2023), which is highly sensitive to the input order. Different orders of input points can result in different outputs and greatly impact the final result. **2) Insufficiently strong local modeling ability:** Mamba enhances the modeling of global features by compressing all contexts into a specific state (Gu and Dao 2023). However, many irrelevant contexts are redundant for local modeling, which sacrifices the representation quality of local geometries. **3) Unidirectional modeling:** Mamba performs unidirectional modeling. For a point cloud sequence processed with Mamba, a point can only interact with points before this point, but not the points after this point, which hinders the bidirectional interaction between different points.

Based on the above analyses, we propose a novel point cloud scene understanding framework, **Pamba**, to address the above problems and fully unleash the potential of Mamba in the point cloud domain. First, we propose the **multi-path serialization** strategy to adapt to the permutation sensitivity of Mamba. Specifically, it rearranges the unordered point cloud to an ordered point sequence according to a pre-defined

pattern so that points that are adjacent in sequence are also neighbouring in space. Based on previous Hilbert and z-order serialization patterns, we redesign a new pattern, named hz serialization, for a better fusion of spatial information in different perspectives within each layer. Then, we randomly assign hz serialization patterns to different blocks to enable the interaction of spatial information in different perspectives across layers. Besides, we propose the **ConvMamba** block to compensate for the shortcomings of Mamba in modeling local geometries and in unidirectional modeling. In detail, it combines convolutions with Mamba to extract both long-distance dependencies and local geometries simultaneously. Moreover, bidirectionality is introduced to ConvMamba to enhance the bidirectional interaction between points. Beyond the above, to enable global modeling, Pamba processes the entire point cloud directly, unlike previous transformer-based methods (Zhao and Koltun 2021; Wu and Zhao 2022; Wang 2023; Liu et al. 2023; Lai et al. 2023; Yang et al. 2023), which split the point cloud into patches and then process them separately. Taking ScanNet v2 (the average number of points is 148k) as an example, Pamba directly processes over 100k points without dividing the points into groups.

As shown in Fig. 1, Pamba has a larger ERF and can capture interactions not only between semantically similar neighbouring parts, but also between objects, e.g., the interaction between sofa and table or the interaction between desk and chair. Notably, Pamba focuses on how to utilize Mamba to capture long-range interactions between points that are hindered in transformer, rather than on some intricate design. The contributions can be summarised as follows:

- We propose a new framework Pamba as a direct application of Mamba to semantic segmentation of 3D point clouds, which captures long-range interaction under linear complexity.
- We propose the multi-path serialization strategy and the ConvMamba block to help Mamba better adapted to point clouds. The former one enables the model to capture spatial information in different perspectives while the latter one compensates for the shortcomings of Mamba in modeling local geometries and in unidirectional modeling.
- We conduct extensive experiments and ablation studies to validate our design choices. Pamba achieves state-of-the-art performance on several highly competitive point cloud segmentation tasks, including ScanNet v2, ScanNet200, S3DIS, and nuScenes.

Point cloud transformer. It is natural to extend transformer into point cloud understanding after the big success of vision transformers (Dosovitskiy et al. 2020), which can be counted as a sub-category of point-based methods. PCT (Guo, Cai, and Hu 2021) and Point Transformer (Zhao and Koltun 2021) are the pioneers in introducing transformer into the field of point cloud. PCT (Guo, Cai, and Hu 2021) directly applies global attention to all points inside the point cloud and thus can only handle point clouds with a few thousand points due to the quadratic complexity of transformer. In contrast, Point Transformer (Zhao and Koltun 2021) first extracts the points’ neighbourhood by KNN (Cover and Hart 1967),

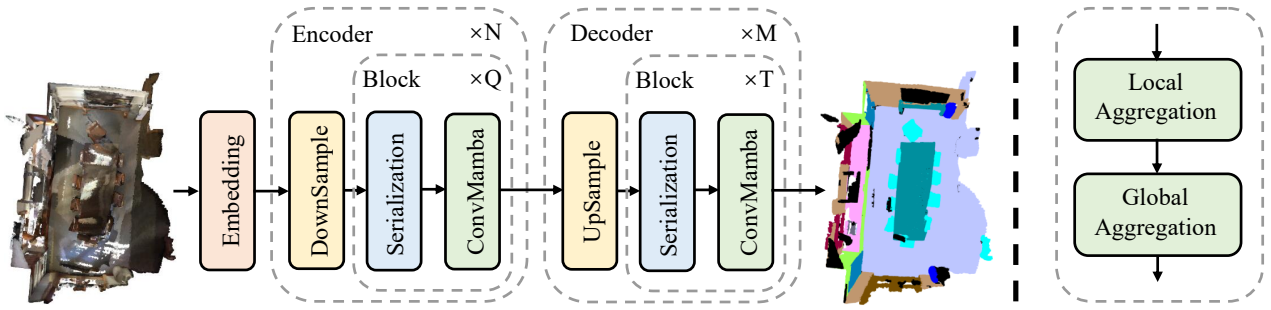


Figure 2: Left: The overall architecture of Pamba; Right: ConvMamba block.

in which the local attention is then applied, achieving much less memory costs than PCT. Following Point Transformer (Zhao and Koltun 2021), many transformer-based methods spring up and achieve state-of-the-art performance, such as PTV2 (Wu and Zhao 2022), PTV3 (Wu et al. 2024), Stratified Transformer (Lai et al. 2022), PatchFormer (Zhang et al. 2022), QueryFormer (Lu, Deng, and Wang 2023) etc.

Point cloud serialization. Unlike the traditional paradigm of partitioning the point cloud into patches through KNN or similar methods, several recent works (Liu et al. 2023; Wang 2023; Wu et al. 2024) propose to arrange the irregular and scattered points into a structured point sequence by directly sorting the whole point cloud through specific patterns, which are categorized as serialization-based methods. These approaches achieve strong performance while keeping memory overhead low, as sorting algorithms could be much more efficient than KNN-related algorithms. FlatFormer (Liu et al. 2023) utilizes windows to partition the point cloud, then sorts windows along x and y axis. However, directly sorting along axes may result in the destruction of spatial proximity, leading to a performance drop. OctFormer (Wang 2023) follows the octree structure to sort the whole point cloud. The nature of the octree ensures that the spatial proximity of the unstructured point cloud can be well preserved during the serialization. Based on prior works, PTV3 (Wu et al. 2024) further combines the Hilbert serialization and the z-order serialization, unleashing the greater potential of serialization-based methods.

State space models. State space models (SSM) originate from the classic Kalman filter model in control systems. SSM can either model long-range interactions like RNN or be trained in parallel like transformer, achieving high efficiency. Recently, many variants of SSM have been proposed, including linear state-space layers (Gu et al. 2021), structured state space model (Gu, Goel, and Ré 2021b), and diagonal state space (Gupta, Gu, and Berant 2022). Mamba (Gu and Dao 2023) is the state-of-the-art SSM-based architecture. It proposes a selective mechanism so that the model parameters vary with inputs, allowing the model to compress context selectively according to current input (Gu and Dao 2023). This principle further enhances the ability to model long-range dependencies. Several recent works adapt Mamba to different fields, including vision (Liu et al. 2024b; Zhu et al. 2024; Liu et al. 2024a; Guo et al. 2024; Pei, Huang, and Xu 2024), graph

neural network (Wang et al. 2024a; Behrouz and Hashemi 2024; Li et al. 2024b) and video (Yang, Xing, and Zhu 2024; Li et al. 2024a). Some concurrent works PM (Liang, Zhou, and Bai 2024) PoinTramba (Wang et al. 2024c), SPM (Wang et al. 2024b) and PCM (Zhang et al. 2024) also apply Mamba to 3D point clouds. However, PM is a pre-training architecture similar to Point-MAE (Pang et al. 2022). PoinTramba enhances PM by combining attention with Mamba. PCM and SPM only apply Mamba within the neighbourhood extracted by KNN, which ignores the global interaction and does not fully utilize Mamba’s ability to capture long-range dependencies. Besides, the above works are mainly evaluated on object-level datasets (Wu and Xiao 2015; Yi et al. 2016; Uy et al. 2019) and not on scene-level datasets, as the latter one is crucial for 3D scene understanding.

2 Method

The general structure of Pamba is shown in Fig. 2. Pamba is inherited from PTV3 (Wu et al. 2024), the state-of-the-art backbone for point cloud perception tasks. PTV3 is chosen for its concise pipeline as well as its strong performance. We start with a brief preliminary of state space models and space-filling curves in Section 3.1. In Section 3.2, we introduce the multi-path serialization strategy. In Section 3.3, we introduce ConvMamba, the main block of Pamba.

2.1 Preliminary

State space model. The state space model (SSM) is initially introduced in the field of control engineering to model dynamic systems. Specifically, the SSM in deep learning encompasses three key variables: the input sequence $x(t)$, the latent state representation $h(t)$, and the output sequence $y(t)$. Additionally, it includes two fundamental equations: the state equation and the observation equation, with \mathbf{A} , \mathbf{B} and \mathbf{C} being system parameters. The SSM is formulated in Eq. 1.

$$\begin{aligned} h'(t) &= \mathbf{A}h(t) + \mathbf{B}x(t), \\ y(t) &= \mathbf{C}h(t). \end{aligned} \quad (1)$$

Mamba. Discretizing the SSM is crucial because it is initially designed for continuous systems and cannot handle discrete data such as images or point clouds. Gu and Dao utilizes the zero-order hold technique to discretize the SSM with a time step Δ . In detail, the continuous parameters \mathbf{A} , \mathbf{B} ,

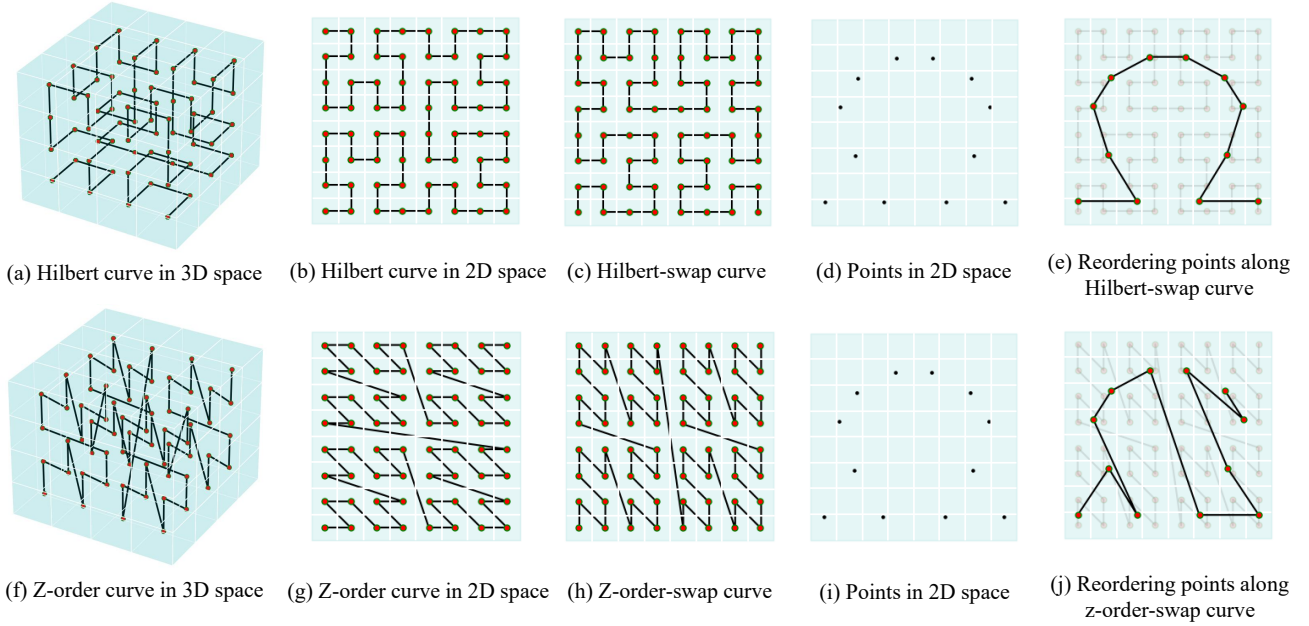


Figure 3: Space-filling curves.

\mathbf{C} are transformed to discrete parameters $\bar{\mathbf{A}}, \bar{\mathbf{B}}, \bar{\mathbf{C}}$ as shown in Eq. 2

$$\bar{\mathbf{A}} = e^{\Delta\mathbf{A}}, \quad \bar{\mathbf{B}} = (e^{\Delta\mathbf{A}} - \mathbf{I})\mathbf{A}^{-1}\mathbf{B}, \quad \bar{\mathbf{C}} = \mathbf{C}. \quad (2)$$

After discretization, the calculation process of SSM can be simplified into a convolution operation, enabling the entire SSM to be trained in parallel, as shown in Eq. 3.

$$\bar{\mathbf{y}} = \mathbf{x} \otimes \bar{\mathbf{K}}, \quad \bar{\mathbf{K}} = (\bar{\mathbf{C}}\bar{\mathbf{B}}, \bar{\mathbf{C}}\bar{\mathbf{A}}\bar{\mathbf{B}}, \dots, \bar{\mathbf{C}}\bar{\mathbf{A}}^{k-1}\bar{\mathbf{B}}). \quad (3)$$

Space-filling curve. A space-filling curve (Peano and Peano 1990) is a curve that fills a multi-dimensional space. When the dimension equals three in the context of point clouds, the space-filling curve traverses all points within a discrete 3D cube without repetition. The best-known space-filling curves include Hilbert curve (Hilbert and Hilbert 1935) and z-order curve (Morton 1966), as shown in Fig. 3 (a), (b), (f) and (g) with a dimension of three and two respectively. Z-order curves are known for its high efficiency, while Hilbert curves are known for its locality-preserving property. For ease of illustration, we elaborate the following in 2D.

The shown space-filling curves in Fig. 3 (b) and (g) adopt a traversal along x , y , and z axes in the order of priority. By simply changing the order of the three axes, similar variants of the space-filling curves are obtained. Here we show two other variants of the space-filling curve named Hilbert-swap curve (swap stands for swapping x axis and y axis) and z-order-swap curve by exchanging the order of the x axis and y axis for traversal as shown in Fig. 3 (c) and (h).

Based on the space-filling curve, an intuitive idea is that points in space can be sorted into a 1-D point sequence along the space-filling curve, which we call point cloud serialization. OctFormer (Wang 2023) and PTV3 (Wu et al. 2024) are

the pioneers that apply the above curves to point clouds. OctFormer (Wang 2023) utilizes z-order curves for serialization, while PTV3 utilizes both Hilbert curves and z-order curves. The spatial proximity in point clouds can be preserved well through point cloud serialization, i.e., points that are adjacent in sequences are also neighbouring in point clouds. An example of some randomly located points is shown in Fig. 3 (d) and (i), which are separately sorted using the Hilbert-swap curve and the z-order-swap curve in Fig. 3 (e) and (j). It could be observed that the Hilbert curve preserves better spatial proximity than the z-order curve, which is already proved by some previous work (Nordin and Telles 2023).

2.2 Multi-path serialization strategy

As Hilbert-based serialization keeps better spatial proximity than z-order-based serialization (Nordin and Telles 2023), it is natural to apply Hilbert-based serialization rather than z-order serialization. However, we observe poor performance when simply applying Hilbert-based serialization in each ConvMamba block in Tab. 5. We attribute this to a single ordering pattern lacking the spatial relationships in multiple perspectives provided by other ordering patterns. PTV3 (Wu et al. 2024) solves this by proposing the "shuffle order strategy" that assigns different serialization patterns to different layers so that the spatial relationships in multiple perspectives provided by different serialization patterns are mixed across layers. However, within each layer, the spatial information therein is still obtained from a single serialization pattern.

To further enhance the fusion of multi-perspective spatial relationships, we propose the multi-path serialization strategy. In detail, we redesign two brand new serialization patterns, namely the hz curve and the hz-swap curve, by integrating both Hilbert curves and z-order curves as shown in Fig. 4.

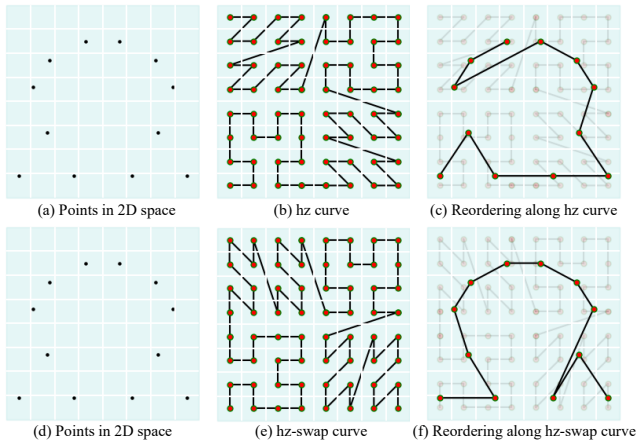


Figure 4: Our introduced hz curve and hz-swap curve.

Examples of sorting along the hz curve and the hz-swap curve are shown in Fig. 4 (c) and (f). Then, for each ConvMamba block, we randomly allocate one pattern from the hz curve and the hz-swap curve to the block. With this strategy, the model captures multi-perspective spatial relationships in each layer and achieves better robustness.

2.3 ConvMamba

In this section, we introduce the ConvMamba block, which aims to synergistically capture global dependencies and local features. It consists of two stages, local aggregation and global aggregation, going serially as shown in Fig. 2.

Global Aggregation. Due to the linear complexity of Mamba, we apply Mamba to process the whole point cloud (more than 100k points) at once instead of applying Mamba within patches like what point transformers (Zhao and Koltun 2021; Wu and Zhao 2022) does, realizing long-range interaction. However, one inherent drawback in the original Mamba (Gu and Dao 2023) is its causality as shown in Fig. 5 (a), meaning a point in the serialized point sequence can only interact with points before this point, not the points after this point. In other words, the block in Fig. 5 (a) only scans the input sequence in one direction. To address this, we introduce the bidirectional mamba mechanism as shown in Fig. 5 (b). Specifically, the whole point cloud is scanned from both forward and backward directions, enabling each point capable of interacting with points on either side of it. It is worth noting that ‘forward SSM’ and ‘backward SSM’ share the same parameters, which is different from some other Mamba-based works (Liu et al. 2024b; Zhang et al. 2024). This principle corresponds with our intention of obtaining consistent features from two opposite scans, putting a consistency constraint on both opposite scans. We then follow the traditional transformer block (Vaswani et al. 2017) and pre-norm pattern (Child et al. 2019) to construct the global aggregation stage by applying an MLP after bidirectional Mamba, both with normalization and skip connection as shown in Fig.5 (c).

Local Aggregation. Bidirectional Mamba is applied to the whole point cloud to capture long-range dependencies by

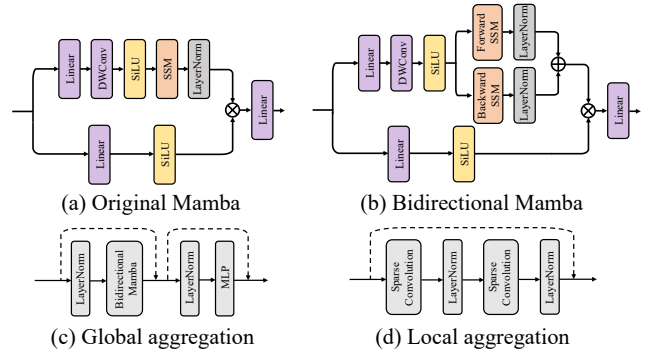


Figure 5: Bidirectional Mamba. (a) Original Mamba structure; (b) Proposed bidirectional Mamba; (c) Global aggregation.

compressing all context into a hidden state. However, some irrelevant contexts are redundant for local modeling, which sacrifices the representation quality of local geometries. Unfortunately, local features are proven to be essential to point clouds (Thomas et al. 2019; Deng et al. 2023; Duan et al. 2024). To address this, we add a local aggregation stage right before the global aggregation stage to explicitly aggregate local information and compensate for Mamba’s shortcomings in modelling local geometries. Inspired by (Chu et al. 2021; Wang 2023; Wu et al. 2024), which verified that various convolutions not only aggregate local features but also provide location information, we simply utilize sparse submanifold convolutions to form the local aggregation stage for its high efficiency and low memory usage. Specifically, we adopt two consecutive sparse convolutions to achieve a larger receptive field than a single convolution, as shown in Fig. 5 (d), as the local information extracted from a larger context is more helpful to the long-range interaction of over 100K points, which is verified in ablations.

3 Experiments

In this section, we aim to evaluate the effectiveness of our proposed Pamba. We introduce the main results of 3D semantic segmentation tasks in section 4.1. In section 4.2, we evaluate the efficiency of Pamba. In section 4.3, we conduct ablation studies on the design choice of Pamba. In section 4.4, we discuss limitations and future work.

3.1 Semantic segmentation

Dataset. We evaluate Pamba on four datasets: ScanNet v2 (Dai et al. 2017), ScanNet200 (Rozenberszki and Dai 2022), S3DIS (Armeni et al. 2016) and nuScenes (Caesar et al. 2019; Fong et al. 2021). ScanNet v2 is a commonly used indoor dataset, containing 1513 room scans in total, the average point number of which is 148k. ScanNet200 shares the same data with ScanNet but has 200 semantic categories, making precise predictions more difficult. The S3DIS dataset is another indoor dataset consisting of 271 rooms in six areas from three different buildings, with a total of 13 semantic categories. nuScenes is an outdoor dataset containing 1,000 driving sequences in total, which is usually more difficult

Methods	Scannet Val	ScanNet200 Val	nuScenes Val	S3DIS Area 5
Point				
PointNet++	53.5	-	-	-
PointConv	61.0	-	-	-
KPConv	69.2	-	-	67.1
PointNeXt	71.5	-	-	70.5
PointMetaBase	72.8	-	-	71.3
Transformer				
PTv1	70.6	27.8	-	70.4
PTv2	75.4	30.2	80.2	71.6
StratifiedFormer	74.3	-	-	72.0
OctFormer	75.7	32.6	-	-
SphereFormer	-	-	78.4	-
Swim3D	76.4	-	-	72.5
PTv3	77.5	35.2	80.4	73.4
Mamba				
PCM	-	-	-	63.4
Pamba	77.6	36.3	80.4	73.5

Table 1: Semantic segmentation result

to handle than indoor datasets. ScanNet, ScanNet200 and nuScenes follow the official data splits to generate corresponding training set, validation set and test set. For S3DIS, the data is split into six areas, the fifth of which is withheld during training and used for evaluation.

Setting. We train our model on 4 RTX 3090 GPUs. AdamW (Loshchilov and Hutter 2017) is adopted for parameter optimization. ScanNet and ScanNet200 are trained 800 epochs, S3DIS is trained 3000 epochs and nuScenes is trained 50 epochs. The sum of CrossEntropy loss and Lovasz loss (Berman, Triki, and Blaschko 2018) is adopted as the overall loss for Pamba as shown in Eq. 4, where L_{CE} represents CrossEntropy loss and L_L represents Lovasz loss. λ_1 and λ_2 are both empirically set to 0.5 during implementation.

$$L = \lambda_1 \cdot L_{CE} + \lambda_2 \cdot L_L \quad (4)$$

Main results. We compare Pamba with a variety of previous state-of-the-art models, including PointNet++ (2017b), PointConv (2019), KPConv (2019), Cylinder3D (2020), PointNeXt (2022), PointMetaBase (2023), PTv1 (2021), PTv2 (2022), StratifiedFormer (2022), OctFormer (2023), SphereFormer (2023), Swim3D (2023), PTv3 (2024) and PCM (2024), using mean Intersection over Union (mIoU) as the metric in Tab. 1. Pamba shows significant priority and exceeds most previous methods. Additionally, Pamba demonstrates competitive performance with PTv3 (2024), the most advanced transformer-based method, and even surpasses it on some datasets.

In detail, for indoor datasets, on ScanNet v2 and S3DIS, Pamba shows comparable performance with PTv3, achieving a mIoU of 77.6% and 73.5% respectively. For outdoor scenes, Pamba is also capable of reaching a mIoU of 80.4%

Methods	Training		Inference	
	Latency	Memory	Latency	Memory
OctFormer (2023)	357ms	9.5G	120ms	9.3G
Swim3D (2023)	758ms	10.3G	529ms	7.0G
PTv2 (2022)	398ms	10.1G	230ms	13.0G
PTv3 (2024)	223ms	5.2G	108ms	4.5G
Pamba	249ms	5.2G	179ms	4.8G

Table 2: Model Efficiency

on nuScenes, showing competitive performance and demonstrating Pamba’s strong generalization ability over both indoor scenes and outdoor scenes.

When it comes to complex scenes with more categories on ScanNet200, Pamba further demonstrates superior performance, achieving a mIoU of 36.3% and surpassing the previous state-of-the-art by 1.1%, which indicates Pamba’s ability of handling complexly-labelled scenarios like real-life scenes. Besides, we conclude that the long-range interaction constructed by Pamba is more effective in complex scenarios with more categories than in simple scenarios. In simple scenarios with dozens of categories, modelling local geometries is enough to predict the right label and interactions between objects of different categories may help less. In complex scenarios with hundreds of categories, complicated coupling between points of different categories could lead to less accurate local modelling. In this case, the constructed interactions by Pamba can effectively alleviate the complexity that comes from excessive categories and enhance model performance.

3.2 Model efficiency

We measure the efficiency of Pamba through two metrics: mean latency and mean memory consumption on ScanNet200 dataset. All measurements are taken on a single RTX 3090 and are compared with previous methods as shown in Tab. 2. Specifically, mean memory consumption is the memory per GPU recorded during training divided by the batch size.

Memory consumption. Tab. 2 compares the mean memory consumption of Pamba with previous works during both the training stage and inference stage. During training, Pamba demonstrates a low memory consumption compared with all previous work. During inference, Pamba exhibits a slightly larger memory consumption, 0.3G greater than PTv3, but is still lower than other methods.

Model latency. Pamba maintains a better latency than many previous methods, while is still slightly slower than some efficient transformer-based models, such as OctFormer (Wang 2023) and PTv3, as shown in Tab. 2. We attribute this phenomenon to the poor parallelism of our proposed Pamba. In detail, OctFormer and PTv3 follow the common paradigm that first partitions the whole point cloud into patches with the same number of points, which can then be handled by self-attention mechanism in parallel at patch level. However, our Pamba handles the whole point cloud at once under the global reception field, with each point being calculated serially. This

Type of Mamba	Val
No Mamba	26.1
Unidirectional Mamba	33.2
Bidirectional Mamba w/ different params	35.7
Bidirectional Mamba	36.3

Table 3: Effectiveness of Mamba.

Depth	Val
0	31.9
1	35.9
2	36.3
3	36.0
4	35.4

Table 4: Depth of local aggregation.

intrinsic difference is the key to the long-range interaction in Pamba, while inevitably making Pamba a bit slower. We posit this trade-off is beneficial overall. Additionally, transformer-based methods benefit from FlashAttention (Dao et al. 2022), an algorithm for deep optimization of attention mechanism in hardware, while SSMs have yet to be studied enough in terms of acceleration as an emerging model.

3.3 Ablation study

Here, we verify the key design choices of Pamba. All ablation studies are conducted on ScanNet200 validation set.

Effectiveness of Mamba. We verify the effectiveness of Mamba in Tab. 3. In detail, three experiments are conducted. First, we conduct experiments by removing all Mamba modules in Pamba, which results in a pure-convolution model. Second, we replace the bidirectional Mamba with unidirectional Mamba. Third, we make the ‘forward SSM’ and ‘backward SSM’ of bidirectional Mamba use different parameters. The result shows that pure-convolution (No Mamba) underperforms, indicating that the long-range interaction constructed by Mamba is essential. The bidirectionality further enhances the long-range interaction and achieves better performance. Besides, the consistency constraint on the forward scan and backward scan is effective.

Depth of local aggregation. In Tab. 4, we conduct experiments on the effects of different numbers of sparse convolutions employed in the local aggregation stage (including zero, meaning no local aggregation). Tab. 4 shows a non-negligible performance gap between Pamba and Pamba without sparse convolutions, indicating that the local aggregation is essential to Pamba. Besides, the depth of 1, 2, 3 and 4 demonstrates similar performance on ScanNet200. Since sparse convolutions are highly efficient, we simply adopt the depth of 2.

Serialization strategy. In Tab. 5, we verify the effectiveness of different combinations of serialization patterns while other settings remain the same as described in the method. First, simply adopting a single serialization pattern generates

Serialization combination	Val
Hilbert	34.4
Z	34.3
Hilbert + Hilbert-swap	35.0
Z + Z-swap	34.7
hz	35.1
Hilbert + Hilbert-swap + Z + Z-swap	35.8
Hilbert + Hilbert-swap + Z + Z-swap + hz + hz-swap	35.8
hz + hz-swap	36.3

Table 5: Different serialization combination.

a base performance. Further results suggest that the increase in the number of serialization patterns obviously improves the model’s performance, which corresponds with our viewpoint that different serialization patterns offer different perspectives on the spatial relationship in point clouds. Besides, the Hilbert-based patterns generally outperform z-order-based patterns. Furthermore, by mixing Hilbert and z-order, the hz-based curve makes the performance even more powerful, up to a mIOU of 36.3%, indicating the mixing of spatial relationships in multiple perspectives within layers is beneficial.

3.4 Limitation

Even though Pamba achieves a considerable latency compared with many previous methods, it is still slightly slower than some efficient transformer-based models, such as OctFormer (Wang 2023) and PTv3 (Wu et al. 2024). We analyze that this efficiency gap comes from the poor parallelism of Pamba. Specifically, OctFormer and PTv3 follow the traditional paradigm that first divides the whole point cloud into groups with the same number of points, which can then be handled by self-attention mechanism in parallel at group level. In contrast, Pamba handles the whole point cloud at once under the global reception field, with each point being calculated serially. This intrinsic difference inevitably makes Pamba a bit slower, reinforcing the need for continued exploration of efficient SSM mechanisms with better parallelism. Besides, Pamba is only evaluated on semantic segmentation tasks and may be extended to other tasks in the future, e.g., instance segmentation and object detection.

4 Conclusion

We explore the direct application of Mamba to semantic segmentation of 3D point clouds, which is a challenging and versatile task for evaluating different techniques. Unlike previous point transformers, we do not follow the common paradigm that first partitions the whole point cloud into patches and then processes each patch in parallel. In contrast, we treat the whole point cloud as a single ‘patch’ and pass it to a Mamba layer to enable global interaction. We make some non-trivial improvements to adapt Mamba to the field of point clouds, including the multi-path serialization strategy and the ConvMamba block. Our Pamba exceeds the state of the art on many point cloud semantic segmentation tasks.

Acknowledgments

This work was supported by National Natural Science Foundation of China (62306294, 62394354), Anhui Provincial Natural Science Foundation 2308085QF222, Youth Innovation Promotion Association.

References

- Armeni, I.; Sener, O.; Zamir, A. R.; Brilakis, I.; Fischer, M.; and Savarese, S. 2016. 3d semantic parsing of large-scale indoor spaces. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 1534–1543.
- Behrouz, A.; and Hashemi, F. 2024. Graph Mamba: Towards Learning on Graphs with State Space Models. *arXiv:2402.08678*.
- Berman, M.; Triki, A. R.; and Blaschko, M. B. 2018. The lovász-softmax loss: A tractable surrogate for the optimization of the intersection-over-union measure in neural networks. In *CVPR*, 4413–4421.
- Caesar, H.; Bankiti, V.; Lang, A. H.; Baldan, G.; and Beijbom, O. 2019. nuScenes: A multimodal dataset for autonomous driving. *arXiv:1903.11027*.
- Child, R.; Gray, S.; Radford, A.; and Sutskever, I. 2019. Generating long sequences with sparse transformers. *arXiv preprint arXiv:1904.10509*.
- Choy, C.; and Savarese, S. 2019. 4d spatio-temporal convnets: Minkowski convolutional neural networks. In *CVPR*.
- Chu, X.; Tian, Z.; Zhang, B.; Wang, X.; and Shen, C. 2021. Conditional positional encodings for vision transformers. *arXiv preprint arXiv:2102.10882*.
- Cover, T.; and Hart, P. 1967. Nearest neighbor pattern classification. *IEEE transactions on information theory*, 21–27.
- Dai, A.; Chang, A. X.; Savva, M.; Halber, M.; Funkhouser, T.; and Nießner, M. 2017. ScanNet: Richly-annotated 3D Reconstructions of Indoor Scenes. In *Proc. Computer Vision and Pattern Recognition (CVPR), IEEE*.
- Dao, T.; Fu, D.; Ermon, S.; Rudra, A.; and Ré, C. 2022. Flashattention: Fast and memory-efficient exact attention with io-awareness. *NeurIPS*, 35: 16344–16359.
- Deng, J.; Shi, S.; Li, P.; Zhou, W.; Zhang, Y.; and Li, H. 2021. Voxel r-cnn: Towards high performance voxel-based 3d object detection. In *Proceedings of the AAAI conference on artificial intelligence*, 2, 1201–1209.
- Deng, X.; Zhang, W.; Ding, Q.; and Zhang, X. 2023. Pointvector: a vector representation in point cloud analysis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 9455–9465.
- Devlin, J.; Chang, M.-W.; Lee, K.; and Toutanova, K. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.
- Dosovitskiy, A.; Beyer, L.; Heigold, G.; Gelly, S.; et al. 2020. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*.
- Duan, L.; Zhao, S.; Xue, N.; and Tao, D. 2024. ConDaFormer: Disassembled Transformer with Local Structure Enhancement for 3D Point Cloud Understanding. *NeurIPS*, 36.
- Ettinger, S.; Cheng, S.; Qi, C. R.; Zhou, Y.; et al. 2021. Large scale interactive motion forecasting for autonomous driving: The waymo open motion dataset. In *CVPR*, 9710–9719.
- Fong, W. K.; Mohan, R.; Caesar, H.; and Beijbom, O. 2021. Panoptic nuScenes: A Large-Scale Benchmark for LiDAR Panoptic Segmentation and Tracking. *arXiv:2109.03805*.
- Geiger, A.; Lenz, P.; Stiller, C.; and Urtasun, R. 2013. Vision meets robotics: The kitti dataset. *The International Journal of Robotics Research*, 32(11): 1231–1237.
- Graham, B.; Engelcke, M.; and Van Der Maaten, L. 2018. 3d semantic segmentation with submanifold sparse convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 9224–9232.
- Gu, A.; and Dao, T. 2023. Mamba: Linear-time sequence modeling with selective state spaces. *arXiv:2312.00752*.
- Gu, A.; Goel, K.; and Ré, C. 2021a. Efficiently modeling long sequences with structured state spaces. *arXiv:2111.00396*.
- Gu, A.; Goel, K.; and Ré, C. 2021b. Efficiently modeling long sequences with structured state spaces. *arXiv:2111.00396*.
- Gu, A.; Johnson, I.; Goel, K.; Rudra, A.; and Ré, C. 2021. Combining recurrent, convolutional, and continuous-time models with linear state space layers. *NeurIPS*, 34: 572–585.
- Guo, H.; Li, J.; Dai, T.; Ouyang, Z.; Ren, X.; and Xia, S.-T. 2024. MambaR: A Simple Baseline for Image Restoration with State-Space Model. *arXiv preprint arXiv:2402.15648*.
- Guo, M.-H.; Cai, J.-X.; and Hu, S.-M. 2021. Pct: Point cloud transformer. *Computational Visual Media*, 7: 187–199.
- Gupta, A.; Gu, A.; and Berant, J. 2022. Diagonal state spaces are as effective as structured state spaces. *NeurIPS*, 35.
- Hilbert, D.; and Hilbert, D. 1935. *Neubegründung der mathematik. erste mitteilung*. Springer.
- Lai, X.; Chen, Y.; Lu, F.; Liu, J.; and Jia, J. 2023. Spherical transformer for lidar-based 3d recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 17545–17555.
- Lai, X.; Liu, J.; Jiang, L.; Wang, L.; Zhao, H.; Liu, S.; Qi, X.; and Jia, J. 2022. Stratified transformer for 3d point cloud segmentation. In *CVPR*, 8500–8509.
- Li, K.; Li, X.; Wang, Y.; He, Y.; Wang, Y.; Wang, L.; and Qiao, Y. 2024a. VideoMamba: State Space Model for Efficient Video Understanding. *arXiv:2403.06977*.
- Li, L.; Wang, H.; Zhang, W.; and Coster, A. 2024b. Stg-mamba: Spatial-temporal graph learning via selective state space model. *arXiv preprint arXiv:2403.12418*.
- Liang, D.; Zhou, X.; and Bai, X. 2024. PointMamba: A Simple State Space Model for Point Cloud Analysis. *arXiv preprint arXiv:2402.10739*.
- Lin, H.; Zheng, X.; Li, L.; Chao, F.; Wang, S.; Wang, Y.; Tian, Y.; and Ji, R. 2023. Meta architecture for point cloud analysis. In *CVPR*, 17682–17691.
- Liu, J.; Yang, H.; Zheng, H.; and Wang, S. 2024a. Swin-UMamba: Mamba-based UNet with ImageNet-based pre-training. *arXiv preprint arXiv:2402.03302*.
- Liu, Y.; Tian, Y.; Ye, Q.; and Liu, Y. 2024b. VMamba: Visual State Space Model. *arXiv:2401.10166*.

- Liu, Z.; Lin, Y.; Cao, Y.; Lin, S.; and Guo, B. 2021. Swin transformer: Hierarchical vision transformer using shifted windows. In *ICCV*, 10012–10022.
- Liu, Z.; Yang, X.; Tang, H.; Yang, S.; and Han, S. 2023. FlatFormer: Flattened window attention for efficient point cloud transformer. In *CVPR*, 1200–1211.
- Loshchilov, I.; and Hutter, F. 2017. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*.
- Lu, J.; Deng, J.; and Wang, C. 2023. Query refinement transformer for 3d instance segmentation. In *CVPR*.
- Maturana, D.; and Scherer, S. 2015. Voxnet: A 3d convolutional neural network for real-time object recognition. In *IROS*, 922–928. IEEE.
- Morton, G. M. 1966. A computer oriented geodetic data base and a new technique in file sequencing. *physics of plasmas*.
- Nordin, A.; and Telles, A. 2023. Comparing the Locality Preservation of Z-order Curves and Hilbert Curves.
- Pang, Y.; Wang, W.; Tay, F. E.; Liu, W.; Tian, Y.; and Yuan, L. 2022. Masked autoencoders for point cloud self-supervised learning. In *ECCV*, 604–621. Springer.
- Park, K.-B.; Kim, M.; and Lee, J. Y. 2020. Deep learning-based smart task assistance in wearable augmented reality. *Robotics and Computer-Integrated Manufacturing*, 101887.
- Peano, G.; and Peano, G. 1990. *Sur une courbe, qui remplit toute une aire plane*. Springer.
- Pei, X.; Huang, T.; and Xu, C. 2024. EfficientVMamba: Atrous Selective Scan for Light Weight Visual Mamba. *arXiv preprint arXiv:2403.09977*.
- Qi, C. R.; Su, H.; Mo, K.; and Guibas, L. J. 2017a. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *CVPR*, 652–660.
- Qi, C. R.; Yi, L.; Su, H.; and Guibas, L. J. 2017b. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. *NeurIPS*.
- Qian, G.; Li, Y.; Elhoseiny, M.; and Ghanem, B. 2022. Pointnext: Revisiting pointnet++ with improved training and scaling strategies. *NeurIPS*, 23192–23204.
- Radford, A.; Salimans, T.; Sutskever, I.; et al. 2018. Improving language understanding by generative pre-training.
- Raghu, M.; Unterthiner, T.; Kornblith, S.; Zhang, C.; and Dosovitskiy, A. 2021. Do vision transformers see like convolutional neural networks? *NeurIPS*, 34: 12116–12128.
- Rozenberszki, D.; and Dai, A. 2022. Language-grounded indoor 3d semantic segmentation in the wild. In *ECCV*.
- Seita, D.; Wang, Y.; Shetty, S. J.; Erickson, Z.; and Held, D. 2023. Toolflownet: Robotic manipulation with tools via predicting tool flow from point clouds. In *Conference on Robot Learning*, 1038–1049. PMLR.
- Thomas, H.; Qi, C. R.; Goulette, F.; and Guibas, L. J. 2019. Kpconv: Flexible and deformable convolution for point clouds. In *ICCV*, 6411–6420.
- Touvron, H.; Lavril, T.; Izacard, G.; Azhar, F.; et al. 2023. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*.
- 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*.
- Vaswani, A.; Shazeer, N.; Kaiser, Ł.; and Polosukhin, I. 2017. Attention is all you need. *NeurIPS*, 30.
- Wang, C.; Tsepa, O.; Ma, J.; and Wang, B. 2024a. GraphMamba: 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): 1–11.
- Wang, T.; Wen, W.; Zhai, J.; Xu, K.; and Luo, H. 2024b. Serialized Point Mamba: A Serialized Point Cloud Mamba Segmentation Model. *arXiv preprint arXiv:2407.12319*.
- Wang, Y.; Sun, Y.; Liu, Z.; Sarma, S. E.; Bronstein, M. M.; and Solomon, J. M. 2019. Dynamic graph cnn for learning on point clouds. *TOG*, 38(5): 1–12.
- Wang, Z.; Chen, Z.; Wu, Y.; Zhao, Z.; Zhou, L.; and Xu, D. 2024c. PoinTramba: A Hybrid Transformer-Mamba Framework for Point Cloud Analysis. *arXiv:2405.15463*.
- Wu, H.; and Zhang, L. 2021. Cvt: Introducing convolutions to vision transformers. In *ICCV*, 22–31.
- Wu, W.; Qi, Z.; and Fuxin, L. 2019. Pointconv: Deep convolutional networks on 3d point clouds. In *CVPR*, 9621–9630.
- Wu, X.; Jiang, L.; Ouyang, W.; He, T.; and Zhao, H. 2024. Point Transformer V3: Simpler Faster Stronger. In *CVPR*.
- Wu, X.; and Zhao, H. 2022. Point transformer v2: Grouped vector attention and partition-based pooling. *NeurIPS*.
- Wu, Z.; and Xiao, J. 2015. 3d shapenets: A deep representation for volumetric shapes. In *Proceedings of the IEEE conference on computer vision and pattern recognition*.
- Yang, Y.; Xing, Z.; and Zhu, L. 2024. Vivim: a Video Vision Mamba for Medical Video Object Segmentation. *arXiv preprint arXiv:2401.14168*.
- Yang, Y.-Q.; Guo, Y.-X.; Tong, X.; and Guo, B. 2023. Swin3d: A pretrained transformer backbone for 3d indoor scene understanding. *arXiv preprint arXiv:2304.06906*.
- Yi, L.; Kim, V. G.; Sheffer, A.; and Guibas, L. 2016. A scalable active framework for region annotation in 3d shape collections. *ToG*, 35(6): 1–12.
- Zhang, C.; Wan, H.; Shen, X.; and Wu, Z. 2022. Patchformer: An efficient point transformer with patch attention. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 11799–11808.
- Zhang, T.; Li, X.; Yuan, H.; Ji, S.; and Yan, S. 2024. Point Cloud Mamba: Point Cloud Learning via State Space Model. *arXiv preprint arXiv:2403.00762*.
- Zhao, H.; and Koltun, V. 2021. Point transformer. In *ICCV*.
- Zhou, H.; Zhu, X.; Song, X.; Ma, Y.; Wang, Z.; and Lin, D. 2020. Cylinder3d: An effective 3d framework for driving-scene lidar semantic segmentation. *arXiv:2008.01550*.
- Zhu, L.; Liao, B.; Zhang, Q.; Wang, X.; Liu, W.; and Wang, X. 2024. Vision Mamba: Efficient Visual Representation Learning with Bidirectional State Space Model. *arXiv:2401.09417*.