ZigzagPointMamba: Spatial-Semantic Mamba for Point Cloud Understanding

Linshuang Diao[1] **Sensen Song**[1*] **Yurong Qian**[2] **Dayong Ren**[3†]

[1]Key Laboratory of Signal Detection and Processing, Xinjiang University
[2]Joint International Research Laboratory of Silk Road Multilingual Cognitive Computing, Xinjiang University
[3]Department of Computer Science and Technology, Nanjing University

{107552304043,songsensen}@stu.xju.edu.cn, qyr@stu.xju.edu.cn, rdyedu@gmail.com

Abstract

State Space models (SSMs) like PointMamba provide efficient feature extraction for point cloud self-supervised learning with linear complexity, surpassing Transformers in computational efficiency. However, existing PointMamba-based methods rely on complex token ordering and random masking, disrupting spatial continuity and local semantic correlations. We propose **ZigzagPointMamba** to address these challenges. The key to our approach is a simple zigzag scan path that globally sequences point cloud tokens, enhancing spatial continuity by preserving the proximity of spatially adjacent point tokens. Yet, random masking impairs local semantic modeling in self-supervised learning. To overcome this, we introduce a Semantic-Siamese Masking Strategy (SMS), which masks semantically similar tokens to facilitate reconstruction by integrating local features of original and similar tokens, thus overcoming dependence on isolated local features and enabling robust global semantic modeling. Our pre-training ZigzagPointMamba weights significantly boost downstream tasks, achieving a 1.59% mIoU gain on ShapeNetPart for part segmentation, a 0.4% higher accuracy on ModelNet40 for classification, and 0.19%, 1.22%, and 0.72% higher accuracies respectively for the classification tasks on the OBJ-BG, OBJ-ONLY, and PB-T50-RS subsets of ScanObjectNN.

1 Introduction

Deep learning-based point cloud analysis requires models to extract and interpret intricate geometric features from unstructured spatial data. Traditional approaches, such as MLP[22, 23] and Transformer architecture[24, 11], are often hindered by high computational complexity, substantial resource demands, and limited generalization across diverse datasets. To address these challenges, PointMamba[16] has emerged as a novel framework. By leveraging a state-space model, PointMamba achieves linear computational complexity and robust global feature aggregation, effectively balancing architectural simplicity with efficiency. Its strong knowledge transfer abilities, especially in self-supervised learning, have shown excellent performance and driven significant research into PointMamba-based models.

Despite the efficiency of PointMamba-based methods in global feature aggregation through state-space models, their dependence on conventional scanning schemes—such as random, Hilbert, or Z-order approaches—often disrupts spatial continuity, leading to suboptimal performance in downstream

*Corresponding author
†Corresponding author

39th Conference on Neural Information Processing Systems (NeurIPS 2025).

tasks like point cloud segmentation and classification. For instance, random scanning fragments local geometric coherence, while Hilbert and Z-order schemes struggle to adapt to complex topologies, resulting in disjointed token sequences that impair feature consistency. To address this, we propose ZigzagPointMamba, a novel method that employs a zigzag scanning path to globally sequence point cloud tokens while preserving spatial proximity. By generating smoother, spatially coherent token sequences, ZigzagPointMamba enhances the quality of feature representations, with preliminary experiments demonstrating a 1.59% mIoU improvement in part segmentation on ShapeNetPart and a 0.4% accuracy gain in classification on ModelNet40.

Building on the spatially coherent token sequences provided by ZigzagPointMamba, we further enhance PointMamba's self-supervised learning capabilities by addressing limitations in its masking strategy. Traditional random masking, which relies on adjacent tokens to reconstruct masked regions, struggles to capture global semantic dependencies, compromising performance in downstream point cloud tasks such as segmentation and classification. To overcome this, we introduce the Semantic-Siamese Masking Strategy (SMS), which leverages the smooth token sequences from ZigzagPointMamba to mask semantically similar local structures, thereby strengthening token-level semantic associations. By integrating local features of original and semantically related tokens, SMS enables robust global semantic modeling, yielding more accurate feature representations. Preliminary results validate the synergy of ZigzagPointMamba and SMS, achieving up to 1.22% accuracy improvements across classification on ScanObjectNN subsets (OBJ-BG, OBJ-ONLY, PB-T50-RS).

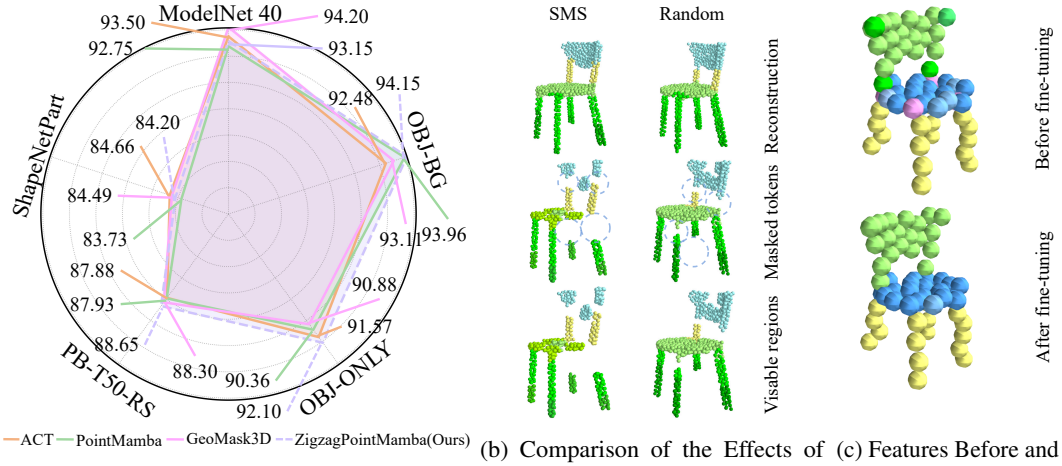
SMS employs a Siamese-like comparison to evaluate semantic similarity between point cloud tokens, enabling targeted masking of coherent structures, such as entire object parts like a chair's armrest or a car's wheel, rather than random token selections. By leveraging smooth, spatially continuous tokens, SMS ensures that masked semantic tokens maintain both spatial proximity and semantic continuity, preserving the topological integrity of the point cloud during self-supervised learning. Specifically, SMS operates through two key steps: (1) The point cloud is decomposed into fine-grained semantic tokens by leveraging the zigzag scanning order, and (2) A Siamese-like mechanism evaluates token-wise similarity scores, and tokens exceeding a predefined similarity threshold are masked. This strategy eliminates redundant local features, forcing the model to reconstruct masked regions by relying on global semantic context from retained tokens. This strategy mitigates the limitations of traditional random masking, which relies solely on local information and disrupts semantic coherence, thereby optimizing global feature modeling and enhancing self-supervised learning capabilities. Combined with ZigzagPointMamba's spatial advantages, SMS achieves superior reconstruction quality and more distinct feature distributions after fine-tuning, Our **ZigzagPointMamba** achieves excellent performance on various point cloud analysis datasets (as shown in Fig 1 (a).). In addition, the reconstruction effect of SMS we proposed is better, and the feature distribution after fine-tuning is also more distinct (as shown in Fig.1 (b) and Fig.1 (c)).

Collectively, our contributions include: (1) a zigzag scan path that preserves spatial proximity, mitigating discontinuities in traditional scanning methods; (2) the SMS approach, which enhances token-level semantic associations for robust global feature modeling; and (3) the integration of spatial and semantic continuity in ZigzagPointMamba, significantly advancing point cloud analysis, particularly in self-supervised learning applications.

2 Related Works

2.1 Point Cloud Analysis Methods Based on MLP

Early deep learning methods for point cloud processing[25, 10, 27], such as PointNet[22], used shared multi-layer perceptrons (MLPs) for feature extraction and max pooling to aggregate global information. However, these methods had limitations in modeling local geometric structures. PointNet++[23] improved this by introducing hierarchical MLPs and farthest point sampling (FPS) for multi-scale feature learning. Subsequently, methods like RandLA-Net[12] proposed lightweight architectures that processed large-scale point clouds through random sampling and local feature aggregation mechanisms, significantly enhancing computational efficiency. PointMLP[20] constructed networks purely based on residual MLPs and introduced a local geometric affine module, achieving certain results in point cloud tasks and providing a network architecture foundation for the application of subsequent self-supervised learning methods in point cloud processing. Later, [39] enhanced the MLP architecture through efficient addition and shift operations, reducing computational complexity while



(a) Comparision of Performance (b) Comparison of the Effects of SMS and Random Masking (c) Features Before and After Fine-tuning

Figure 1: As can be seen from Fig. 1 (a), compared with ACT, PointMamba, and GeoMask3D, our proposed ZigzagPointMamba performs better on the ScanObjectNN dataset. Fig. 1 (b) presents a stark contrast between the effects of SMS and random masking, highlighting the superiority of our proposed method in terms of reconstruction. Fig. 1 (c) demonstrates the features before and after fine-tuning, indicating the effectiveness of our method in refining feature representations.

improving point cloud classification performance. HPE[41] improved MLPs using high-dimensional positional encoding, mapping the 3D coordinates of points to a high-dimensional space to effectively enhance the representation of point cloud positional information and further boost the MLP's ability to model local structures. PointMT[38] combines the efficiency of MLPs with the global feature capture capability of Transformers, and introduces innovative mechanisms to address the computational bottlenecks of traditional Transformers, enabling efficient point cloud analysis.

2.2 The Deep Evolution of Transformer Architecture and Masking Strategies

In recent years, Transformer-based point cloud processing methods have made synergistic progress in both architectural design and masking strategies. PCT[9] first introduced self-attention mechanisms into point cloud processing, while Point Transformer V1[37] enhanced geometric modeling through vector attention. Subsequently, Point Transformer V2[34] and OctFormer[32] improved computational efficiency via hierarchical attention and octree partitioning, respectively. GPSFormer[31] and DAPoinTr[15] have achieved innovative integration of graph structures and dynamic path mechanisms. Meanwhile, the evolution of masking strategies is deeply intertwined with Transformer architecture development: Point-BERT[35] was the first to adapt the masked pre-training paradigm from natural language processing to point clouds, establishing a self-supervised learning framework through coordinate reconstruction tasks. Point-MAE[21] further strengthened feature learning by increasing the masking ratio, while Point-M2AE[36] introduced a multi-scale masking framework that synergizes with Transformer's hierarchical architecture. Methods such as GeoMAE[29] incorporate geometric[5] awareness into masking design, enabling models to better understand spatial layouts while maintaining computational efficiency.

2.3 The Development of State-Space Models in Point Cloud Processing

Due to the quadratic complexity of Transformers, State Space Models (SSMs) have become a research hotspot for their linear computational complexity[26]. The Mamba model[8] demonstrated the high-efficiency potential of SSMs in sequence processing and proposed an effective linear-time sequence modeling method. In further research, the attention mechanism of the Mamba model was explored, revealing how to enhance the capture of key information while maintaining efficient computation[1]. Subsequently, many Mamba-based variants have been applied to various fields[18, 33, 33]. For example, U-Mamba[19] enhances long-range dependency modeling in biomedical image segmentation, and LocalMamba[14] further optimizes the performance of visual State Space Models

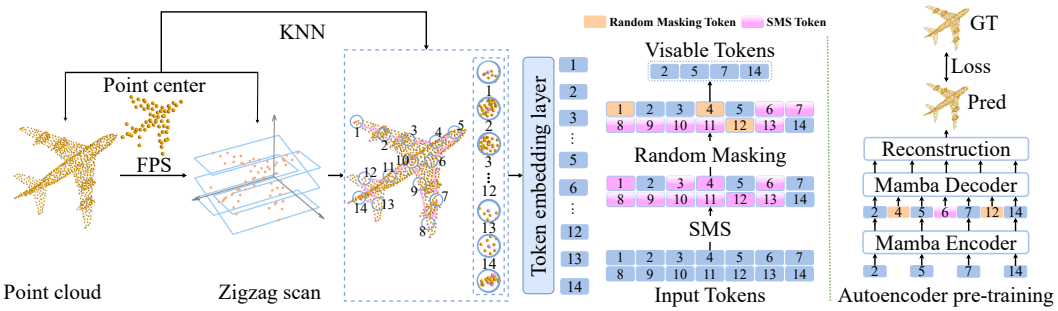


Figure 2: ZigzagPointMamba pre-training pipeline. Select key point cloud points with FPS. Extract feature labels via KNN algorithm and lightweight PointNet. Serialize using the zigzag scan path. Input serialized features into a point cloud MAE architecture with SMS for training, obtaining point cloud feature representations and providing parameters for downstream tasks.

by introducing windowed selective scanning. However, these models still face significant challenges when processing complex geometric structure data such as point clouds.

PointMamba[16] was the first to apply the SSM framework to point cloud processing, achieving remarkable progress in feature extraction. However, its scan method may hinder the capture of geometric structures, while random masking may obscure key information. To address these issues, we propose the ZigzagPointMamba model. By combining the zigzag scan path and the Semantic-Siamese Masking Strategy (SMS), ZigzagPointMamba ensures that spatially adjacent points maintain proximity and effectively learns global semantics.

3 Methods

3.1 Background: State Space Models and PointMamba

State Space Models (SSMs) serve as a classical framework for processing sequential and spatial data,capturing long-range dependencies through a recursive state transition equation:

$$h_t = Ah_{t-1} + Bx_t, \tag{1}$$

where h_t is the state at time step t, x_t is the observation (or input) at step t, and A, B are learned transition matrices. This model exhibits $O(n)$ linear computational complexity, making it highly efficient for large-scale data modeling.

In the field of point cloud analysis, PointMamba adapts the SSM framework to process unordered point clouds by treating each point as an independent state unit. The modified state transition is:

$$h_t = A_t h_{t-1} + B_t x_t, \tag{2}$$

where $h_t \in \mathbb{R}^d$ represents the state of point,$x_t \in \mathbb{R}^m$ contains the point's 3D coordinates and features, A_t, B_t are dynamic parameters conditioned on x_t.

3.2 The structure of ZigzagPointMamba

We present **ZigzagPointMamba**, an enhanced architecture derived from PointMamba that introduces two key innovations: (1) a novel **zigzag scan path** for structured point cloud traversal, and (2) a **Semantic-Siamese Masking Strategy (SMS)** for enhanced token-level semantic associations for robust global feature modeling.

In the pre-training process of the ZigzagPointMamba architecture (Fig. 2), input point cloud data first undergoes the zigzag scan path. Key points are selected via Farthest Point Sampling (FPS), followed by zigzag scan applied on XY, XZ, and YZ planes. Hierarchical layering and alternating sorting generate traversal paths with clear spatial logic, transforming unordered point clouds into ordered sequences to enhance feature spatial proximity and continuity. As illustrated, feature vectors of adjacent points in the ordered sequence better reflect their 3D adjacency, establishing a foundation for capturing local geometric features. Subsequently, features are extracted from these paths using KNN

and a lightweight PointNet to generate point tokens. The SMS calculates semantic similarity from token feature vectors, identifies redundant regions via thresholding, and applies masking to prevent the model from over-relying on local information, thereby enhancing the capture of long-range semantic relationships.

To process 3D point cloud data and assist deep learning models in capturing spatial structure, we use an improved zigzag scan path approach. This approach generates structured traversal paths on orthogonal planes in 3D space. In traditional 2D image processing, the 2D zigzag scan path[13] is a commonly used method. As shown in the left sub-figure of Fig. 4, the 2D zigzag scan path traverses data points in a specific zigzag pattern, effectively organizing the data order in 2D space. However, when dealing with 3D point cloud data, the scenario becomes more complex. A 3D point cloud can be represented as $P = \{p_i\}_{i=1}^{N}$, where $p_i = (x_i, y_i, z_i) \in \mathbb{R}^3$. Our 3D zigzag scan path extends the 2D approach by performing scanning operations on three key planes: the XY-plane, XZ-plane, and YZ-plane, to comprehensively analyze the spatial relationships within 3D point clouds. In implementation, the point cloud is first divided into layers along different coordinate axes through coordinate-based layering; subsequently, segment-based alternating sorting is used to construct traversal scan paths with specific zigzag patterns on each of the three key planes. The effect is illustrated in Fig. 3.

Scan generation on the XY-Plane First, we sort all points in ascending order based on the Z-coordinate. This step arranges the point cloud according to the values of Z, which facilitates the subsequent layering operation. After sorting, the sorted point cloud is divided into L_{xy} layers, where $L_{xy} = \lceil \frac{M}{3} \rceil$. For the k-th layer, denoted as L_k^{xy}, it is represented as:

$$L_k^{xy} = \left\{ p_i \in P \mid z_{(k-1)\frac{N}{L_{xy}}} \leq z_i < z_{k\frac{N}{L_{xy}}} \right\}, \quad k = 1, 2, \ldots, L_{xy}. \tag{3}$$

Here, $z_{(k-1)\frac{N}{L_{xy}}}$ and $z_{k\frac{N}{L_{xy}}}$ are the values of the Z-coordinates at the corresponding positions in the sorted point cloud. By this layering, the point cloud is divided along the Z-direction, ensuring that points within the same layer are similar in their Z-coordinates.

Within each layer L_k^{xy}, the points are first sorted in ascending order based on their X-coordinates. This operation organizes the points along the X-direction within each layer. Next, the points are divided into S segments, where $S = \min\left(\left\lfloor \frac{|L_k^{xy}|}{d} \right\rfloor, m\right)$, ensuring that each segment contains approximately d points. For the s-th segment S_s^{xy}, the points are alternately sorted by the Y-coordinate: When s is even, the points are sorted in ascending order of Y. That is, for $p_i, p_j \in S_s^{xy}$, if $y_i < y_j$, then p_i comes before p_j. When s is odd, the points are sorted in descending order of Y. That is, for $p_i, p_j \in S_s^{xy}$, if $y > y_i$, then p_i comes before p_j. This alternating sorting creates a zigzag scan path within each layer. Finally, the segments are connected in sequence to form a scan R_k^{xy}.

Scan generation on the XZ-Plane Similar to the XY-plane, points are sorted ascendingly by the Y-coordinate first. The point cloud is divided into L_{xz} layers, where $L_{xz} = \lfloor \frac{M}{3} \rfloor + I_{M \bmod 3 \geq 1}$. For the k-th layer L_k^{xz}, it is defined as $L_k^{xz} = \left\{ p_i \in P \mid y_{(k-1)\frac{N}{L_{xz}}} \leq y_i < y_{k\frac{N}{L_{xz}}} \right\}, k = 1, \cdots, L_{xz}$. Points in each layer are sorted by X-coordinate, segmented, and sorted by Z-coordinate to form R_k^{xz}.

Scan generation on the YZ-Plane Likewise, points are sorted by the X-coordinate first. The point cloud is divided into L_{yz} layers, where $L_{yz} = \lfloor \frac{M}{3} \rfloor$. For the k-th layer L_k^{yz}, it is defined as $L_k^{yz} = \left\{ p_i \in P \mid x_{(k-1)\frac{N}{L_{yz}}} \leq x_i < x_{k\frac{N}{L_{yz}}} \right\}, k = 1, \cdots, L_{yz}$. After sorting points in each layer by Y-coordinate, segmenting, and sorting by Z-coordinate, we get R_k^{yz}.

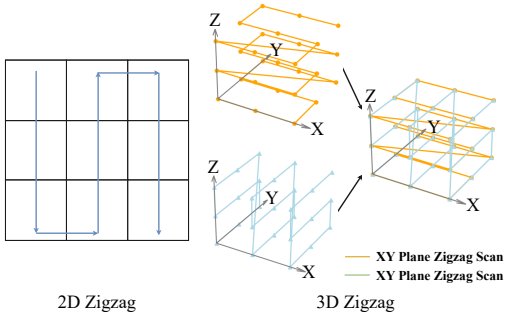


Figure 3: Comparison of 2D and 3D zigzag. The 3D strategy scans on multiple planes. As an extension of the 2D one, it aids the model in preserving spatial proximity.

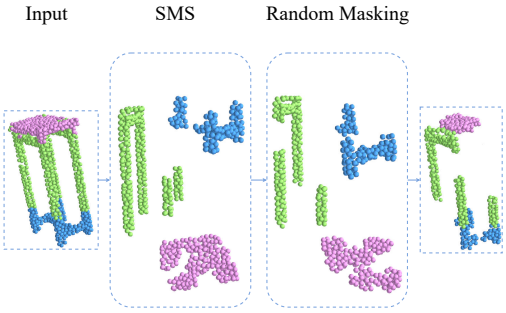


Figure 4: Details of Masking. Leverage SMS to mask out tokens with high semantic feature similarity in the point cloud. Then, apply random masking to a subset of the remaining tokens to enhance the robustness of the pre-training model.

3.3 Masking Process

The traditional random masking strategy is still prone to losing key information and disrupting the semantic coherence and geometric structural relationships that have been sorted out when masking the point cloud data processed by the zigzag scan path. Although the point cloud tokens arranged by the zigzag scan path ensure the continuity of spatially adjacent features, the random masking fails to make good use of this advantage. To address this issue, we introduce the SMS. Based on the semantic similarity of point cloud features, the SMS selects the masking regions. By fully leveraging the structured foundation established by the zigzag scan, it can not only preserve the important semantic information and geometric structure but also enhance the model's ability to model global semantic features, significantly improving the performance of the MAE in handling downstream tasks.

3.3.1 Semantic-Siamese Masking Strategy (SMS)

Algorithm 1 Semantic-Siames Masking Strategy

Input: $group\_input\_tokens$: point cloud feature tensor with shape B, G, C (batch_size, number of groups, feature_dimension).
$threshold$: SMS retention $threshold$ (default 0.8), controlling the proportion of tokens to retain.
Output: $bool\_masked\_pos$: boolean mask tensor with shape B, G, C, indicating which tokens are masked.

$B, G, C \leftarrow$shape($group\_input\_tokens$) // Get tensor dimensions
$tokens\_norm \leftarrow$ F.normalize($group\_input\_tokens$, dim $= -1$) // Normalize feature vectors to unit length
$similarity\_matrix \leftarrow$ torch.bmm($tokens\_norm, tokens\_norm^T$).clamp$(0, 1)$ // Compute cosine similarity matrix and clamp to [0,1]
$redundancy\_score \leftarrow \sum_{\text{dim}=-1}(similarity\_matrix)$ // Calculate redundancy score for each token
$k \leftarrow \max(1, \lfloor threshold \times G \rfloor)$ // Determine number of tokens to retain (at least 1)
if $k = 0$
    return torch.zeros($[B, G]$, dtype $=$ torch.bool)
$thresholds \leftarrow$ torch.topk($redundancy\_score, k = k$, largest $=$ torch.False)$.values[:, -1]$ // Get k-th smallest redundancy score as threshold
$bool\_masked\_pos \leftarrow redundancy\_score > thresholds$ // Generate mask (tokens with higher redundancy are masked)
return $bool\_masked\_pos$

The main goal of SMS is to remove the parts of the input data that are irrelevant or redundant to the task, keeping the most representative and informative tokens (As shown in Algorithm1, its effect is presented in Fig. 1 (b) and Fig. 4). This process takes place during the self-supervised learning phase of ZigzagPointMamba.First, for the input T (with shape $B \times G \times C$), we normalize the tokens to ensure that the features of each token are transformed into unit vectors, thus eliminating scale differences between different feature dimensions. The specific formula is:

$$T_{norm_{b,g,c}} = \frac{T_{b,g,c}}{\sqrt{\sum_{c'=0}^{C-1} T_{b,g,c'}^2}}, \tag{4}$$

where $b \in \{1, \cdots, B\}$, $g \in \{1, \cdots, G\}$, and $c \in \{1, \cdots, C\}$. This step converts the feature vector of each token into a unit vector, ensuring that the features have a uniform scale.

Next, we compute the cosine similarity between the normalized feature vectors, resulting in a similarity matrix. The similarity between two tokens i and j in batch b is calculated using the following formula:

$$S_{b,i,j} = \frac{f_{b,i} \cdot f_{b,j}}{\|f_{b,i}\| \|f_{b,j}\|}, \tag{5}$$

where $f_{b,i}$ and $f_{b,j}$ are the normalized feature vectors of the tokens i-th and j-th in batch b-th respectively. Since we have already normalized the feature vectors, $\|f_{b,i}\| = 1$ and $\|f_{b,j}\| = 1$,so the formula simplifies to:

$$S_{b,i,j} = \sum_{c=0}^{C-1} T_{norm_{b,i,c}} \cdot T_{norm_{b,j,c}}. \tag{6}$$

Also, we limit the values of the similarity matrix to the range $[0, 1]$ via the following transformation:$S_{b,i,j} = \max(0, \min(1, S_{b,i,j}))$. This matrix represents the similarity between every pair of tokens, where i and j are indices of two different tokens. Then, we sum each row of the similarity matrix to obtain the redundancy score for each token, as shown in the following formula:

$$R_{b,i} = \sum_{j=0}^{G-1} S_{b,i,j}. \tag{7}$$

This redundancy score reflects how similar each token is to the other tokens. A higher redundancy score indicates that the token is more redundant. Based on the semantic threshold $t_{semantic}$, we calculate the number of tokens that need to be retained, denoted as $k = \lfloor t_{semantic} \cdot G \rfloor$.Next, we sort the redundancy scores and choose the k-th smallest redundancy score as the threshold for each batch:

$$t_b = \text{topk}(R_b, k = k, largest = \text{False})[k-1]. \tag{8}$$

This threshold is used to identify redundant tokens, which are marked for masking, generating the initial semantic mask. The formula for mask generation is:

$$M_{semantic_{b,i}} = \begin{cases} \text{True}, & \text{if } R_{b,i} > t_b, \\ \text{False}, & \text{otherwise.} \end{cases} \tag{9}$$

Tokens with redundancy scores exceeding the threshold are marked as True for masking.

3.3.2 Random Mask Generation

After generating the SMS, we proceed with random masking of the remaining tokens. The main purpose of this stage is to further reduce redundancy by introducing randomness and to avoid overfitting of the model. When masking is required, we randomly select a certain percentage of tokens from those that have not been masked by the SMS. The number of tokens to be masked is calculated using the following formula:$N_{mask_b} = \lfloor R_{mask} \cdot |I_{available_b}| \rfloor$.Then, we randomly choose the indices of these available tokens and mark them as masked:$M_{final_{b,I_{mask_b}}} = \text{True}$.

4 Experiments

We present in this section the experimental configuration, datasets, and evaluation results for assessing the performance of **ZigzagPointMamba**. Our experimental framework incorporates the newly proposed zigzag scan path and SMS. Extensive evaluation across multiple benchmark datasets on various

Table 1: Object Classification on ScanObjectNN Dataset. We conducted experiments on three subsets of the ScanObjectNN dataset: the OBJ-BG subset, OBJ-ONLY subset, and PB-T50-RS subset.

Methods	Reference	Param.(M)	FLOPs(G)	OBJ-BG	OBJ-ONLY	PB-T50-RS
Point-Bert[35]	CVPR 22	22.1	4.8	87.3	88.12	83.07
MaskPoint[17]	CVPR 22	22.1	4.8	89.70	89.30	84.60
PointMAE[21]	ECCV 22	22.1	4.8	90.02	88.29	84.60
PointM2AE[36]	NeurIPS 22	15.3	3.6	91.22	88.81	86.43
ACT[7]	ICLR 23	22.1	4.8	93.29	91.91	88.21
ReCon[24]	ICML 23	43.6	5.3	94.15	93.12	89.73
GeoMask3D[2]	TMLR 25	-	-	93.11	90.36	88.30
PointMamba([16])(baseline)	NeurIPS 24	**12.3**	**3.1**	93.96	90.88	87.93
ZigzagPointMamba(Ours)		**12.3**	**3.1**	**94.15**	**92.10**	**88.65**

downstream tasks demonstrates that our approach achieves superior performance improvements over existing methods in terms of classification accuracy, segmentation performance, and model robustness.

4.1 Pre-training Setup

In the pre-training phase [35, 40] , self-supervised learning extracts generalizable representations from unannotated point cloud data, providing transferable parameters for downstream tasks. We introduce the zigzag scan path and SMS to boost the model's local structure capture. To adapt to different resolutions, point clouds are patch-divided linearly (1024 points into 64 patches with 32 points per patch via KNN). The encoder has 12 vanilla Mamba blocks (384 dims each), and the decoder uses 4 Mamba blocks for reconstruction. We randomly select one path, setting the random masking ratio at 0.6 and SMS ratio at 0.8. Training uses AdamW optimizer (lr = 0.001), Cosine annealing scheduler, 300 epochs, batch size 128, and loss of Chamfer distance L2 [3]. Experiments are run on a NVIDIA A40 GPU.

Table 2: Classification on ModelNet40 Dataset. We report the overall accuracy from 1024 points without voting.

Methods	Reference	Param.(M)	FLOPs(G)	OA(%)
Point-Bert[35]	CVPR 22	22.1	4.8	92.7
MaskPoint[17]	CVPR 22	22.1	4.8	92.6
PointMAE[21]	ECCV 22	22.1	4.8	93.2
PointM2AE[36]	NeurIPS 22	15.3	3.6	93.4
ACT[7]	TCLR 23	22.1	4.8	93.6
GeoMask3D[2]	TMLR 25	-	-	94.20
PointMamba([16])(baseline)	NeurIPS 24	12.3	1.5	92.75
ZigzagPointMamba(Ours)		**12.3**	**1.5**	**93.15**

Table 3: Part Segmentation on ShapeNetPart Dataset. The mIoU for all classes (Cls.) and for all instances (Inst.) are reported.

Methods	Reference	Inst.mIoU	Cls.mIoU
Point-BERT	TMLR 25	85.6	84.1
MaskPoint[17]	TMLR 25	86.0	84.4
PointMAE[21]	ECCV 22	86.1	84.1
PointM2AE[36]	NeurIPS 22	86.51	84.86
ACT[7]	ICLR 23	86.14	84.66
GeoMask3D[2]	TMLR 25	86.04	84.49
PointMamba([16])(baseline)	NeurIPS 24	85.28	82.57
ZigzagPointMamba(Ours)		**85.78**	**84.16**

4.2 Datasets and Performance Evaluation of Downstream Tasks

Table 4: Few-shot learning on ModelNet40. A dedicated dataset for few-shot learning constructed based on ModelNet40.

Methods	Reference	5-way		10-way	
		10-shot	20-shot	10-shot	20-shot
Point-Bert[35]	CVPR 22	94.6±3.0	96.3±2.5	91.0±5.0	92.7±4.8
MaskPoint[17]	CVPR 22	95.0±3.7	97.2±1.5	91.4±4.5	93.4±3.2
PointMAE[21]	ECCV 22	96.3±3.1	97.8±1.8	92.6±4.0	95.0±2.8
PointM2AE[36]	NeurIPS 22	96.8±2.0	98.3±1.5	92.3±4.2	95.2±2.5
ACT[7]	ICLR 23	96.8±2.1	98.0±1.5	93.3±4.0	95.6±3.0
PointGPT-S[6]	NeurIPS 23	96.8±1.8	98.6±1.2	92.6±3.5	95.2±2.5
ReCon[24]	ICML 23	97.3±1.8	98.0±1.5	93.3±4.3	95.8±2.8
PointMamba([16])(baseline)	NeurIPS 24	**96.0±2.0**	**99.0±1.0**	88.5±2.4	84.0±1.8
ZigzagPointMamba(Ours)		**96.0±2.1**	**99.0±1.2**	**90.0±2.2**	**84.5±1.5**

Table 5: The effect of the thresholds of different SMS.

Setting	OA(%)	
	OBJ - ONLY	PB - T50 - RS
0.6/0.5	90.71	87.99
0.6/0.6	91.57	87.68
0.6/0.7	91.22	88.45
0.6/0.8	**92.08**	**88.65**
0.6/0.9	91.91	88.36

ModelNet40: In the classification experiment on ModelNet40 [28]. As shown in Table 2, we conducted experiments without using the voting strategy. The classification accuracy of ZigzagPointMamba reached 93.15%, representing an 0.4% increase compared to PointMamba (Please note that the data of PointMamba in the experimental tables are obtained from our own experiments). The ModelNetFewShot dataset, constructed based on ModelNet40, is specifically designed for few-shot learning research. The selection of few-shot data in it strictly follows the following rules: randomly select 5 or 10 categories from the 40 categories of ModelNet40 as the task categories, and then randomly select 10 or 20 samples from each selected category as the support set. We conducted

15 independent few-shot learning experiments on ZigzagPointMamba using the ModelNetFewShot dataset. In the 10-way 10-shot scenario, the average classification accuracy of ZigzagPointMamba increased by 1.5% compared to PointMamba; in the 20-way 20-shot scenario, the average accuracy increased by 0.5%. As shown in Table 4, ZigzagPointMamba also demonstrates promising classification performance when handling few-shot data.

ScanObjectNN: As summarized in Table 1, we comprehensively evaluate **ZigzagPointMamba** across three distinct experimental configurations of ScanObjectNN [30]: OBJ-BG (objects with background), OBJ-ONLY (isolated objects), PB-T50-RS(perturbed objects with 50% translation and random scaling). ZigzagPointMamba demonstrates consistent improvements over the baseline PointMamba in all scenarios: In OBJ-ONLY, ZigzagPointMamba achieves $\mathbf{92.10}\%$ classification accuracy ($\uparrow \mathbf{1.22}\%$ versus PointMamba's 90.88%), highlighting its efficacy in handling clean object geometries. For OBJ-BG with background interference, our method attains $\mathbf{94.15}\%$ accuracy ($\uparrow \mathbf{0.19}\%$ over PointMamba's 93.96%), showcasing robustness to environmental noise. Under the most challenging PB-T50-RS conditions, ZigzagPointMamba maintains $\mathbf{88.65}\%$ accuracy ($\uparrow \mathbf{0.72}\%$ versus 87.93%), validating its resilience to severe geometric perturbations.

ShapeNetPart: As can be seen from Table 3, ZigzagPointMamba performs remarkably well in the segmentation task of ShapeNetPart[4]. Its Inst.mIoU reaches 85.78%, which is 0.5% higher than that of PointMamba. The Cls.mIoU is 84.16%, 1.59% higher than that of PointMamba.(The comparison of its segmentation effects is shown in the supplementary materials.)

4.3 Analysis and ablation study

To deeply explore the specific contributions of the zigzag scan path and SMS to the model performance, we conducted a series of ablation experiments on the OBJ-ONLY and PB-T50-RS datasets and observed the changes in the model performance.

Table 6: The effect of different attention.

Setting	OA(%)	
	OBJ-ONLY	PB-T58-RS
Attention	91.22	88.17
Multi-Attention	90.53	87.79
SMS	**92.08**	**88.51**

Table 7: The effect of different scanning curves.

Scanning curve	OBJ-ONLY	PB-T58-RS
Random	92.60	90.18
Z-order and Trans-Z-order	93.29	90.36
Hilbert and Z-order	93.29	90.88
Trans-Hilert and Trans-Z-order	93.29	91.91
Hilbert and Trans-Hilbert	90.88	87.93
zigzag scan path (Ours)	**92.10**	**88.65**

Influence of Different SMS Thresholds: In the "Setting" column of the Table 5, the paired values represent the random masking ratio and the SMS masking ratio respectively. Among them, when the SMS masking ratio is 0.8 and the random masking ratio is 0.6, the performance is optimal. The accuracies in the OBJ-ONLY and PB-T50-RS settings reach 92.08% and 88.65% respectively. When the threshold is lower than 0.8, redundant information will interfere with the model's learning process, affecting the accuracy of classification and segmentation. For example, when processing samples from the ScanObjectNN dataset, the model is easily interfered by noise features. When the threshold is higher than 0.8, over-masking occurs, resulting in the loss of key information. This makes it difficult for the model to grasp the features and semantic relationships in complex point cloud data, ultimately leading to a decline in performance.

Influence of Different Attention Mechanisms: By comparing different attention mechanisms, we found that when using SMS, the model achieved accuracies of 92.08% and 88.51% in the OBJ-ONLY and PB-T50-RS settings, respectively, which are better than the standard attention mechanism and the multi-attention mechanism, as shown in Table 6. This indicates that SMS can more effectively help the model capture semantic features and strengthen the modeling of long-range semantic relationships. Compared with the standard attention mechanism, SMS performs masking operations based on semantic similarity, which can more accurately screen out the information valuable for the model's learning and reduce the interference of redundant information, thus improving the model's performance in complex tasks. Although the multi-attention mechanism can pay attention to data from multiple perspectives, it is less effective than SMS in focusing on semantic features, resulting in a slightly inferior overall performance.

Influence of Different Scanning Curves: We conducted an in-depth exploration of the impact of different scanning curves on the model performance. As shown in Table 7, the zigzag scan path achieved accuracies of 92.10% and 88.65% in the OBJ-ONLY and PB-T50-RS settings, respectively. Compared with other scanning curve settings, the effect of this result is not particularly outstanding from a data perspective. However, we cannot solely evaluate the pros and cons of the zigzag scan path based on the increase in accuracy. In our actual model, the zigzag scan path is closely integrated with the SMS. The two complement each other and work together to improve the model performance.

5 Conclusion

In this paper, we introduced **ZigzagPointMamba**, an innovative state-space model that addresses critical limitations in existing PointMamba-based approaches for point cloud self-supervised learning. Through extensive experiments on multiple benchmark datasets, we demonstrated that our approach improves classification accuracy, segmentation performance, and robustness, especially under noisy and occluded conditions. Our results show that the zigzag scan path preserves spatial continuity in point clouds, while the SMS helps the model focus on global structures, preventing over-reliance on local features. Overall, ZigzagPointMamba provides a powerful pre-trained backbone that effectively supports downstream point cloud analysis tasks, offering a practical and robust foundation for a wide range of applications.

References

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

[2] Ali Bahri, Moslem Yazdanpanah, Mehrdad Noori, Milad Cheraghalikhani, Gustavo Adolfo Vargas Hakim, David Osowiechi, Farzad Beizaee, Ismail Ben Ayed, and Christian Desrosiers. Geomask3d: Geometrically informed mask selection for self-supervised point cloud learning in 3d. *arXiv preprint arXiv:2405.12419*, 2024.

[3] Ainesh Bakshi, Piotr Indyk, Rajesh Jayaram, Sandeep Silwal, and Erik Waingarten. Near-linear time algorithm for the chamfer distance. *Advances in Neural Information Processing Systems*, 36:66833–66844, 2023.

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

[5] Chen Chen, Yisen Wang, Honghua Chen, Xuefeng Yan, Dayong Ren, Yanwen Guo, Haoran Xie, Fu Lee Wang, and Mingqiang Wei. Geosegnet: point cloud semantic segmentation via geometric encoder–decoder modeling. *The Visual Computer*, 40(8):5107–5121, 2024.

[6] Guangyan Chen, Meiling Wang, Yi Yang, Kai Yu, Li Yuan, and Yufeng Yue. Pointgpt: Auto-regressively generative pre-training from point clouds. *Advances in Neural Information Processing Systems*, 36:29667–29679, 2023.

[7] Runpei Dong, Zekun Qi, Linfeng Zhang, Junbo Zhang, Jianjian Sun, Zheng Ge, Li Yi, and Kaisheng Ma. Autoencoders as cross-modal teachers: Can pretrained 2d image transformers help 3d representation learning? *arXiv preprint arXiv:2212.08320*, 2022.

[8] Albert Gu and Tri Dao. Mamba: Linear-time sequence modeling with selective state spaces. *arXiv preprint arXiv:2312.00752*, 2023.

[9] Meng-Hao Guo, Jun-Xiong Cai, Zheng-Ning Liu, Tai-Jiang Mu, Ralph R Martin, and Shi-Min Hu. Pct: Point cloud transformer. *Computational visual media*, 7:187–199, 2021.

[10] Yanwen Guo, Yuanqi Li, Dayong Ren, Xiaohong Zhang, Jiawei Li, Liang Pu, Changfeng Ma, Xiaoyu Zhan, Jie Guo, Mingqiang Wei, et al. Lidar-net: A real-scanned 3d point cloud dataset for indoor scenes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 21989–21999, 2024.

[11] Yuan He, Guyue Hu, and Shan Yu. Contrastive semantic-aware masked autoencoder for point cloud self-supervised learning. *IEEE Signal Processing Letters*, 2025.

[12] Qingyong Hu, Bo Yang, Linhai Xie, Stefano Rosa, Yulan Guo, Zhihua Wang, Niki Trigoni, and Andrew Markham. Randla-net: Efficient semantic segmentation of large-scale point clouds. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11108–11117, 2020.

[13] Vincent Tao Hu, Stefan Andreas Baumann, Ming Gui, Olga Grebenkova, Pingchuan Ma, Johannes Fischer, and Björn Ommer. Zigma: A dit-style zigzag mamba diffusion model. In *European Conference on Computer Vision*, pages 148–166. Springer, 2024.

[14] Tao Huang, Xiaohuan Pei, Shan You, Fei Wang, Chen Qian, and Chang Xu. Localmamba: Visual state space model with windowed selective scan. *arXiv preprint arXiv:2403.09338*, 2024.

[15] Yinghui Li, Qianyu Zhou, Jingyu Gong, Ye Zhu, Richard Dazeley, Xinkui Zhao, and Xuequan Lu. Dapointr: Domain adaptive point transformer for point cloud completion. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 5066–5074, 2025.

[16] Dingkang Liang, Xin Zhou, Wei Xu, Xingkui Zhu, Zhikang Zou, Xiaoqing Ye, Xiao Tan, and Xiang Bai. Pointmamba: A simple state space model for point cloud analysis. *arXiv preprint arXiv:2402.10739*, 2024.

[17] Haotian Liu, Mu Cai, and Yong Jae Lee. Masked discrimination for self-supervised learning on point clouds. In *European Conference on Computer Vision*, pages 657–675. Springer, 2022.

[18] Yue Liu, Yunjie Tian, Yuzhong Zhao, Hongtian Yu, Lingxi Xie, Yaowei Wang, Qixiang Ye, Jianbin Jiao, and Yunfan Liu. Vmamba: Visual state space model. *Advances in neural information processing systems*, 37:103031–103063, 2024.

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

[20] Xu Ma, Can Qin, Haoxuan You, Haoxi Ran, and Yun Fu. Rethinking network design and local geometry in point cloud: A simple residual mlp framework. *arXiv preprint arXiv:2202.07123*, 2022.

[21] Yatian Pang, Wenxiao Wang, Francis EH Tay, Wei Liu, Yonghong Tian, and Li Yuan. Masked autoencoders for point cloud self-supervised learning. In *European conference on computer vision*, pages 604–621. Springer, 2022.

[22] Charles R Qi, Hao Su, Kaichun Mo, and Leonidas J Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 652–660, 2017.

[23] Charles Ruizhongtai Qi, Li Yi, Hao Su, and Leonidas J Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. *Advances in neural information processing systems*, 30, 2017.

[24] Zekun Qi, Runpei Dong, Guofan Fan, Zheng Ge, Xiangyu Zhang, Kaisheng Ma, and Li Yi. Contrast with reconstruct: Contrastive 3d representation learning guided by generative pretraining. In *International Conference on Machine Learning*, pages 28223–28243. PMLR, 2023.

[25] Dayong Ren, Jiawei Li, Zhengyi Wu, Jie Guo, Mingqiang Wei, and Yanwen Guo. Mffnet: multimodal feature fusion network for point cloud semantic segmentation. *The Visual Computer*, 40(8):5155–5167, 2024.

[26] Dayong Ren, Zhe Ma, Yuanpei Chen, Weihang Peng, Xiaode Liu, Yuhan Zhang, and Yufei Guo. Spiking pointnet: Spiking neural networks for point clouds. *Advances in Neural Information Processing Systems*, 36, 2024.

[27] Dayong Ren, Zhengyi Wu, Jiawei Li, Piaopiao Yu, Jie Guo, Mingqiang Wei, and Yanwen Guo. Point attention network for point cloud semantic segmentation. *Science China Information Sciences*, 65(9):192104, 2022.

[28] Jiachen Sun, Qingzhao Zhang, Bhavya Kailkhura, Zhiding Yu, Chaowei Xiao, and Z Morley Mao. Benchmarking robustness of 3d point cloud recognition against common corruptions. *arXiv preprint arXiv:2201.12296*, 2022.

[29] Xiaoyu Tian, Haoxi Ran, Yue Wang, and Hang Zhao. Geomae: Masked geometric target prediction for self-supervised point cloud pre-training. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13570–13580, 2023.

[30] Mikaela Angelina Uy, Quang-Hieu Pham, Binh-Son Hua, Thanh Nguyen, and Sai-Kit Yeung. Revisiting point cloud classification: A new benchmark dataset and classification model on real-world data. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 1588–1597, 2019.

[31] Changshuo Wang, Meiqing Wu, Siew-Kei Lam, Xin Ning, Shangshu Yu, Ruiping Wang, Weijun Li, and Thambipillai Srikanthan. Gpsformer: A global perception and local structure fitting-based transformer for point cloud understanding. In *European Conference on Computer Vision*, pages 75–92. Springer, 2024.

[32] Peng-Shuai Wang. Octformer: Octree-based transformers for 3d point clouds. *ACM Transactions on Graphics (TOG)*, 42(4):1–11, 2023.

[33] Ziyang Wang, Jian-Qing Zheng, Yichi Zhang, Ge Cui, and Lei Li. Mamba-unet: Unet-like pure visual mamba for medical image segmentation. *arXiv preprint arXiv:2402.05079*, 2024.

[34] Xiaoyang Wu, Yixing Lao, Li Jiang, Xihui Liu, and Hengshuang Zhao. Point transformer v2: Grouped vector attention and partition-based pooling. *Advances in Neural Information Processing Systems*, 35:33330–33342, 2022.

[35] Xumin Yu, Lulu Tang, Yongming Rao, Tiejun Huang, Jie Zhou, and Jiwen Lu. Point-bert: Pre-training 3d point cloud transformers with masked point modeling. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 19313–19322, 2022.

[36] Renrui Zhang, Ziyu Guo, Peng Gao, Rongyao Fang, Bin Zhao, Dong Wang, Yu Qiao, and Hongsheng Li. Point-m2ae: multi-scale masked autoencoders for hierarchical point cloud pre-training. *Advances in neural information processing systems*, 35:27061–27074, 2022.

[37] Hengshuang Zhao, Li Jiang, Jiaya Jia, Philip HS Torr, and Vladlen Koltun. Point transformer. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 16259–16268, 2021.

[38] Qiang Zheng, Chao Zhang, and Jian Sun. Pointmt: Efficient point cloud analysis with hybrid mlp-transformer architecture. *arXiv preprint arXiv:2408.05508*, 2024.

[39] Qiang Zheng, Chao Zhang, and Jian Sun. Sa-mlp: Enhancing point cloud classification with efficient addition and shift operations in mlp architectures. *arXiv preprint arXiv:2409.01998*, 2024.

[40] Xiao Zheng, Xiaoshui Huang, Guofeng Mei, Yuenan Hou, Zhaoyang Lyu, Bo Dai, Wanli Ouyang, and Yongshun Gong. Point cloud pre-training with diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 22935–22945, 2024.

[41] Yanmei Zou, Hongshan Yu, Zhengeng Yang, Zechuan Li, and Naveed Akhtar. Improved mlp point cloud processing with high-dimensional positional encoding. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 7891–7899, 2024.

A Qualitative Analysis

A.1 Masking Process Visualization

As noted in Sec.3.3, the entire masking workflow first filters high-similarity point tokens via SMS using a threshold, then randomly masks about 60% of the remaining tokens. An asymmetric Mamba autoencoder then extracts point features, which are reconstructed via a simple prediction head. Fig. 5 shows qualitative results of mask modeling on the ShapeNet validation set, where the ZigzagPointMamba model effectively reconstructs masked regions. The zigzag scan path generates structured 3D paths to enhance feature sequence continuity for capturing local geometry; SMS masks redundant features by semantic similarity to preserve key semantics and strengthen global modeling. Their combination allows ZigzagPointMamba to reliably reconstruct masked areas at high masking rates, providing strong self-supervised knowledge for downstream tasks.

A.2 Part segmentation visualization

In this subsection, we present the qualitative results of the part segmentation performed by the ZigzagPointMamba model on the ShapeNetPart validation set, including the ground truth results and the predicted results. As shown in Fig. 6, thanks to the synergistic effect of the zigzag scan path and the SMS, the ZigzagPointMamba model demonstrates outstanding performance in the part segmentation task.

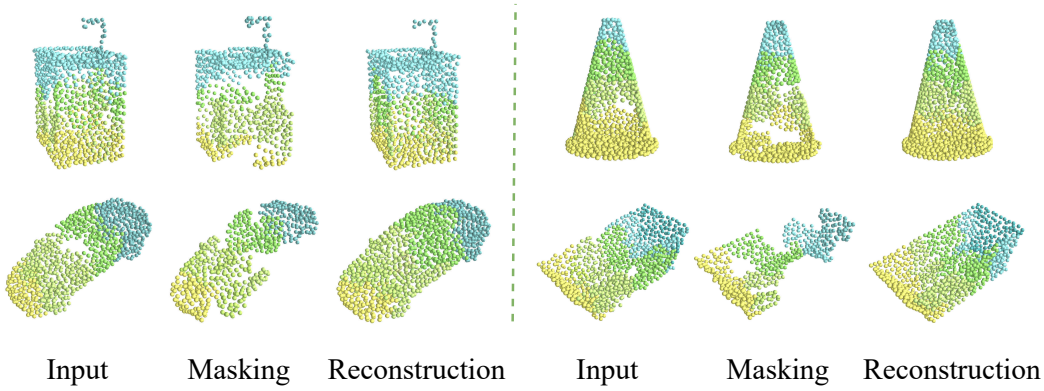


Figure 5: This figure shows the qualitative analysis results of the mask predictions made by the ZigzagPointMamba model on the ShapeNet validation set.

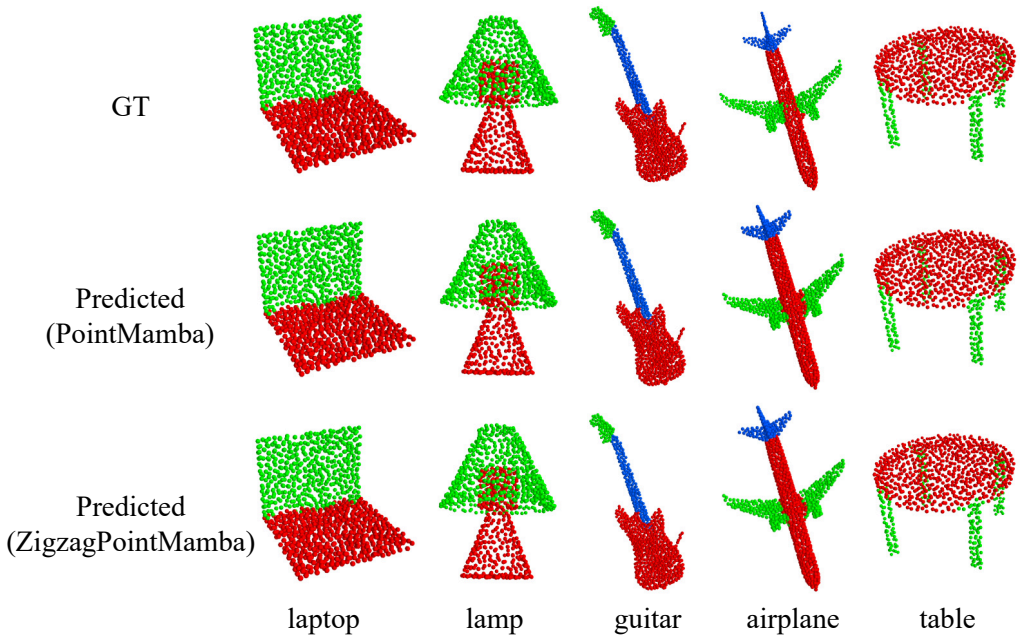


Figure 6: The qualitative outcomes of part segmentation achieved by our ZigzagPointMamba model on the ShapeNetPart dataset.