2024 4th International Conference on Intelligent Technology and Embedded Systems (ICITES) | 979-8-3315-0629-2/24/$31.00 ©2024 IEEE | DOI: 10.1109/ICITES62688.2024.10777444

PointABM: Integrating Bidirectional Mamba and Multi-Head Self-Attention for Point Cloud Analysis

Jia-Wei Chen, Yu-Jie Xiong*, Dong-Hai Zhu, Jia-Chen Zhang, Zheng Zhou
The School of Electronic and Electrical Engineering
Shanghai University of Engineering Science
Shanghai, China
xiong@sues.edu.cn

***Abstract*—In recent years, Transformer has achieved dominance across a multitude of disciplines, attributed to its superior global modeling capabilities. Currently, the Mamba model, distinguished by its linear complexity, is emerging as a formidable challenger. Despite existing advancements, considerable room for improvement remains in point cloud processing, especially within embedded application contexts such as robotic navigation and autonomous vehicles. Based on this observation, we propose PointABM, a hybrid model that integrates Mamba and Transformer architectures to enhance global feature extraction, thereby improving the performance of 3D point cloud analysis. More specifically, first we design a Transformer Block to enhance the representation of global features. Then, we propose a bidirectional Mamba, which comprises both a traditional token forward SSM and an innovative backward SSM. Experimental results demonstrate that integrating Mamba with Transformer significantly enhances the model's capability to analyze 3D point clouds, offering substantial improvements in both efficiency and accuracy for critical applications.**

***Index Terms*—3D Point Cloud, Bidirectional Mamba, Transformer, Robotic Navigation, Autonomous Vehicles.**

I. Introduction

Point cloud analysis is one of the most widely studied fields of computer vision [1]. It has wide applications in fields such as autonomous vehicles and robotic navigation, playing a crucial role in the development of artificial intelligence. 3D point clouds are primarily obtained through LiDAR scanning. Continuous-wave LiDAR operates by emitting laser waves from a transmitter. When these waves strike an object, they are reflected back and captured by a receiver within LiDAR scanner. The distance to the object is then calculated by measuring the phase shift of the reflected laser light. To enhance processing speed and reduce data transmission loads, this raw laser data can be processed directly on edge computing devices located within LiDAR system. These edge devices analyze the data in real-time, generating and processing 3D point cloud data locally.

As a 3D image, Point clouds have their own unique data characteristics. It composed of numerous unordered and unpatterned points in three-dimensional space. This necessitates that the entire developmental trajectory of point cloud research be devoted to addressing the challenge posed by the disordered nature of point clouds. To address this challenge, a variety of methods in deep learning have arisen. Vox-based method voxelize the 3D space to enable the application of 3D discrete convolutions [2]. However, this ignores the sparsity of the 3D point cloud.

Then first work of point-based PointNet [3] and PointNet++ [4] utilise single symmetric function,max pooling to solving this problem. Subsequently, series point-base models such as PointNeXt [5], PointMLP [6], PointCNN [7] etc., training form scratch comes out. Transformer-based model achieve remarkable progress by its attention mechanism. Attention can effectively capture the relationship between points in point cloud, but also posed quadratic complexity for Transformer [8]. This will cause the increase in model parameters and computational requirements. The permutation invariance of the Transformer endows it with higher compatibility compared to other models. This establishes a foundation for our upcoming proposal to integrate the Transformer and Mamba [9] models.

Recently, Mamba first incorporates the integration of time-varying parameters into state space models, bringing a new selection mechanism to effectively compress context. Additionally, it proposes an efficient hardware-aware algorithm to enhance performance. This makes it a strong challenger to Transformers. However, the application of Mamba to point clouds is limited by its unidirectional model, resulting in less than expected performance in the field of point cloud processing.

To address these issues, we present PointABM. Mamba and Transformer are innovatively combined within a novel method. The powerful self-attention mechanism of the Transformer is leveraged to initially encode the point cloud features, aiming to obtain a more comprehensive representation of local features. Furthermore, its inherent input permutation invariance provides a foundation for its integration with Mamba. In order to break through the limitations of mamba's unidirectional encoding of point cloud features, we introduced Bidirectional Mamba to process point cloud data from both forward and reverse directions. PointABM successfully maintains the lightweight characteristics of Mamba while effectively leveraging the powerful feature processing capabilities of the Transformer's self-attention mechanism. And we adopted a masked autoencoder pre-training strategy similar to Point-MAE, and our method demonstrated exceptional adaptability to this approach.

In summary, this work makes the following contributions:

- We propose PointABM, a hybrid model that includes

both Transformer and Mamba, retains the lightweight characteristics of Mamba, and leverages the self-attention mechanism of Transformers.
- Mamba and Transformer architectures are successfully combined and applied to point cloud analysis, achieving substantial performance improvements with a relatively small increase in the number of parameters.
- Experiments reveal that PointABM achieves superior performance compared to the Transformer based models.

II. Related work

In recent computer vision field, point cloud plays an important role in representing the 3D sence because of its rich expression information. The purpose of point cloud analysis is to identify the overall attributes of the point cloud, enabling a clearer understanding of its structure and composition for applications that rely on accurate 3D representations. Initially, point clouds were converted into multi-view data(Multi-View convolutional neural network [10]), voxels(VoxNet [2]), or meshes as indirect methods to learn the representation of 3D objects. However, these methods often lead to the loss of the objects' 3D geometric information or excessive memory consumption.Point-based methods like PointNet [3] and PointNet++ [4] effectively address the limitations of earlier techniques by processing raw point clouds directly, thereby preserving the original geometric integrity. PointNet, introduced by Qi et al., uses a shared multilayer perceptron (MLP) to learn features at the individual point level, aggregating them into a global descriptor via max pooling. To better capture local geometric structures, PointNet++ extends this framework with a hierarchical approach that involves sampling and grouping layers, allowing for detailed multi-scale analysis of point cloud data.

A. Point Cloud Transformers

After the debut of the Point Cloud Transformer [11] (PCT), Transformer [8] have continued to be among the most commonly used models in point cloud analysis [12], [13]. This model leverages the powerful self-attention mechanism of Transformers to better capture the complex spatial relationships in point clouds by dynamically focusing on different parts of the input data, enabling it to effectively understand and represent the intricate structures and patterns present within point cloud datasets. The success of PCT demonstrated how to handle the unordered nature of point cloud data through self-attention, while effectively extracting information about the relative positions and attributes between points.

Subsequently, PointBERT [14] and PointMAE [15] each proposesed innovative pre-training methods for point clouds, effectively integrating self-supervised learning within the Transformer architecture. Both models employ strategy of randomly masking portions of point cloud, significantly enhancing their ability to process and comprehend the intricate features of point cloud data. Furthermore, these two methods provide stable and reliable pre-training strategies for subsequent models, which in turn reduces the dependency on large labeled datasets. This advancement makes it possible to achieve high performance in point cloud analysis even with limited annotated data, paving the way for more efficient and scalable solutions in the field.

The exceptional performance of Transformers makes them highly suitable for integration into autoencoders, substantially enhancing the effectiveness of downstream point cloud analysis tasks. However, the attention mechanism's $O(n^2d)$ time complexity, with n as the input token sequence length and d as the Transformer dimension, leads to substantial computational challenges as the input size grows, limiting their efficiency.

B. State Space Models

State Space Models (SSM), inspired by continuous systems, have emerged as promising frameworks for modeling sequential data. The Structured State Space Sequence Model (S4) [16], a predecessor in this field, is notable for capturing long-range dependencies with linear complexity and strong performance across various domains. To mitigate computational burdens, methods like HTTYH [17], DSS [18], and S4D [19] employ diagonal matrices within S4. Building on S4, the newly proposed S6 model introduces significant advancements in efficiency and scalability. Mamba [9] further enhances this by introducing selective SSM mechanism, achieving linear-time inference and effective training through hardware-aware algorithm. This innovation has extended to various domains, inspiring works in graph modeling, medical segmentation, and video understanding.

Building on the foundation of the S4 model, the newly introduced Mamba (S6) model incorporates significant advancements that enhance both efficiency and scalability. Mamba introduces a selective State Space Model (SSM) mechanism, achieving near-linear complexity and enabling highly effective training through a hardware-aware algorithm. This approach optimizes performance while adapting to the capabilities of the underlying hardware infrastructure, making it a significant leap forward in the field.

PointMamba [20] is the first to introduce the Mamba model into the field of point cloud classficiation. However, its performance did not meet the expected standards. Consequently, this paper proposes an innovative approach by integrating Transformers with the Mamba model to harness the strengths of both technologies. This hybrid architecture aims to improve the robustness and accuracy of point cloud classficiation by combining the Mamba model's efficient processing capabilities with the powerful contextual understanding provided by Transformers.

III. Method

A. Overall

Our method is designed to leverage the strengths of both the Transformer and Mamba models in the field of 3D point cloud analysis. To achieve this, we devise a Transformer block to facilitate the integration of both models during the manipulation of 3D point clouds. In this section, we introduces

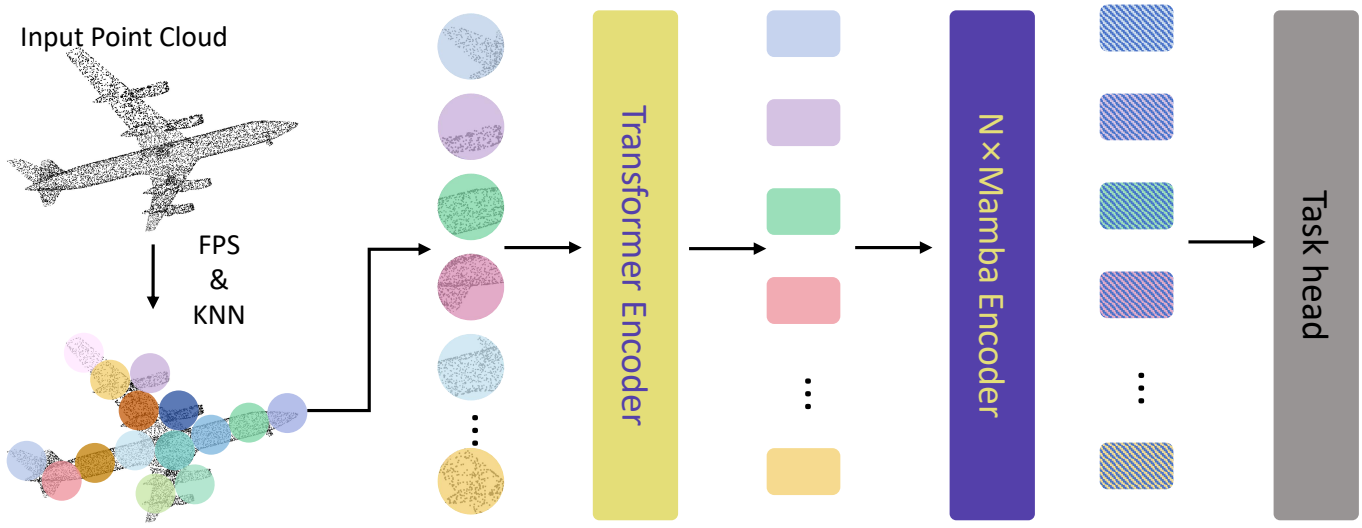

Fig. 1. The pipeline of PointABM. Initially, FPS and KNN are employed to extract keypoints and segment them into patches from the input point cloud. Then sent them into Transformer Encoder. Finally, the encoded features are loaded into a Mamba Encoder composed of N bidirectional Mambas.

multi-head self-attention, bidirectional Mamba, and the key design elements.

B. Transformer Block

PointABM utilizes a standard Transformer architecture that comprises multi-head self-throwing blocks and feed-forward network (FFN) blocks. The process is shown in Figure 1(a). After resorting, positional encoding is assigned to the features of each center point.

$$X' = X + \left(\text{Pos} \cdot P + \sum_{i=1}^{n} \alpha_i \phi_i(\text{Pos}, P) \right) \tag{1}$$

The encoded features are segmented and fed into individual self-attention heads. For each head, the input features are multiplied by three learnable weight matrices: W_Q, W_K, W_V.

$$Q = W_Q X'; \quad K = W_K X'; \quad V = W_V X'. \tag{2}$$

The Q, K, V matrices undergo self-attention processing.

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}} \right) V \tag{3}$$

Subsequently, the processed features are then combined, rejoined to the original features through a residual connection, and normalized. The introduction of self-attention also enhances our model's adaptability to pre-training methods based on masked autoencoder.

C. Bidirectional Mamba Block

The original design of the Mamba block was intended for one-dimensional sequence prediction, which leads to a lack of understanding of the global spatial information required for Point Clouds.To address this issue, we introduced Bidirectional Mamba(Bi-Mamba), which process prediction with forward and backward SSM.

Algorithm 1 Transformer Block Process

Input: token sequence $\mathbf{P}_{n-1}$: (S,G,C)
Number of heads H
Output: token sequence $\mathbf{P}_n$: (S,G,C)

$PE \leftarrow$ Positional Encoding(G, C)
$x \leftarrow x + PE$
$PE \leftarrow$ Initialize Positional Encoding with dimensions (C)
for each Head in P **do**
  $Q_h \leftarrow \text{Linear}_h^Q(\mathbf{P}_{n-1})$
  $K_h \leftarrow \text{Linear}_h^K(\mathbf{P}_{n-1})$
  $V_h \leftarrow \text{Linear}_h^V(\mathbf{P}_{n-1})$
  $A_h \leftarrow \text{Attention}(Q_h, K_h, V_h)$
  $\mathbf{P}_{n-1} \leftarrow A_h + \mathbf{P}_{n-1}$
  $\mathbf{P}_{n-1} \leftarrow \text{Norm}(\mathbf{P}_{n-1})$
  $\text{FFN}_{input} \leftarrow \text{Linear}(\text{ReLU}(\text{Linear}(\mathbf{P}_{n-1})))$
  $\mathbf{P}_{n-1} \leftarrow \mathbf{P}_{n-1} + \text{FFN}_{input}$
  $\mathbf{P}_{n-1} \leftarrow \text{Norm}(\mathbf{P}_{n-1})$
end forreturn $\mathbf{P}_n$: (S, G, C) =0

The backward SSM and forward SSM possessed by Bi-SSM are utilized to process point cloud features. For each direction, one-dimensional convolution is first applied to the input point x to obtain x'. Subsequently, an MLP layer projects x' onto $\mathbf{B}_o$, $\mathbf{C}_o$, and Δ_o.

$$x' = \text{SILU}(\text{Conv1d}(\text{Linear}^2(\text{Norm}(P_{n-1})))) \tag{4}$$

$$\Delta B_0 = \log(1 + \exp(\text{Linear}^A + \text{Parameter}^A)) \tag{5}$$

$$y_0 = \text{SSM}\left(A_0, \Delta B_0 \otimes \text{Linear}^0(x'), \text{Linear}^1(x')\right)(x') \tag{6}$$

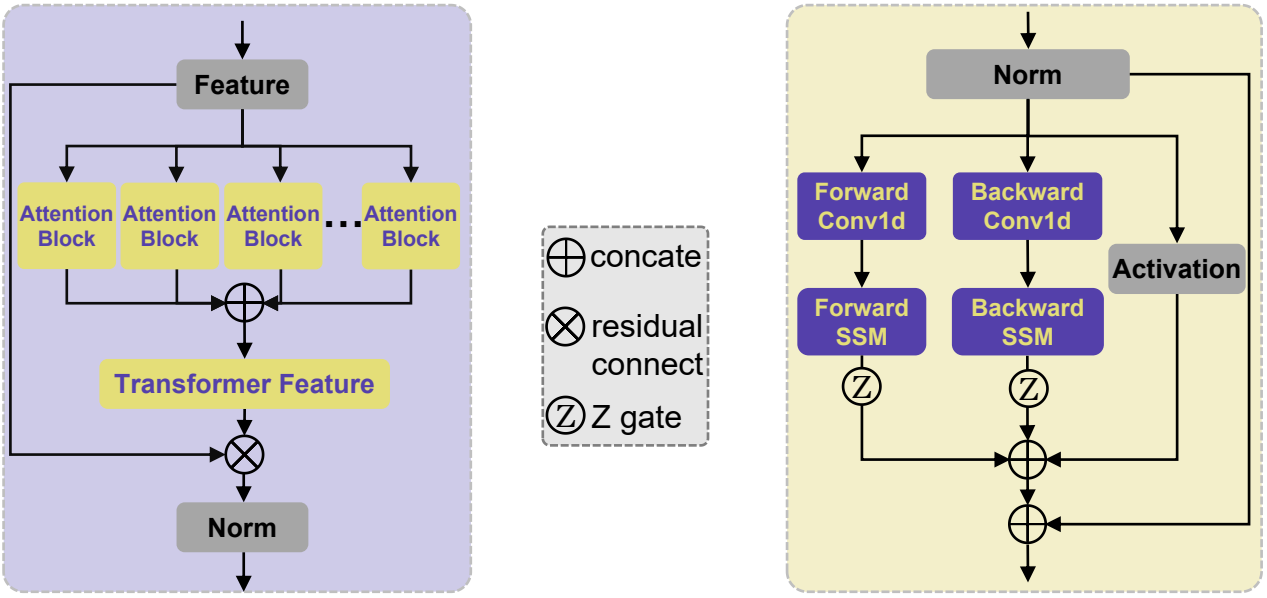

Fig. 2. Transformer Block and Bidirectional Mamba Block.

Algorithm 2 Bidirectional Mamba Block Process

Input: token sequence $\mathbf{P}_{n-1}$: (S,G,C)
Output: token sequence $\mathbf{P}_n$: (S,G,C)

$\mathbf{P}'_{n-1}$: (S,G,C) $\leftarrow$ Norm(P_{n-1})
x : (S,G,C) $\leftarrow$ Linear$^{x}(P_{n-1})$
z : (S,G,C) $\leftarrow$ Linear$^{z}(P_{n-1})$
for o in {forward, backward} **do**
  x'_o : (S,G,C) $\leftarrow$ SiLU(Conv1d$_o$(x))
  $\mathbf{B}_o$: (S,G,C) $\leftarrow$ Linear$_o^B(x'_o)$
  $\mathbf{C}_o$: (S,G,C) $\leftarrow$ Linear$_o^C(x'_o)$
  Δ_o : (S,G,C) $\leftarrow \log(1+\exp(\textbf{Linear}_o^{\Delta} + \textbf{Parameter}_o^{\Delta}))$
  $\overline{\mathbf{A}}_o$(S,G,C,N) $\leftarrow \Delta_o \otimes \textbf{Parameter}_o^A$: (C,N)
  $\overline{\mathbf{B}}_o$(S,G,C,N) $\leftarrow \Delta_o \otimes \mathbf{B}_o$
  y_o : (S,G,C) $\leftarrow \textbf{SSM}(\overline{\mathbf{A}}_o, \overline{\mathbf{B}}_o, \mathbf{C}_o)(x'_o)$
end for
$y'_{forward}$: (S,G,C) $\leftarrow y_{forward} \bigodot \textbf{SiLU}(z)$
$y'_{backward}$: (S,G,C) $\leftarrow y_{backward} \bigodot \textbf{SiLU}(z)$
P_n : (S,G,C) $\leftarrow$ Linear$^{P}(y'_{forward} + y'_{backward}) + P_{m-1}$
Return: P$_n$ =0

Then, Δ_o is used to update $\mathbf{A}_o$ and $\mathbf{B}_o$. After obtaining y_{forward} and y_{backward} via SSM, the values are passed through a Z-gate and then summed to obtain the final output $\mathbf{P}_o$.

$$z = \text{SILU}(\text{Linear}^{3}(P_{n-1})) \tag{7}$$

$$P_n = \text{Linear}'((y_{\text{forward}} + y_{\text{backward}}) \otimes z) + P_{n-1} \tag{8}$$

IV. Experiments

In this section, we will introduce the specific implementation details of the experiment. Then we evaluated the performance of PointABM on ModelNet and three variants of ScanObjectNN. Finally, we show the results of the ablation study for our model.

A. Implementation Details

To address the issue of varying point cloud resolutions, we divide points into different batches.

In Modelnet40 [31], the process begins by using farthest point sampling to select a random set of 1024 points, which are then divided into N = 64 point patches, each containing G = 32 points. For ScanObjectNN [32] and ShapeNetPart [33], with a point count of M = 2048, the division is into N = 128 patches, each holding G = 32 points.

The PointABM encoder features a combination of one Transformer layer and 12 Bi-SSM layers, each with a feature dimension C = 384. Each Transformer block consists of 8 heads. We utilize the AdamW optimizer and employ a cosine learning rate decay strategy. During the pretraining phase, the ShapeNetCore dataset [33], comprising 51,300 3D models, serves as the pretranning dataset. The rest of the settings are essentially the same as those used for training from scratch. All experiments are conducted using one NVIDIA RTX 4090 GPU.

B. Experiments Results

Classification in ScanObjectNN [32]:
ScanObjectNN dataset comprises 15,000 objects segmented into 15 categories, captured from real-world indoor environments characterized by their cluttered backgrounds. This dataset presents three distinct variants for testing and analysis:

TABLE I
OBJECT CLASSIFICATION ON SCANOBJECTNN.

Methods	Backbone	Param.(M)	FLOPs(G)	OBJ-BG(%)	OBJ-ONLY(%)	PB-T50-RS(%)
Supervised Learning Only						
PointNet [3]	-	3.5	0.5	73.3	79.2	68.0
PointNet++ [4]	-	1.5	1.7	82.3	84.3	77.9
PointCNN [7]	-	0.6	0.9	86.1	85.5	78.5
DGCNN [21]	-	1.8	2.4	82.8	86.2	78.1
PRA-Net [22]	-		-	-	-	81.0
MVTN [23]	-	11.2	43.7	-	-	82.8
PointNeXt [5]	-	1.4	1.6	-	-	87.7
PointMLP [6]	-	12.6	31.4	-	-	85.4
DeLA [24]	-	5.3	1.5	-	-	90.4
Training from scratch						
PointMamba [20]	Mamba	12.3	3.1	88.29	87.78	82.48
PointABM(ours)	Mamba & Transformer	15.1	9.6	91.57	90.36	86.19
Training from pre-training						
Point-BERT [14]	Transformer	22.1	4.8	87.43	88.12	83.07
MaskPoint [25]	Transformer	22.1	4.8	89.30	88.10	84.30
Point-MAE [15]	Transformer	22.1	4.8	90.02	88.29	85.18
Point-M2AE [26]	Transformer	15.3	3.6	91.22	88.29	85.18
PointMamba-pre [20]	Mamba	12.3	3.1	90.71	88.47	84.87
PCM [27]	Mamba	34.2	45.0	-	-	88.10
PointABM-pre(ours)	Mamba & Transformer	15.1	9.6	**93.29**	**92.43**	**88.29**

TABLE II
OBJECT CLASSIFICATION ON MODELNET40.

Methods	Param.(M)	FLOPs(G)	OA(%)
Supervised Learning Only			
PointNet [3]	3.5	0.5	89.2
PointNet++ [4]	1.5	1.7	90.7
PointCNN [7]	0.6	0.9	92.2
DGCNN [21]	1.8	2.4	92.9
PRA-Net [22]	-	-	93.1
MVTN [23]	11.2	43.7	93.8
PointNeXt [5]	1.4	1.6	94.0
PointMLP [6]	13.2	31.4	**94.0**
DeLA [24]	5.3	1.5	**94.0**
PointMamba [20]	12.3	3.1	92.4
PointABM(ours)	15.1	9.6	92.6
Training from pre-training			
Point-BERT [14]	22.1	4.8	93.4
MaskPoint [25]	22.1	4.8	93.8
Point-MAE [15]	22.1	4.8	**94.4**
Point-M2AE [26]	15.3	3.6	94.0
PointMamba-pre [20]	12.3	3.1	93.6
PCM [27]	34.2	45.0	93.4
PointABM-pre(ours)	15.1	9.6	93.1

TABLE III
OBJECT SEGMENTATION IN SHAPENETPART.

Methods	Param.(M)	mIoU_C(%)	mIoU_I(%)
Training from scratch			
PointNet [3]	3.6	80.4	83.7
PointNet++ [4]	1.0	81.9	85.1
DGCNN [21]	1.3	82.3	85.2
Transformer [8]	27.1	83.4	85.1
PointABM(ours)	20.0	84.1	85.7
Training from pre-training			
OcCo [28]	27.1	83.4	84.7
PointContrast [29]	37.9	-	85.1
CrossPoint [30]	-	-	85.5
PointBERT [14]	27.1	84.1	85.6
Point-MAE [15]	27.1	84.2	86.1
PointMamba [20]	17.4	84.4	86.0
PointABM(ours)	20.0	84.1	86.0

OBJ_BG, OBJ_ONLY, and PB_T50_RS, each designed to evaluate different aspects of object recognition under varying complexly conditions. The configuration for our experiments taking a subset of 2,048 points per object and using rotation as data augmentation. **PointABM** surpasses most effective Transformer-based method PointMAE, 3.58%, 4.14%, 3.42% on OBJ_BG, OBJ_ONLY, and PB_T50_RS. Besides, **PointABM** also exceeding Mamba-based mothod PointMamba 2.58%, 3.96%, 3.33%.

Classification in Modelnet40 [31]:
Modelnet40 is a widely recognized synthetic dataset for 3D object classification, comprising 12,311 clean CAD models across 40 categories. The dataset is conventionally split into 9,843 instances for training and 2,468 for testing, adhering to established protocols. Each category is represented by 100 unique models, establishing ModelNet40 as a fundamental benchmark in the field. During training, random scaling and translation are employed to enhance generalization. Despite its status as a clean dataset, PointABM's inability to fully demonstrate interference resistance still resulted in an impres-

TABLE IV
THE EFFECT OF TRANSFORMER EMBEDDING.

fusion	feature dimension	Param.(M)	OBJ_BG(%)	OBJ_ONLY(%)	PB_T50_RS(%)
None	384	12.3	88.30	87.78	82.48
Concatenation	768	47.7	**90.72**	88.29	**84.62**
Residual Connection	384	14.8	90.43	**89.15**	84.55

TABLE V
THE EFFECT OF BIDIRECTIONAL MAMBA.

method	feature dimension	Param.(M)	OBJ_BG (%)	OBJ_ONLY (%)	PB_T50_RS (%)
None	384	12.3	88.30	87.78	82.48
Bi-SSM	384	12.5	91.22	89.84	85.94
Concatenation	768	48.1	**91.57**	88.81	84.17
Residual Connection	384	15.1	**91.57**	**90.36**	**86.19**

sive accuracy rate of 93.1 %.

Segmentation in ShapeNetPart [33]:
ShapeNetPart dataset is a widely recognized synthetic dataset for 3D object part segmentation, comprising 16,881 meticulously annotated CAD models across 16 categories. Each model is detailed with annotations for various components of the object, such as the legs, seat, and back of a chair, making it an ideal choice for research and development of algorithms aimed at fine-grained recognition and segmentation of complex 3D objects.

C. Ablation Study

To improve the effectiveness of each component, a study was conducted on the utility of each component within the architecture using the ScanObjectNN [32] dataset. And to ensure the purity of the ablation study results, all our ablation experiments were conducted using training from scratch.

Transformer Block
As the first to integrate Transformer and Mamba in the point cloud field, we attempted two feature fusion methods: concatenation and residual connection. TABLE.II shows each feature fusion method brought a noticeable improvement. This indicates that Transformer embedding can effectively offer more refined feature information to the Mamba model. Concatenation feature dimension even take better accuracy. But with the doubling of feature dimensions, the size of the model increases dramatically. Moreover, in the subsequent ablation studies of BI-SSM, the feature fusion method using residual connections demonstrated superior compatibility.

Bidirectional Mamba Block
In this section, the focus is on examining the effectiveness of the Bidirectional Mamba and exploring its outcomes when combined with two different feature fusion methods within the Transformer block. We achieved improvements of +2.92%, +2.06%, and +2.06% on three variants by directly using Bidirectional Mamba. Additionally, after applying Bidirectional Mamba following a residual-connected Transformer Block, we observed further enhancements of +3.27%, +2.58%, and +3.71%. This demonstrates the effectiveness of Bidirectional Mamba for the classification of unordered point clouds.

V. CONCLUSION

This paper presents PointABM, a method for point cloud analysis that integrates bidirectional Mamba and Transformer. Specifically, PointABM maintains the near-linear complexity of Mamba while enhancing its capability to understand point cloud information through the self-attention mechanism of the Transformer. Through experimentation and validation, PointABM demonstrates satisfactory performance, achieving significant improvements with a modest increase in the number of parameters.

REFERENCES

[1] S. Shi, Z. Wang, J. Shi, X. Wang, and H. Li, "From points to parts: 3d object detection from point cloud with part-aware and part-aggregation network," *IEEE Transactions on Pattern Analysis & Machine Intelligence*, vol. 43, no. 08, pp. 2647–2664, 2021.

[2] D. Maturana and S. Scherer, "Voxnet: A 3d convolutional neural network for real-time object recognition," in *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2015, pp. 922–928.

[3] R. Q. Charles, H. Su, M. Kaichun, and L. J. Guibas, "Pointnet: Deep learning on point sets for 3d classification and segmentation," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 77–85.

[4] C. R. Qi, L. Yi, H. Su, and L. J. Guibas, "Pointnet++: deep hierarchical feature learning on point sets in a metric space," in *Proceedings of the 31st International Conference on Neural Information Processing Systems*, ser. NIPS'17. Red Hook, NY, USA: Curran Associates Inc., 2017, p. 5105–5114.

[5] G. Qian, Y. Li, H. Peng, J. Mai, H. A. Al Kader Hammoud, M. Elhoseiny, and B. Ghanem, "Pointnext: revisiting pointnet++ with improved training and scaling strategies," in *Proceedings of the 36th International Conference on Neural Information Processing Systems*, ser. NIPS '22. Red Hook, NY, USA: Curran Associates Inc., 2024.

[6] X. Ma, C. Qin, H. You, H. Ran, and Y. Fu, "Rethinking network design and local geometry in point cloud: A simple residual MLP framework," in *International Conference on Learning Representations*, 2022.

[7] Y. Li, R. Bu, M. Sun, W. Wu, X. Di, and B. Chen, "Pointcnn: Convolution on x-transformed points," in *Advances in Neural Information Processing Systems*, S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, Eds., vol. 31. Curran Associates, Inc., 2018.

[8] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," in *Proceedings of the 31st International Conference on Neural Information Processing Systems*, ser. NIPS'17. Red Hook, NY, USA: Curran Associates Inc., 2017, p. 6000–6010.

[9] A. Gu and T. Dao, "Mamba: Linear-time sequence modeling with selective state spaces," 2024, arXiv:2312.00752 [cs.LG].

[10] H. Su, S. Maji, E. Kalogerakis, and E. Learned-Miller, "Multi-view convolutional neural networks for 3d shape recognition," in *2015 IEEE International Conference on Computer Vision (ICCV)*, 2015, pp. 945–953.

[11] M.-H. Guo, J.-X. Cai, Z.-N. Liu, T.-J. Mu, R. R. Martin, and S.-M. Hu, "Pct: Point cloud transformer," *Computational Visual Media*, vol. 7, no. 2, p. 187–199, 2021.

[12] N. Engel, V. Belagiannis, and K. Dietmayer, "Point transformer," *IEEE Access*, vol. 9, pp. 134 826–134 840, 2021.

[13] X. Wu, Y. Lao, L. Jiang, X. Liu, and H. Zhao, "Point transformer v2: Grouped vector attention and partition-based pooling," in *Advances in Neural Information Processing Systems*, S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh, Eds., vol. 35. Curran Associates, Inc., 2022, pp. 33 330–33 342.

[14] X. Yu, L. Tang, Y. Rao, T. Huang, J. Zhou, and J. Lu, "Point-bert: Pre-training 3d point cloud transformers with masked point modeling," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition*. Los Alamitos, CA, USA: IEEE, 2022, pp. 19 291–19 300.

[15] Y. Pang, W. Wang, F. E. Tay, W. Liu, Y. Tian, and L. Yuan, "Masked autoencoders for point cloud self-supervised learning," in *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part II*. Springer, 2022, pp. 604–621.

[16] A. Gu, K. Goel, and C. Re, "Efficiently modeling long sequences with structured state spaces," in *International Conference on Learning Representations*, 2022.

[17] A. Gu, I. Johnson, A. Timalsina, A. Rudra, and C. Re, "How to train your HIPPO: State space models with generalized orthogonal basis projections," in *International Conference on Learning Representations*, 2023.

[18] A. Gupta, A. Gu, and J. Berant, "Diagonal state spaces are as effective as structured state spaces," in *Advances in Neural Information Processing Systems*, S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh, Eds., vol. 35. Curran Associates, Inc., 2022, pp. 22 982–22 994.

[19] A. Gu, K. Goel, A. Gupta, and C. Ré, "On the parameterization and initialization of diagonal state space models," in *Advances in Neural Information Processing Systems*, S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh, Eds., vol. 35. Curran Associates, Inc., 2022, pp. 35 971–35 983.

[20] D. Liang, X. Zhou, W. Xu, X. Zhu, Z. Zou, X. Ye, X. Tan, and X. Bai, "Pointmamba: A simple state space model for point cloud analysis," 2024.

[21] Y. Wang and J. M. Solomon, "Object dgcnn: 3d object detection using dynamic graphs," in *Advances in Neural Information Processing Systems*, M. Ranzato, A. Beygelzimer, Y. Dauphin, P. Liang, and J. W. Vaughan, Eds., vol. 34. Curran Associates, Inc., 2021, pp. 20 745–20 758.

[22] Z. Liu, X. Yuan, Y. Li, Z. Shangguan, L. Zhou, and B. Hu, "Pra-net: Part-and-relation attention network for depression recognition from facial expression," *Computers in Biology and Medicine*, vol. 157, p. 106589, 2023.

[23] A. Hamdi, S. Giancola, and B. Ghanem, "Mvtn: Multi-view transformation network for 3d shape recognition," in *ICCV*, 2021, pp. 1–11.

[24] B. Chen, Y. Xia, Y. Zang, C. Wang, and J. Li, "Decoupled local aggregation for point cloud learning," 2023.

[25] H. Liu, M. Cai, and Y. J. Lee, "Masked discrimination for self-supervised learning on point clouds," in *Computer Vision – ECCV 2022*, S. Avidan, G. Brostow, M. Cissé, G. M. Farinella, and T. Hassner, Eds. Cham: Springer Nature Switzerland, 2022, pp. 657–675.

[26] R. Zhang, Z. Guo, P. Gao, R. Fang, B. Zhao, D. Wang, Y. Qiao, and H. Li, "Point-m2ae: Multi-scale masked autoencoders for hierarchical point cloud pre-training," in *Advances in Neural Information Processing Systems*, S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh, Eds., vol. 35. Curran Associates, Inc., 2022, pp. 27 061–27 074.

[27] T. Zhang, X. Li, H. Yuan, S. Ji, and S. Yan, "Point cloud mamba: Point cloud learning via state space model," 2024.

[28] H. Wang, Q. Liu, X. Yue, J. Lasenby, and M. J. Kusner, "Unsupervised point cloud pre-training via occlusion completion," in *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021, pp. 9762–9772.

[29] S. Xie, J. Gu, D. Guo, C. R. Qi, L. Guibas, and O. Litany, "Pointcontrast: Unsupervised pre-training for 3d point cloud understanding," in *Computer Vision – ECCV 2020*, A. Vedaldi, H. Bischof, T. Brox, and J.-M. Frahm, Eds. Cham: Springer International Publishing, 2020, pp. 574–591.

[30] M. Afham, I. Dissanayake, D. Dissanayake, A. Dharmasiri, K. Thilakarathna, and R. Rodrigo, "Crosspoint: Self-supervised cross-modal contrastive learning for 3d point cloud understanding," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2022, pp. 9902–9912.

[31] Z. Wu, S. Song, A. Khosla, F. Yu, L. Zhang, X. Tang, and J. Xiao, "3d shapenets: A deep representation for volumetric shapes," in *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. Los Alamitos, CA, USA: IEEE Computer Society, 2015, pp. 1912–1920.

[32] M. Uy, Q. Pham, B. Hua, T. Nguyen, and S. Yeung, "Revisiting point cloud classification: A new benchmark dataset and classification model on real-world data," in *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*. Los Alamitos, CA, USA: IEEE Computer Society, 2019, pp. 1588–1597.

[33] A. X. Chang, T. Funkhouser, L. Guibas, P. Hanrahan, Q. Huang, Z. Li, S. Savarese, M. Savva, S. Song, H. Su, J. Xiao, L. Yi, and F. Yu, "Shapenet: An information-rich 3d model repository," 2015.