LABMamba: A Locally-Guided Alternating Bidirectional Mamba Method for Point Cloud Classification

Yang Yu [1], Zili Yan [1], Xingpeng Zhang †[1]
[1]School of Computer Science and Software Engineering, Southwest Petroleum University, Chengdu, 610500 China

***Abstract*—The unordered nature and complex local structures of point clouds challenge accurate feature modeling. While state-space models (SSMs) like Mamba offer linear-time global modeling, they struggle with fine-grained local geometry and irregular 3D topologies. We propose LABMamba, a geometric-aware SSM framework for point cloud classification. Our key innovation is a geometric-aware group embedding (GAGE) module that integrates explicit geometric priors (normals, curvature, density) with learned features, addressing Mamba's local geometric limitation. We also introduce an alternating bidirectional Mamba strategy that alternately models sequence and channel dimensions, differing from parallel bidirectional designs. This improves performance, reduces sensitivity to point ordering, and minimizes reliance on heuristic serialization. Extensive experiments show state-of-the-art or competitive results, with ablations validating each component.**

***Index Terms*—Point cloud classification, Point cloud segmentation, Local geometric features, Bidirectional Mamba**

I. Introduction

In recent years, point cloud analysis has achieved significant progress with the development of deep neural networks, playing a crucial role in 3D vision tasks such as object classification and scene understanding [1]. However, point clouds are inherently unordered and irregular, making it challenging to simultaneously capture fine-grained local geometric structures and long-range dependencies.

Existing approaches attempt to address these challenges using Multi-Layer Perceptrons (MLPs) [1]–[4], Transformers [5], and more recently, State Space Models (SSMs) such as Mamba [6], [7]. While MLP-based methods effectively model local neighborhoods, they lack global context awareness. Transformer-based methods introduce global attention but suffer from quadratic complexity and scalability issues. Mamba-based methods provide an efficient alternative with linear complexity and strong global modeling capability. However, they typically (1) lack explicit modeling of fine-grained local geometry and (2) remain sensitive to point serialization order, which limits their robustness in irregular 3D scenarios.

To address these limitations, we propose LABMamba, a geometric-aware SSM framework that integrates explicit geometric descriptors with learned features via the GAGE module, effectively enhancing local geometric perception. Our framework introduces a novel alternating bidirectional Mamba strategy for complementary sequence and channel dependency modeling, differing from parallel variants, and exhibits reduced sensitivity to point cloud serialization order. Extensive experiments on multiple benchmarks demonstrate state-of-the-art or competitive performance, supported by thorough ablation studies.

This work is supported by the National Natural Science Foundation of China (No. U23A2046 and 62441610). † Corresponding author: xpzhang@swpu.edu.cn

II. Related Work

A. MLP-based Methods

MLP-based methods are fundamental in point cloud processing for their simplicity and efficiency. PointNet [1] pioneered end-to-end learning using shared MLPs and symmetric functions for permutation invariance, but lacked explicit local geometric capture. To address this, PointNet++ [2] incorporated farthest point sampling (FPS) and neighborhood aggregation for multi-scale local modeling. Building on this, PointMLP [12] introduced local affine modules and residual connections to enhance local perception. Through structural enhancement and path optimization, the evolving MLP paradigm balances efficiency and accuracy in 3D vision tasks.

B. Transformer-based Methods

Transformer-based methods leverage the self-attention mechanism for global modeling and have become pivotal for point cloud processing. PCT [5] enhances local geometry by building local contexts and using offset attention. Recently, Point Transformer V3 (PTV3) [13] simplifies the architecture and improves efficiency via enhanced local-global aggregation. Methods like Point-MAE [14] extend Transformers to self-supervised pre-training through structural reconstruction. Despite these advances, such methods still face quadratic complexity with large point clouds. Ongoing evolution focuses on structural refinement, lightweight design, and self-supervised integration, enhancing modeling capability and flexibility for 3D vision tasks.

C. Mamba-based Methods

Mamba-based methods have gained momentum in 3D tasks due to their linear complexity and effective state-space

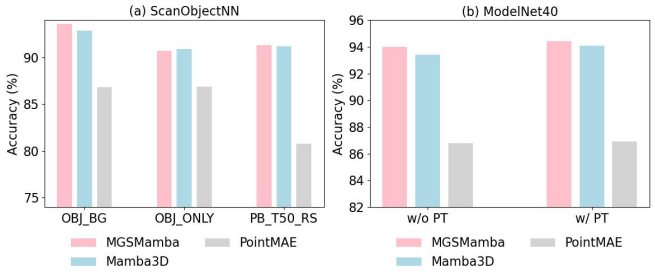

Fig. 1. LABMamba achieves superior overall accuracy against PointMAE and Mamba3D.

modeling. PointMamba [6] first adapted Mamba for point clouds. Subsequent works like PCM [10], NIMBA [9], and OctreeMamba [8] optimized serialization to suit Mamba's mechanics. Mamba3D [11] further integrated local structures with hierarchical perception. These adaptations highlight Mamba's advantages in balancing global context, efficiency, and flexibility for lightweight 3D modeling.

Unlike Mamba3D's parallel bidirectional modeling, our LABMamba introduces an alternating bidirectional strategy that sequentially processes sequence and channel dimensions for more synergistic feature integration. It further incorporates geometric priors via a GAGE module and jointly enhances local-global perception through: (1) a geometric-aware group embedding module, (2) a local enhancement mechanism, and (3) an alternating bidirectional Mamba for efficient dependency modeling. As shown in Fig. 1, LABMamba consistently outperforms representative Transformer-based and Mamba-based approaches.

III. Methodology

A. Overview

The overall architecture of LABMamba is shown in Fig. 2. The input point cloud is first grouped using FPS and KNN. Center points are embedded via an MLP, while neighborhood features are encoded by GAGE. The resulting features are processed by LGEM for local propagation and aggregation. Each LGEM is followed by a Mamba block: odd layers use bidirectional sequence Mamba for sequence modeling, and even layers use bidirectional channel Mamba for channel modeling. For classification, the output of the 12th Mamba block is fed to the classification head. For segmentation, multi-scale features from the 4th, 8th, and 12th layers are fused and passed to the segmentation head to enhance semantic parsing.

In summary, the overall point embedding and encoding layer structure can be represented as follows:

$$\begin{aligned}
\mathbf{z}_0 &= \mathrm{GAGE}(\mathbf{x}_p), \\
\mathbf{z}'_\ell &= \mathrm{LGEM}(\mathbf{z}_{\ell-1} + \mathbf{E}_{\mathrm{pos}}), \quad \ell = 1, \ldots, T, \\
\mathbf{z}_\ell &= \begin{cases} \text{C-bi-SSM}(\mathbf{z}'_\ell), & \ell = 1, 3, \ldots, T-1 \\ \text{S-bi-SSM}(\mathbf{z}'_\ell), & \ell = 2, 4, \ldots, T \end{cases}
\end{aligned}$$

where $\mathbf{z}_\ell$ denotes the output at the ℓ-th layer. The input patches $\mathbf{x}_p \in \mathbb{R}^{L\times K\times 3}$ are projected into the encoding space $\mathbf{z}_0 \in \mathbb{R}^{L\times C}$ via GAGE. $\mathbf{E}_{\mathrm{pos}}$ denotes the positional embedding obtained by applying an MLP to the center points. In the network, sequence bidirectional Mamba and channel bidirectional Mamba blocks are stacked alternately.

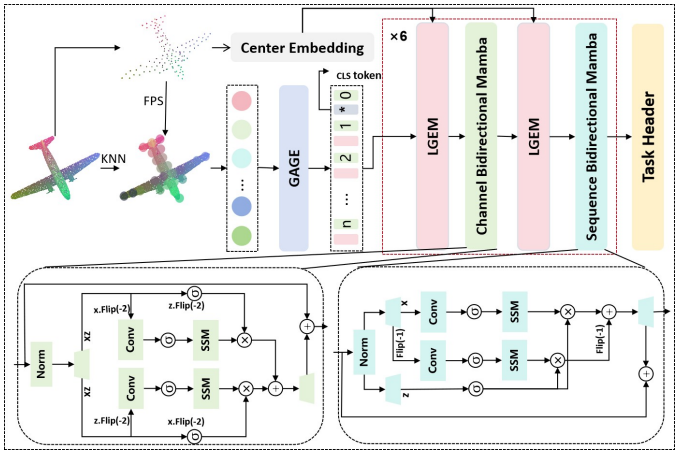

Fig. 2. LABMamba Network Architecture. It comprises point cloud grouping, feature embedding, local propagation, alternating bidirectional Mamba blocks, and task-specific heads for classification and segmentation.

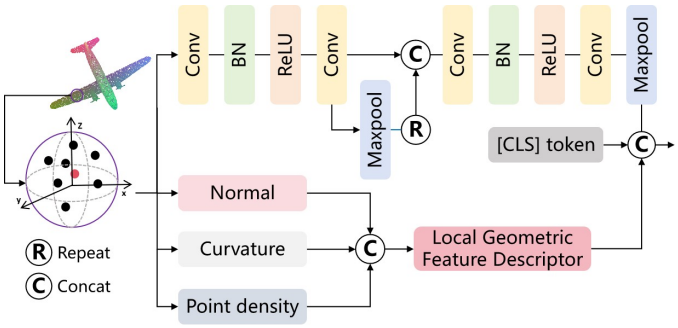

Fig. 3. The architecture of the GAGE.

B. Geometric-Aware Group Embedding

To enhance the representation capability of local geometric structures within point cloud groups, we propose a geometric-aware group embedding (GAGE) module that explicitly incorporates geometric priors into the feature extraction process. As illustrated in Fig. 3, the module first applies two consecutive stages of convolution, normalization, and pooling operations to each group, thereby generating preliminary local features. This stage maps the raw point-level information into a compact and discriminative latent space, providing a solid foundation for subsequent geometric encoding.

To further capture the intrinsic geometric properties of point clusters, a set of geometric descriptors is computed in the spherical coordinate system. For any point $\mathbf{p}_i = (x_i, y_i, z_i)$ in the group, its spherical coordinate representation is:

$$r_i = \sqrt{x_i^2 + y_i^2 + z_i^2}, \tag{1}$$

$$\theta_i = \arccos\left(\frac{z_i}{r_i}\right) \quad \phi_i = \arctan\left(\frac{y_i}{x_i}\right). \tag{2}$$

These spherical coordinates (r_i, θ_i, ϕ_i) together with the Cartesian coordinates $\mathbf{p}_i$, serve as the basis for computing the local covariance matrix $\mathbf{C}$ in Equation (7). The spherical representation captures directional geometric relationships, while the covariance matrix is computed in Cartesian space to ensure stable local surface estimation. The covariance matrix then enables the estimation of geometric descriptors including normal vectors and curvature.

Based on this representation, the normal vector $\mathbf{n}_i$ is estimated using local covariance analysis:

$$\mathbf{C} = \frac{1}{k}\sum_{i=1}^{k}(\mathbf{p}_i - \bar{\mathbf{p}})(\mathbf{p}_i - \bar{\mathbf{p}})^\top, \mathbf{n} = \arg\min_{\mathbf{v}} \mathbf{v}^\top \mathbf{C}\mathbf{v}, \quad (3)$$

where $\bar{\mathbf{p}}$ denotes the centroid of the group, and $\mathbf{n}$ corresponds to the eigenvector associated with the smallest eigenvalue of the covariance matrix $\mathbf{C}$. Furthermore, the curvature κ is defined as $\kappa = \frac{\lambda_{\min}}{\lambda_1+\lambda_2+\lambda_3}$, where $\lambda_{\min}$ denotes the smallest eigenvalue, and $\lambda_1, \lambda_2, \lambda_3$ are the ordered eigenvalues of $\mathbf{C}$. In addition, the point density is $\rho = \frac{k}{V}$, where k is the number of neighboring points within the group, and V is the volume of the local spherical neighborhood.

The normal vectors, curvature, and density jointly constitute the local geometric descriptor, which explicitly characterizes the structural properties of the group. To further integrate geometric priors with learned features, a learnable positional token $\mathbf{t}_{pos}$ is introduced for each group to enhance spatial awareness in the embedding space. The final group representation is formulated as:

$$\mathbf{F}_C = \text{Concat}(\mathbf{f}_{init}, \mathbf{g}, \mathbf{t}_{pos}), \quad (4)$$

where $\mathbf{f}_{init} \in \mathbb{R}^{C_1}$ denotes the preliminary features extracted by the convolutional layers, $\mathbf{g} \in \mathbb{R}^{C_2}$ represents the concatenated geometric descriptor (normal vectors, curvature, and density), and $\mathbf{t}_{pos} \in \mathbb{R}^{C_3}$ is the learnable positional token. The Concat operation concatenates these three components along the channel dimension to form $\mathbf{F}_C \in \mathbb{R}^{C_1+C_2+C_3}$.

By jointly leveraging learned features, explicit geometric attributes, and positional tokens, the GAGE module provides a more discriminative and geometry-aware representation of point cloud groups, thereby enabling deeper geometric modeling in subsequent network layers.

It is important to note the distinct roles of the GAGE and LGEM modules. GAGE operates at the *initial feature extraction stage*, focusing on injecting explicit geometric priors into the raw point representation. In contrast, the subsequent LGEM module operates at the *feature propagation stage*, enhancing the interaction of already-learned features within local neighborhoods. This two-stage design ensures both geometric prior injection and feature-level refinement, addressing different aspects of local geometric modeling.

C. Local Geometry-Aware Enhancement Module

Local geometric features play a pivotal role in point cloud modeling. To more effectively capture and exploit neighborhood information, we propose the local geometry-aware enhancement Module (LGEM), which facilitates the propagation and aggregation of local features, thereby strengthening the modeling of local geometric structures in point clouds. The architecture of LGEM is illustrated in Fig. 4.

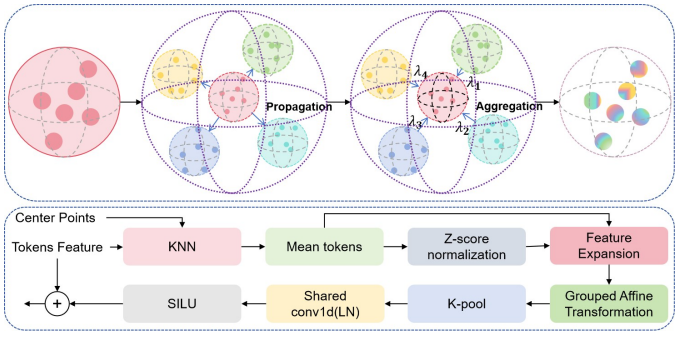

Fig. 4. The architecture of the LGEM. Note that λ represents a learnable scaling parameter for the affine transformation, and the Z-score normalization refers to standardizing the feature differences by their mean and variance.

Specifically, neighborhood features for each center point are first obtained using KNN, and their mean is computed as the normalization baseline. This process is expressed by Eq. (5) and (6):

$$\mathbf{F}_{mean} = \text{mean}\left(\text{KNN}(\mathbf{F}_C, C_{xyz})\right), \quad (5)$$

$$\tilde{\mathbf{F}}_K = \frac{\mathbf{F}_K - \mathbf{F}_{mean}}{\sqrt{\text{Var}(\mathbf{F}_K - \mathbf{F}_{mean}) + \epsilon}} \quad (6)$$

where ϵ is a small constant added for numerical stability, $\mathbf{F}_C$ denotes the features of the center points, C_{xyz} represents the coordinates of the center points, $\mathbf{F}_K$ indicates the neighborhood features, $\mathbf{F}_{mean}$ is the mean feature of the neighborhood, and $\tilde{\mathbf{F}}_K$ is the normalized neighborhood feature.

Subsequently, the normalized feature $\tilde{\mathbf{F}}_K$ is concatenated with the neighborhood mean feature $\mathbf{F}_{mean}$ repeated to match the neighborhood size. Locally adaptive affine transformation parameters α_k and β_k are then applied to compute the propagated feature representation $\mathbf{F}_K$, as formulated below:

$$\mathbf{F}_K = \left[\tilde{\mathbf{F}}_K \oplus \mathbf{F}_{mean}\right] \times \alpha_k + \beta_k \quad (7)$$

where $\oplus$ denotes the feature concatenation operation.

To further achieve feature fusion within local regions, a K-pooling weighted aggregation mechanism is introduced to fuse features inside the local neighborhood. The aggregated neighborhood feature representation $\mathbf{F}_p$ is calculated as:

$$\mathbf{F}_p = \sum_{i=1}^{k} \frac{\exp(\mathbf{F}_{k^i})}{\sum_{j=1}^{k}\exp(\mathbf{F}_{k^j})} \cdot \mathbf{F}_{k^i} \quad (8)$$

Finally, to maintain feature dimensional consistency and enhance model stability, a shared convolutional layer and residual connection structure are introduced for feature enhancement. The computation is then formulated as follows:

$$\mathbf{F}_{C'} = \text{SiLU}\big(\text{Conv}(\text{Norm}(\mathbf{F}_p))\big) + \mathbf{F}_C \quad (9)$$

where $\mathbf{F}_{C'}$ denotes the updated center point features output by the LGEM.

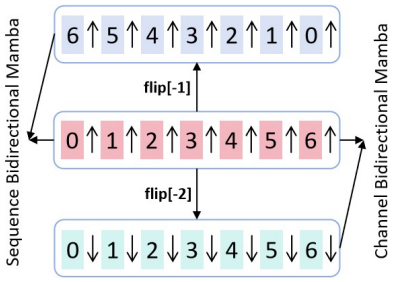

Fig. 5. The architecture of the alternating bidirectional Mamba module.

D. Alternating Bidirectional Mamba Strategy

Mamba was originally unidirectional for long sequences, with bidirectional extensions later improving context modeling in NLP and vision. However, point clouds are unordered and unstructured, causing standard Mamba to lose spatial information and lack explicit channel modeling, which limits cross-dimensional semantic correlation. While Mamba3D combined forward sequence and backward channel modeling, its parallel approach underutilizes Mamba's sequential nature. To address this, we propose an alternating bidirectional Mamba strategy (see Fig. 5): odd layers perform sequence bidirectional modeling, even layers focus on channel bidirectional modeling. This hierarchical alternation synergistically enhances temporal and channel representations, enabling more comprehensive capture of point cloud semantics and structure.

Specifically, given the input feature $\mathbf{F}_{C'} \in \mathbb{R}^{B\times N\times C}$, the forward SSM input to all MambaBlocks is set as $\mathbf{F}_{C'}$. For odd-numbered MambaBlocks, the point sequence dimension N is modeled as a sequence of length N, while the reversed sequence along this dimension is used as input to the backward SSM. For even-numbered MambaBlocks, the reversed feature along the channel dimension C is input to the backward SSM. In summary, the computations of both the sequence bidirectional Mamba and the channel bidirectional Mamba can be uniformly expressed by Eq. (10).

$$\begin{aligned}\text{S-bi-SSM}(\mathbf{F}) &= \mathbf{F} + \text{for-SSM}(\mathbf{F}_{\mathcal{L}}) + \text{S-back-SSM}(\mathbf{F}_{\mathcal{L}-})\\ \text{C-bi-SSM}(\mathbf{F}) &= \mathbf{F} + \text{for-SSM}(\mathbf{F}_{\mathcal{L}}) + \text{C-back-SSM}(\mathbf{F}_{\mathcal{C}-})\end{aligned} \quad (10)$$

where $\mathbf{F}$ be the input feature tensor. $\mathbf{F}_{\mathcal{L}}$ preserves the original order along both sequence and channel dimensions; $\mathbf{F}_{\mathcal{L}-}$ reverses the sequence dimension; $\mathbf{F}_{\mathcal{C}-}$ reverses the channel dimension. $\text{for\_SSM}(\cdot)$ is the forward SSM; $\text{S\_back\_SSM}(\cdot)$ and $\text{C\_back\_SSM}(\cdot)$ are the backward SSMs along sequence and channel dimensions, respectively. S-bi-SSM($\mathbf{F}$) with the bidirectional sequence SSM output, and C-bi-SSM($\mathbf{F}$) combines original input with the bidirectional channel SSM output.

IV. Experiments

A. Experimental Details

Datasets. We validate our model on three benchmarks: ModelNet40 [15], ScanObjectNN [16], and ShapeNetPart [17]. ModelNet40 is a synthetic dataset containing 12,311 CAD models across 40 categories, serving as a standard classification benchmark. ScanObjectNN, derived from real indoor scans, includes about 15,000 objects in 15 categories with subsets of varying noise and occlusion (OBJ_ONLY, OBJ_BG, PB_T50_RS), making it more challenging. ShapeNetPart comprises 16,880 models from 16 categories with 50 semantic parts, used for segmentation evaluation.

TABLE I
Comparison of classification accuracies (%) on the ModelNet40 dataset, with the best performance highlighted in bold.

Method	Backbone	Params (M)	FLOPs (G)	ModelNet40
PointNet [1]	MLP	3.5	0.5	89.2
PointNet++ [2]	MLP	1.5	1.7	90.7
DGCNN [3]	MLP	1.8	2.4	92.2
PointCNN [4]	MLP	0.6	-	92.9
PointNeXt [19]	MLP	1.4	3.6	92.9
GBNet [20]	MLP	8.8	-	93.8
PRA-Net [21]	MLP	2.3	-	93.7
MVTN [22]	MLP	11.2	43.7	93.8
PCT [5]	Transformer	2.9	2.3	93.2
Point-BERT [23]	Transformer	22.1	4.8	93.4
PointConT [18]	Transformer	-	-	93.5
PointMamba [6]	Mamba	12.3	3.6	92.4
PCM [10]	Mamba	34.2	45.0	92.6
Mamba3D [11]	Mamba	16.9	3.9	93.4
Pointramba [7]	Hybrid	19.5	-	92.8
LABMamba w/o vot.	Mamba	17.0	3.9	**94.0**
LABMamba w/ vot.	Mamba	17.0	3.9	**94.4**

Settings. Experiments are implemented in PyTorch on an NVIDIA RTX 4090 GPU. The model is trained from scratch. For classification, we use 12 Mamba Blocks, 128 local regions with 32 points each, AdamW optimizer (initial lr=0.0005, weight decay=0.05), cosine annealing scheduler, batch size 32, and 300 epochs. For segmentation, the initial lr is 0.0002 and batch size is 16; other settings match the classification setup.

Computational Overhead of Geometric Features. The geometric descriptors used in the GAGE module, including normal vectors, curvature, and point density, are computed as a preprocessing step based on local neighborhoods. In practice, this process introduces only a modest overhead relative to the overall inference pipeline and can be efficiently parallelized on GPU. Therefore, it does not significantly affect the efficiency of the proposed framework.

B. Experiment on ModelNet40

As shown in Table I, a systematic comparison was conducted between the proposed LABMamba method and representative MLP-based approaches, Transformer-based methods, and Mamba-based methods. The results demonstrate that LABMamba consistently outperforms existing mainstream Transformer-based methods in overall classification accuracy on the ModelNet40 dataset. Even without employing a voting scheme, LABMamba achieves a 0.5% improvement over the current best-performing Transformer model, PointConT [18], and a 1.6% gain compared to the baseline PointMamba [6]. Furthermore, it attains a 0.6% accuracy increase compared to the advanced Mamba architecture Mamba3D [11]. After incorporating the voting strategy, LABMamba reaches a classification accuracy of 94.4% on this dataset, thereby fully validating its effectiveness and superior performance on synthetic point cloud data.

Furthermore, we compare the training accuracy of LABMamba with that of PointMamba and Mamba3D on the Mod-

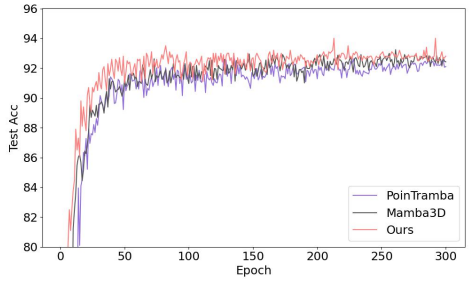

Fig. 6. Comparison of training accuracy among LABMamba, PointMamba, and Mamba3D on the ModelNet40 dataset.

elNet40 dataset (see Fig. 6). The results indicate that LABMamba not only achieves higher accuracy but also demonstrates improved convergence efficiency. Although minor fluctuations can be observed in the training curves, the model consistently reaches higher accuracy within fewer epochs.

C. Experiment on ScanObjectNN

Table II compares LABMamba with MLP-, Transformer-, and Mamba-based methods. LABMamba achieves 93.63% accuracy on OBJ_BG, surpassing PointMamba [6] by 5.33% and outperforming existing methods. With voting, accuracy improves to 95.35%, demonstrating robust feature modeling. On the more challenging PB_T50_RS subset, LABMamba reaches 91.29% (92.76% with voting), exceeding Mamba3D [11] and confirming its strong adaptability in complex real-world scenarios.

D. Experiment on ShapeNetPart

As shown in Table III, a systematic comparison was conducted between the proposed LABMamba method and representative MLP-based approaches, the Transformer-based method PointMAE, and the Mamba-based method Mamba3D. The experimental results demonstrate that LABMamba maintains robust modeling performance in point cloud semantic understanding tasks. Without employing a voting strategy, LABMamba achieves a class-average Intersection over Union (mIoU_C) of 83.8% and an instance-average Intersection over Union (mIoU_I) of 85.8%. Compared to the MLP-based method DGCNN [3], LABMamba improves mIoU_C by 1.5% and mIoU_I by 0.6%. Furthermore, it surpasses the current state-of-the-art Mamba architecture model Mamba3D [11], with gains of 0.1% in both metrics. Additionally, partial segmentation visualization results are presented in Fig. 7 for qualitative assessment. These results confirm LABMamba's effectiveness and superiority in complex semantic segmentation. The parameter increase, mainly from feature embedding and geometric integration modules, has limited impact on FLOPs but slightly raises memory usage.

E. Ablation Study

We evaluate LABMamba's core components on ModelNet40 and OBJ_BG, analyzing: (1) module effectiveness; (2) alternating bidirectional Mamba; (3) geometric descriptors; (4) point ordering.

Effectiveness of Component Modules. Table IV shows progressive integration improves performance. GAGE adds explicit geometric descriptors, improving ScanObjectNN by 1.0%. LGEM further refines features via local interaction, giving additional gains. This two-stage approach addresses geometric prior injection and feature refinement.

Specifically, GAGE raises accuracy by 0.4% on ModelNet40 and 1.0% on ScanObjectNN. Adding LGEM yields further gains of 0.4% and 0.5%, respectively.

We compare Mamba variants: Single-Seq, Bi-Seq, Bi-Channel, and full Alternating. Single-Seq struggles with long-range dependencies; Bi-Seq improves this, Bi-Channel enhances channel interactions. The full Alternating design performs best, focusing alternately on sequence and channel dimensions to reduce interference and better integrate global context (sequence) with semantic relationships (channel).

Ablation of the Alternating Bidirectional Mamba Structure. We explore alternative designs for the alternating bidirectional Mamba via ablation studies (Table V). Varying the ratio between sequence- and channel-wise bidirectional Mamba (e.g., 1:3 or 3:1) leads to performance decline, as overemphasizing one dimension disrupts balanced synergy. Parallel architectures, while theoretically possible, increase parameters and introduce redundancy due to the SSM's inherent selective filtering. Adding a gating mechanism to mitigate redundancy yielded unsatisfactory results.

Ablation of Geometric Descriptors. We ablate the three geometric descriptors (normals, curvature, and density) in the GAGE module. The complete set yields the best accuracy on both ModelNet40 and OBJ_BG (Table VI), confirming that explicit geometric priors enhance local structure representation. Notably, simply concatenating raw geometric features with point coordinates—without GAGE's convolution preprocessing and structured fusion—leads to suboptimal results, underscoring the importance of our module design. Removing normals reduces accuracy by 0.2% and 0.5% on the two datasets, respectively, highlighting their role in surface orientation. Excluding curvature causes drops of 0.3% and 0.4%, indicating its sensitivity to local shape variations. Point density removal only decreases OBJ_BG accuracy by 0.2%, reflecting its auxiliary role in capturing structural compactness. These findings robustly demonstrate that integrating explicit geometric priors with learned features improves discriminative power.

Necessity of Ordering Strategies.

We compare various ordering strategies—random ordering, Hilbert curve, Z-order, and coordinate-based ordering—finding that ordering choice has only a marginal effect on performance (Table VII). Our alternating bidirectional Mamba strategy requires no predefined ordering yet achieves superior feature extraction, consistently outperforming models dependent on explicit ordering schemes. This demonstrates stronger structural modeling and better generalization. Notably, random ordering performs best, likely because it introduces implicit stochastic regularization that prevents overfitting to specific spatial patterns, while structured orderings impose rigid lo-

TABLE II

COMPARISON OF CLASSIFICATION ACCURACIES (%) ON THREE VARIANTS OF THE SCANOBJECTNN DATASET, WITH THE BEST PERFORMANCE HIGHLIGHTED IN BOLD.

Method	Backbone	Params (M)	FLOPs (G)	OBJ_BG	OBJ_ONLY	PB_T50_RS
PointNet [1]	MLP	3.5	0.5	73.3	79.2	68.0
PointNet++ [2]	MLP	1.5	1.7	82.3	84.3	77.9
DGCNN [3]	MLP	1.8	2.4	82.8	86.2	78.1
PointCNN [4]	MLP	0.6	-	86.1	85.5	78.5
GBNet [20]	MLP	8.8	-	-	81.0	81.0
PRA-Net [21]	MLP	2.3	-	-	81.0	81.0
MVTN [22]	MLP	11.2	43.7	92.6	**92.3**	82.8
PointMLP [12]	MLP	12.6	31.4	-	-	85.4±0.3
PointNeXt [19]	MLP	1.4	3.6	-	-	87.7±0.4
P2P-HorNet [24]	MLP	-	34.6	-	-	89.3
DeLA [25]	MLP	5.3	1.5	-	-	90.4
PCT [5]	Transformer	2.9	2.3	-	-	-
Point-BERT [23]	Transformer	22.1	4.8	79.9	80.6	77.2
PointMAE [14]	Transformer	22.1	4.8	86.8	86.9	80.8
SPoTr [26]	Transformer	1.7	10.8	-	-	88.60
PointMamba [6]	Mamba	12.3	3.6	88.30	87.78	82.48
PCM [10]	Mamba	34.2	45.0	-	-	88.10±0.3
Mamba3D [11]	Mamba	16.9	3.9	92.94	90.88	91.22
LABMamba w/o vot.	Mamba	17.0	3.9	**93.63**	90.70	**91.29**
LABMamba w/ vot.	Mamba	17.0	3.9	**95.35**	92.08	**92.76**

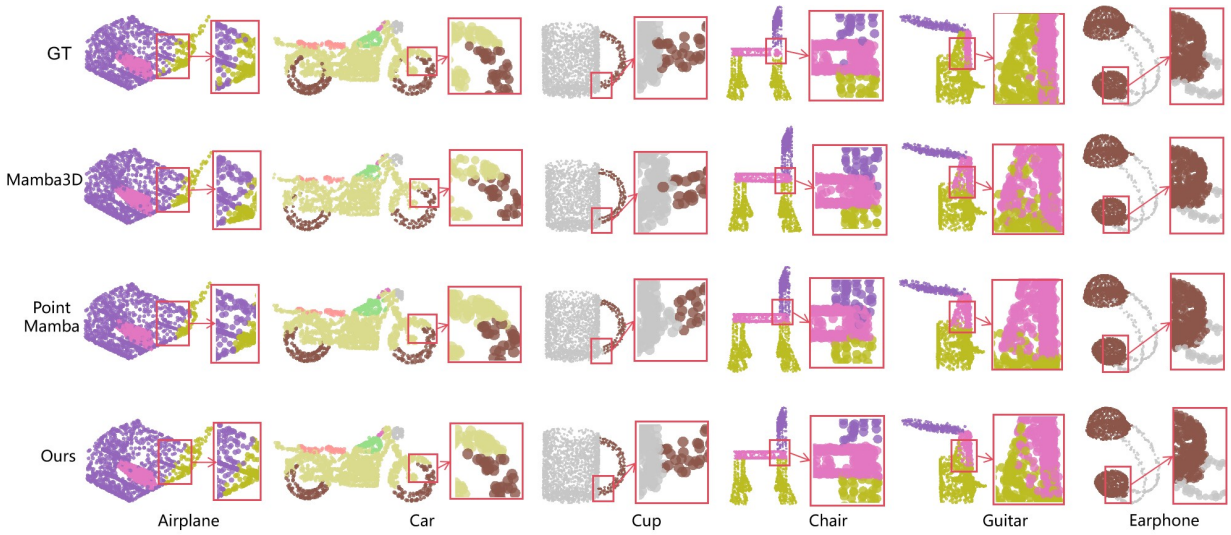

Fig. 7. The qualitative results of part segmentation of our LABMamba, PointMamba, and Mamba3D on the ShapeNetPart dataset. LABMamba provides more accurate boundary delineation in complex regions (e.g., airplane engines, chair legs) compared to Mamba3D and PointMamba, demonstrating the advantage of explicit geometric modeling through GAGE and LGEM modules.

TABLE III

COMPARISON OF SEGMENTATION ACCURACIES (%) ON THE SHAPENETPART DATASET, WITH THE BEST PERFORMANCE HIGHLIGHTED IN BOLD.

Method	Backbone	mIoU$_C$ (%)	mIoU$_I$ (%)	Params (M)
PointNet [1]	MLP	80.4	83.7	3.6
PointNet++ [2]	MLP	81.9	85.1	**1.0**
DGCNN [3]	MLP	82.3	85.2	1.3
PointMAE [14]	Transformer	83.4	85.2	1.3
Pointramba [7]	Hybrid	-	85.7	25.4
PointMamba [6]	Mamba	83.6	85.5	18.3
Mamba3D [11]	Mamba	83.7	85.7	23.0
LABMamba	Mamba	**83.8**	**85.8**	29.2

cality constraints that may limit global feature aggregation flexibility.

TABLE IV

ABLATION STUDY ON THE EFFECT OF KEY COMPONENTS IN LABMAMBA.

GAGE	LGEM	Bi-Seq	Bi-Channel	Alternating	ModelNet40 (%)	OBJ_BG (%)
					92.3	90.6
✓					92.7	91.6
✓	✓				93.1	92.1
✓	✓	✓			93.5	93.1
✓	✓		✓		93.7	93.2
✓	✓			✓	**94.0**	**93.6**

V. CONCLUSION

This paper presents LABMamba, a geometric-aware SSM framework for point cloud classification. Our main contribution integrates explicit geometric priors into the Mamba architecture via the GAGE module, addressing a key limitation

TABLE V
IMPACT OF SEQ/CHANNEL RATIO AND ARCHITECTURE ON MODEL PERFORMANCE.

Seq/Channel Ratio	ModelNet40 (%)	OBJ_BG (%)
1:3	93.8	93.1
3:1	93.5	93.3
6:6	93.4	92.7
Parallel	92.7	91.7
Alternating	**94.0**	**93.6**

TABLE VI
CONTRIBUTION OF GEOMETRIC DESCRIPTOR COMPONENTS TO MODEL PERFORMANCE.

Configuration	ModelNet40 (%)	OBJ_BG (%)
w/o only Conv features	93.6	92.9
w/o Normal vector	93.8	93.1
w/o Curvature	93.7	93.2
w/o density	94.0	93.4
Full descriptor	**94.0**	**93.6**

of existing Mamba-based methods. The proposed alternating bidirectional Mamba strategy provides a distinct approach for joint sequence-channel modeling compared to parallel designs, with reduced sensitivity to input ordering. Extensive experiments validate its effectiveness, and the performance gains justify the modest overhead from geometric computation.

LABMamba has limitations: the GAGE module's robustness to noisy or incomplete data could be enhanced, and the LGEM module's generalization to highly heterogeneous data with varying density, scale, and topology needs improvement. Future work will enhance generalization for complex scenarios like scene understanding, explore structure-adaptive mechanisms for improved flexibility, and extend to broader real-world applications.

TABLE VII
IMPACT OF ORDERING STRATEGIES ON MODEL PERFORMANCE.

Ordering Strategy	ModelNet40 (%)	OBJ_BG (%)
Hilbert ordering	93.4	93.1
Z-order ordering	93.0	92.7
xyz ordering	93.1	93.2
Random	**94.0**	**93.6**

REFERENCES

[1] C. R. Qi, H. Su, K. Mo, and L. J. Guibas, "PointNet: Deep Learning on Point Sets for 3D Classification and Segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2017, pp. 652–660. 1, 4, 6

[2] C. R. Qi, L. Yi, H. Su, and L. J. Guibas, "PointNet++: Deep Hierarchical Feature Learning on Point Sets in a Metric Space," in *Adv. Neural Inf. Process. Syst. (NeurIPS)*, vol. 30, 2017. 1, 4, 6

[3] Y. Wang, Y. Sun, Z. Liu, S. E. Sarma, M. M. Bronstein, and J. M. Solomon, "Dynamic Graph CNN for Learning on Point Clouds," *ACM Trans. Graph.*, vol. 38, no. 5, pp. 1–12, 2019. 1, 4, 5, 6

[4] Y. Li, R. Bu, M. Sun, W. Wu, X. Di, and B. Chen, "PointCNN: Convolution On X-Transformed Points," in *Adv. Neural Inf. Process. Syst. (NeurIPS)*, 2018, pp. 820–830. 1, 4, 6

[5] M.-H. Guo, J.-X. Cai, Z.-N. Liu, T.-J. Mu, R. R. Martin, and S.-M. Hu, "PCT: Point Cloud Transformer," *Comput. Vis. Media*, vol. 7, no. 2, pp. 187–199, 2021. 1, 4, 6

[6] 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," *arXiv preprint arXiv:2402.10739*, 2024. 1, 2, 4, 5, 6

[7] Z. Wang, Z. Chen, Y. Wu, Z. Zhao, L. Zhou, and D. Xu, "PoinTramba: A Hybrid Transformer-Mamba Framework for Point Cloud Analysis," *arXiv preprint arXiv:2405.15463*, 2024. 1, 4, 6

[8] J. Liu, R. Yu, Y. Wang, Y. Zheng, T. Deng, W. Ye, and H. Wang, "Point Mamba: A Novel Point Cloud Backbone Based on State Space Model with Octree-based Ordering Strategy," *arXiv preprint arXiv:2403.06467*, 2024. 2

[9] N. Köprücü, D. Okpekpe, and A. Orvieto, "NIMBA: Towards Robust and Principled Processing of Point Clouds With SSMs," *arXiv preprint arXiv:2411.00151*, 2024. 2

[10] T. Zhang, X. Li, H. Yuan, S. Ji, and S. Yan, "Point Could Mamba: Point Cloud Learning via State Space Model," *arXiv preprint arXiv:2403.00762*, 2024. 2, 4, 6

[11] X. Han, Y. Tang, Z. Wang, and X. Li, "Mamba3D: Enhancing Local Features for 3D Point Cloud Analysis via State Space Model," in *Proc. ACM Int. Conf. Multimedia*, 2024, pp. 4995–5004. 2, 4, 5, 6

[12] 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 *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2022. 1, 6

[13] X. Wu, Y. Lao, L. Jiang, X. Liu, and H. Zhao, "Point Transformer V3: Simpler, Faster, Stronger," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2024, pp. 4840–4851. 1

[14] Y. Pang, W. Wang, F. E. H. Tay, W. Liu, Y. Tian, and L. Yuan, "Masked Autoencoders for Point Cloud Self-supervised Learning," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2022, pp. 604–621. 1, 6

[15] Z. Wu, S. Song, A. Khosla, F. Yu, L. Zhang, X. Tang, and J. Xiao, "3D ShapeNets: A Deep Representation for Volumetric Shapes," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2015, pp. 1912–1920. 4

[16] M. A. Uy, Q.-H. Pham, B.-S. Hua, T. Nguyen, and S.-K. Yeung, "Revisiting Point Cloud Classification: A New Benchmark Dataset and Classification Model on Real-World Data," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, 2019, pp. 1588–1597. 4

[17] L. Yi, V. G. Kim, D. Ceylan, I.-C. Shen, M. Yan, H. Su, C. Lu, Q. Huang, A. Sheffer, and L. Guibas, "A Scalable Active Framework for Region Annotation in 3D Shape Collections," *ACM Trans. Graph.*, vol. 35, no. 6, pp. 1–12, 2016. 4

[18] Y. Liu, B. Tian, Y. Lv, L. Li, and F.-Y. Wang, "Point Cloud Classification using Content-Based Transformer via Clustering in Feature Space," *IEEE/CAA J. Autom. Sinica*, vol. 11, no. 1, pp. 231–239, 2023. 4

[19] G. Qian, Y. Li, H. Peng, J. Mai, H. A. A. K. Hammoud, M. Elhoseiny, and B. Ghanem, "PointNeXt: Revisiting PointNet++ with Improved Training and Scaling Strategies," in *Adv. Neural Inf. Process. Syst. (NeurIPS)*, vol. 35, 2022, pp. 23 192–23 204. 4, 6

[20] S. Qiu, S. Anwar, and N. Barnes, "Geometric Back-projection Network for Point Cloud Classification," *IEEE Trans. Multim.*, vol. 24, pp. 1943–1955, 2021. 4, 6

[21] S. Cheng, X. Chen, X. He, Z. Liu, and X. Bai, "Pra-Net: Point Relation-aware Network for 3D Point Cloud Analysis," *IEEE Trans. Image Process.*, vol. 30, pp. 4436–4448, 2021. 4, 6

[22] A. Hamdi, S. Giancola, and B. Ghanem, "MVTN: Multi-view Transformation Network for 3D Shape Recognition," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, 2021, pp. 1–11. 4, 6

[23] 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 *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2022, pp. 19 313–19 322. 4, 6

[24] Z. Wang, X. Yu, Y. Rao, J. Zhou, and J. Lu, "P2P: Tuning Pre-trained Image Models for Point Cloud Analysis with Point-to-Pixel Prompting," in *Adv. Neural Inf. Process. Syst. (NeurIPS)*, vol. 35, 2022, pp. 14 388–14 402. 6

[25] B. Chen, Y. Xia, Y. Zang, C. Wang, and J. Li, "Decoupled Local Aggregation for Point Cloud Learning," *arXiv preprint arXiv:2308.16532*, 2023. 6

[26] J. Park, S. Lee, S. Kim, Y. Xiong, and H. J. Kim, "Self-positioning Point-based Transformer for Point Cloud Understanding," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2023, pp. 21 814–21 823. 6