BEYOND STATIC LOCALITY: ADAPTIVE STATE SPACE MODELING FOR 3D POINT CLOUDS

Anonymous authors
Paper under double-blind review

ABSTRACT

State space models (SSMs) have recently been explored as an efficient framework for point cloud representation learning. However, existing point-cloud SSMs typically treat locality as a fixed architectural assumption, limiting their ability to adapt local modeling to diverse geometric structures and hierarchical feature representations. We propose AdaLoMamba3D, a locality-adaptive state-space framework that models locality as an adaptive property determined by learned representations. First, local interaction ranges are adapted according to representation hierarchy, enabling different representation levels to employ distinct neighborhood ranges. Center-neighbor relations are further normalized based on local feature statistics in a channel-wise manner, enabling different regions to obtain adaptive normalization references and reducing the bias introduced by uniform normalization. The resulting representations are propagated through bidirectional state-space modeling, where information exchange between complementary branches is conditioned on current token states. Together, these designs form a unified framework for adaptive locality. Experimental results show competitive and consistent performance across classification, few-shot recognition, and part segmentation.

1 INTRODUCTION

Point cloud understanding plays a fundamental role in many 3D vision applications, including autonomous driving Lang et al. (2019); Shi et al. (2020), robotic perception and manipulation Liang et al. (2019); Mousavian et al. (2019), and 3D scene understanding Qi et al. (2019); Jiang et al. (2020). Unlike images defined on regular grids, point clouds are inherently unordered, irregular, and non-uniformly distributed, requiring effective models to capture both local geometric structures and long-range dependencies. Early point-based networks Qi et al. (2017a;b) mainly focused on local feature aggregation, while recent Transformer-based approaches Yu et al. (2022); Pang et al. (2022) achieved strong representation capability through global token interaction. However, the quadratic complexity of self-attention limits their scalability to large-scale point-cloud representation learning, motivating the exploration of more efficient architectures.

Recently, state space models (SSMs), particularly Mamba Gu & Dao (2024), have emerged as efficient alternatives to Transformers due to their linear complexity and strong long-range modeling capability, motivating recent studies on their application to point cloud analysis. PointMamba Liang et al. (2024) introduces Mamba into point clouds by constructing ordered token sequences through space-filling curves, demonstrating the potential of SSMs for efficient 3D representation learning. Nevertheless, point clouds differ from sequential data, where local geometric structures play a critical role. Mamba3D Han et al. (2024) addresses this issue by incorporating Local Norm Pooling (LNP) for local feature propagation and a bidirectional SSM for enhanced global modeling, demonstrating the effectiveness of locality-aware modeling in point-cloud state-space models.

Despite these advances, existing point-cloud SSMs mainly focus on introducing local modeling into state-space architectures, while the locality itself is still largely treated as a fixed architectural design. Specifically, local feature calibration, neighborhood range, and state-space feature interaction are typically predefined and remain unchanged during representation learning. However, point clouds naturally contain heterogeneous local structures and hierarchical semantic patterns. Different regions may exhibit distinct geometric variations, shallow layers require fine-grained local details, whereas deeper layers benefit from broader contextual information. Moreover, different point tokens

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

may require different information flows during bidirectional state-space modeling. These limitations motivate an investigation into whether locality should remain a fixed prior in point-cloud SSMs, or should it adapt according to geometry, representation hierarchy, and feature states.

Motivated by these limitations, we propose AdaLoMamba3D, a locality-adaptive state-space framework for 3D point cloud learning. Rather than treating locality as a fixed architectural prior, our approach allows local modeling to vary with representation hierarchy, local feature distributions, and token states. Specifically, the neighborhood range is progressively enlarged across network depth so that shallow layers preserve fine-grained geometric details while deeper layers incorporate broader contextual information. Within each neighborhood, center-neighbor relations are calibrated using local channel-wise statistics, allowing different regions to adopt normalization references that better reflect their feature variations. In addition, the interaction between forward and reverse state-space branches is conditioned on the current token representation, making bidirectional information exchange dependent on feature states rather than a predefined fusion rule. These mechanisms jointly introduce adaptivity into neighborhood scale, local feature calibration, and state-space interaction, enabling locality to evolve throughout representation learning. Experiments on classification, few-shot recognition, and part segmentation show that the proposed framework performs consistently across different point-cloud learning settings.

Our main contributions are summarized as follows:

- We revisit locality modeling in point-cloud state-space models and investigate the limitations of static locality. We introduce an adaptive-locality formulation in which local modeling varies with geometry, representation hierarchy, and feature states.
- We introduce AdaLoMamba3D, a locality-adaptive state-space framework consisting of adaptive local neighborhood modeling and state-adaptive Mamba fusion. The proposed design jointly adapts neighborhood scale, local feature statistics, and state-space interactions while preserving the original backbone behavior through identity-preserving initialization.
- Experiments on classification, few-shot recognition, and part segmentation evaluate AdaLoMamba3D across different point-cloud learning settings, while ablation studies analyze the effect of each adaptive component.

2 Related Work

We review the most relevant studies from two aspects: point cloud representation learning and state space models for point cloud analysis.

2.1 Point Cloud Representation Learning

Point cloud representation learning has progressively evolved from point-wise feature extraction to hierarchical local modeling and global context modeling. PointNet Qi et al. (2017a) pioneered direct learning on unordered point sets, while PointNet++ Qi et al. (2017b) and subsequent methods Wang et al. (2019); Qian et al. (2022) further strengthened local geometric modeling through hierarchical neighborhoods and dynamic interactions. With the rise of Transformers, long-range dependencies among point tokens have received increasing attention. Meanwhile, self-supervised pre-training has become an important paradigm for transferable 3D representation learning, with representative methods including Point-BERT Yu et al. (2022), Point-MAE Pang et al. (2022), Point-M2AE Zhang et al. (2022), and more recent approaches such as Point-PQAE Zhang et al. (2025b). Despite their strong representation capability, Transformer-based methods typically incur quadratic complexity with respect to the number of tokens, motivating more efficient alternatives for joint local and global modeling.

2.2 State Space Models for Point Clouds

State space models (SSMs), particularly Mamba, provide efficient long-range modeling with linear complexity, motivating recent studies on point-cloud SSMs. Early methods such as Point-Mamba Liang et al. (2024) and Mamba3D Han et al. (2024) demonstrated the effectiveness of combining sequential state-space modeling with point-cloud geometric structures. Subsequent studies,

including SI-Mamba Bahri et al. (2025), StruMamba3D Wang et al. (2025), and recent methods such as CloudMamba Qu et al. (2026), further explored geometry-aware ordering, structural preservation, and more robust state-space modeling for 3D data. Despite these advances, existing methods mainly focus on point serialization, structural encoding, or state-space propagation, while local modeling is still largely based on fixed architectural configurations. Unlike prior methods that primarily modify serialization, structural encoding, or state-space propagation, our approach focuses on the adaptivity of locality itself.

3 METHOD

Figure 1 illustrates the overall architecture of AdaLoMamba3D. The model follows a local-to-global processing pipeline consisting of point tokenization, a stack of locality-adaptive Mamba blocks, and task-specific prediction heads. Built upon Mamba3D Han et al. (2024), our model adopts the point-cloud state-space backbone while introducing adaptive mechanisms from three aspects: local statistics, neighborhood scale, and bidirectional state interaction. These adaptations are organized into two stages: adaptive local neighborhood modeling, which jointly incorporates Progressive Locality Expansion (PLE) and Local Statistical Adaptation (LSA), and state-adaptive bidirectional fusion, implemented by State-Adaptive Mamba Fusion (SAF). We first introduce the overall architecture and data flow, followed by the detailed designs of the two stages.

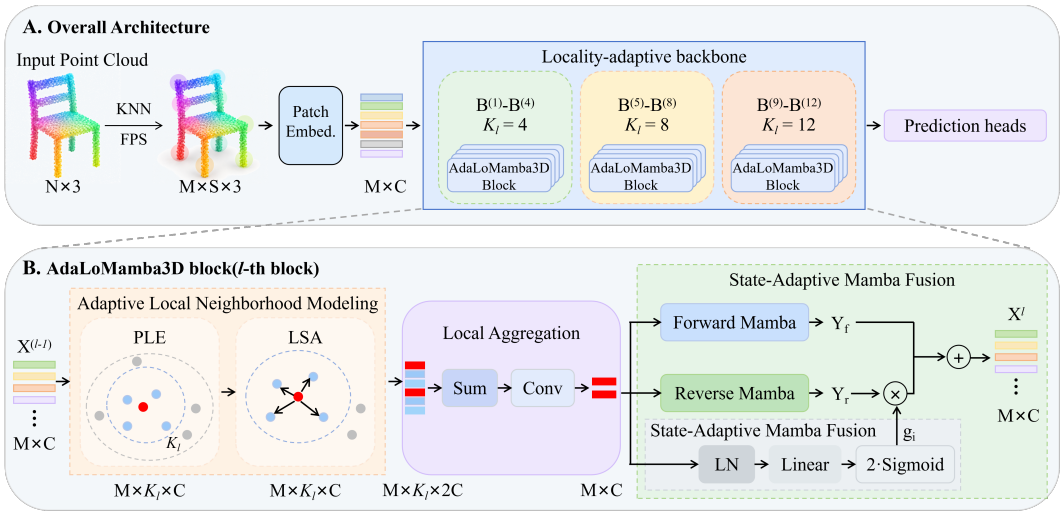


Figure 1: Overview of AdaLoMamba3D. The input point cloud is transformed into local tokens through FPS sampling, KNN grouping, and patch embedding. The proposed network progressively enlarges neighborhood ranges and adjusts local modeling according to feature representations. The AdaLoMamba3D block integrates Progressive Locality Expansion (PLE), Local Statistical Adaptation (LSA), and State-Adaptive Mamba Fusion (SAF).

3.1 OVERALL ARCHITECTURE

Point tokenization. Given an input point cloud $\mathcal{P} = \{\mathbf{p}_n\}_{n=1}^{N} \in \mathbb{R}^{N\times 3}$, we first employ farthest point sampling (FPS) to select M representative center points. For each center, its S nearest points are grouped by K-nearest neighbors (KNN) to form a local point patch. A shared patch encoder then maps each patch into a C-dimensional feature representation, producing the initial token sequence

$$\mathbf{X}^{(0)} = [\mathbf{x}_1^{(0)}, \ldots, \mathbf{x}_M^{(0)}] \in \mathbb{R}^{M\times C}. \tag{1}$$

Here, S denotes the number of raw points within each input patch.

Locality-adaptive backbone. The token sequence is subsequently processed by L locality-adaptive Mamba blocks,

$$\mathbf{X}^{(l)} = \mathcal{B}^{(l)}\left(\mathbf{X}^{(l-1)}\right), \qquad l = 1, \ldots, L, \tag{2}$$

where $\mathcal{B}^{(l)}$ denotes the l-th AdaLoMamba3D block.

Each block performs local neighborhood modeling followed by bidirectional state-space modeling. For a center token $\mathbf{x}_i^{(l-1)}$, PLE first determines a depth-dependent neighborhood size K_l and constructs the corresponding token-level neighborhood $\mathcal{N}_i^{(l)}$. Specifically, the backbone is divided into three stages with $K_l \in \{4, 8, 12\}$ for the shallow, middle, and deep stages, respectively. LSA then statistically calibrates the center–neighbor relations within $\mathcal{N}_i^{(l)}$, after which the calibrated local features are aggregated to obtain a locality-enhanced token representation. The resulting token sequence is further processed by bidirectional Mamba branches for long-range dependency modeling, and SAF generates token-dependent modulation weights for the interaction between the two state-space branches.

Prediction heads. After the final locality-adaptive block, the learned representations are fed into task-specific heads. For object and few-shot classification, the global representation is mapped to category predictions through a classification head. For part segmentation, multi-level representations are aggregated and decoded to produce point-wise predictions.

3.2 Adaptive Local Neighborhood Modeling

Local geometric structures are crucial for point cloud representation learning. However, conventional local aggregation methods usually rely on fixed neighborhood configurations and treat neighboring features with uniform contributions, which may reduce their flexibility in capturing diverse geometric patterns. To address this limitation, we introduce an Adaptive Local Neighborhood Modeling module, consisting of Progressive Locality Expansion (PLE) and Local Statistical Adaptation (LSA).

Progressive Locality Expansion. Existing point-cloud networks often employ fixed neighborhood sizes throughout the backbone, which may provide limited flexibility in balancing local geometric details and high-level contextual information. PLE introduces a depth-aware neighborhood strategy by progressively enlarging the receptive field across network stages.

For the l-th AdaLoMamba3D block, each token $\mathbf{x}_i^{(l-1)}$ constructs its local neighborhood by selecting the K_l nearest tokens:

$$\mathcal{N}_i^{(l)} = \{\mathbf{x}_{ij}^{(l-1)}\}_{j=1}^{K_l}. \tag{3}$$

In our implementation, the neighborhood sizes are set as $K_l \in \{4, 8, 12\}$ for shallow, middle, and deep stages, respectively. This progressive design provides smaller neighborhoods for early layers and larger receptive fields for deeper layers, facilitating hierarchical feature extraction.

Local Statistical Adaptation. Although PLE adaptively adjusts the spatial extent of local interactions, different local regions may still exhibit heterogeneous feature distributions due to variations in geometric structures. For example, smooth surfaces usually produce limited feature variations within a neighborhood, whereas boundary regions and corners often involve more complex local patterns with larger feature responses. Therefore, applying a shared normalization scale across different local regions may lead to suboptimal feature calibration, which may reduce the consistency of feature normalization across heterogeneous regions.

To address this issue, we introduce Local Statistical Adaptation (LSA), which performs neighborhood-specific and channel-wise calibration of center-neighbor feature relations based on local feature statistics.

Given a center token $\mathbf{x}_i^{(l-1)}$ and its j-th neighboring token $\mathbf{x}_{ij}^{(l-1)}$, we first compute their feature difference:

$$\Delta f_{ijc} = x_{ijc}^{(l-1)} - x_{ic}^{(l-1)}, \tag{4}$$

where i, j, and c denote the center token index, neighboring token index, and feature channel index, respectively. The center-neighbor difference describes the local variation pattern between neighboring tokens and the center token. Compared with directly using absolute features, this relative

representation reduces the influence of feature bias caused by different local positions and enables the model to focus on local structural variations.

Instead of adopting a shared normalization factor for all local regions, we estimate a channel-wise statistical scale for each local neighborhood:

$$\sigma_{ic} = \sqrt{\frac{1}{K_l}\sum_{j=1}^{K_l}(\Delta f_{ijc})^2 + \epsilon}, \tag{5}$$

where σ_{ic} represents the response magnitude of the c-th feature channel within the local neighborhood of the i-th center token. For each center token, the estimated statistics form a channel-wise scale vector:

$$\boldsymbol{\sigma}_i = [\sigma_{i1}, \sigma_{i2}, \cdots, \sigma_{iC}]. \tag{6}$$

Different from a single global normalization factor, $\boldsymbol{\sigma}_i$ provides a region-specific normalization reference according to local feature variation. Meanwhile, channel-wise estimation allows different feature dimensions to adapt to their individual response distributions.

The center-neighbor relation is then calibrated as:

$$\widehat{\Delta f}_{ijc} = \frac{\Delta f_{ijc}}{\sigma_{ic}}. \tag{7}$$

Through this operation, each feature channel is normalized according to its local variation magnitude, reducing the influence of heterogeneous response scales among different neighborhoods while preserving the relative variation patterns inside each local structure.

After statistical calibration, the normalized local relation is concatenated with the center feature to retain both relative neighborhood information and center semantics:

$$\mathbf{g}_{ij} = [\widehat{\Delta \mathbf{f}}_{ij}, \mathbf{x}_i^{(l-1)}], \tag{8}$$

where $\mathbf{g}_{ij} \in \mathbb{R}^{2C}$ denotes the enhanced representation of the j-th neighboring token.

The enhanced neighborhood features are further aggregated along the neighborhood dimension:

$$\boldsymbol{\alpha}_{ij} = \frac{\exp(\mathbf{g}_{ij})}{\sum_{m=1}^{K_l}\exp(\mathbf{g}_{im})}, \tag{9}$$

where $\boldsymbol{\alpha}_{ij} \in \mathbb{R}^{2C}$ represents the channel-wise contribution of the j-th neighboring feature.

The local representation is obtained by weighted aggregation:

$$\mathbf{h}_i^{(l)} = \sum_{j=1}^{K_l} \boldsymbol{\alpha}_{ij} \odot \mathbf{g}_{ij}, \tag{10}$$

where $\odot$ denotes channel-wise multiplication.

Finally, the aggregated feature is projected back to the original embedding dimension:

$$\widetilde{\mathbf{x}}_i^{(l)} = \phi\left(\mathbf{h}_i^{(l)}\right), \tag{11}$$

where $\phi(\cdot)$ denotes a learnable feature projection.

3.3 State-Adaptive Mamba Fusion

After local relation modeling and aggregation, the locality-enhanced token representations are passed to the bidirectional Mamba module for long-range dependency modeling. Following Mamba3D Han et al. (2024), two state-space branches process the token sequence along complementary directions, producing $\mathbf{y}_i^f$ and $\mathbf{y}_i^r$ for the i-th token. A conventional fusion directly combines the two branches as

$$\mathbf{y}_i = \mathbf{y}_i^f + \mathbf{y}_i^r. \tag{12}$$

Although simple and effective, this fixed fusion assigns the same branch interaction pattern to all tokens, regardless of their current feature states.

To enable input-dependent state interaction, we introduce State-Adaptive Mamba Fusion (SAF). Given the input token $\mathbf{x}_i$, SAF predicts a channel-wise modulation vector

$$\mathbf{g}_i = s \cdot \sigma\left(\mathbf{W}_g \,\mathrm{LN}(\mathbf{x}_i) + \mathbf{b}_g\right), \tag{13}$$

where $\mathrm{LN}(\cdot)$ denotes LayerNorm, $\sigma(\cdot)$ is the sigmoid function, and s controls the modulation range. The two state-space branches are then fused as

$$\mathbf{y}_i = \mathbf{y}_i^f + \mathbf{g}_i \odot \mathbf{y}_i^r, \tag{14}$$

Thus, the contribution of the complementary state-space branch becomes token-dependent.

SAF is initialized with an identity-preserving strategy. We set

$$s = 2, \qquad \mathbf{W}_g = \mathbf{0}, \qquad \mathbf{b}_g = \mathbf{0}, \tag{15}$$

which gives

$$\mathbf{g}_i = 2\sigma(0) = 1. \tag{16}$$

Therefore, at initialization,

$$\mathbf{y}_i = \mathbf{y}_i^f + \mathbf{y}_i^r, \tag{17}$$

exactly recovering the original Bi-Mamba fusion rule. This initialization reduces the potential disturbance to the pretrained feature mapping at the beginning of optimization while allowing the model to gradually learn token-dependent branch interactions during training.

4 Experiments

In this section, we evaluate AdaLoMamba3D on object classification, few-shot recognition, and part segmentation, followed by ablation studies on its architectural components, neighborhood schedules, and fusion strategies.

4.1 Implementation details.

We employ a 12-block backbone with feature dimension $C = 384$. For the pre-training, we utilize the ShapeNet dataset Chang et al. (2015) with 1,024 input points per shape. Each point cloud is partitioned into 64 local groups with 32 points per group. We use AdamW as the optimizer with an initial learning rate of 0.001 and a weight decay of 0.05. A cosine learning-rate scheduler is adopted for 300 epochs, with the first 10 epochs used for warm-up. The total batch size is set to 128. For all downstream experiments involving pre-training, including object classification, few-shot classification, and part segmentation, the backbone is initialized from the same checkpoint pretrained on ShapeNet, while task-specific prediction heads are optimized according to the corresponding downstream protocols. All experiments are implemented with PyTorch 1.13.1 and CUDA 11.7 and conducted on NVIDIA GeForce RTX 3090 GPUs with 24 GB memory.

4.2 Comparison on Downstream Tasks

Shape Classification. We evaluate our method on ScanObjectNN Uy et al. (2019) and ModelNet40 Wu et al. (2015) under both supervised training and pre-training settings. As shown in Table 1, when trained from scratch, our method improves over SI-Mamba by 1.73, 1.38, and 3.99 percentage points on OBJ_BG, OBJ_ONLY, and PB_T50_RS, respectively. With pre-training, our method

Table 1: **Shape Classification on ScanObjectNN and ModelNet40 Datasets**. The best results are highlighted in bold. #P and #F denote the number of parameters (M) and FLOPs (G), respectively.

Reference	Method	Backbone	ScanObjectNN			ModelNet40 (1k pts)		#P (M)	#F (G)
			OBJ_BG	OBJ_ONLY	PB_T50_RS	w/o Vote	w/ Vote		
Supervised Learning Only									
CVPR'17	PointNet Qi et al. (2017a)	–	73.3	79.2	68.0	89.2	–	3.5	0.5
NeurIPS'17	PointNet++ Qi et al. (2017b)	–	82.3	84.3	77.9	90.7	–	1.5	1.7
TOG'19	DGCNN Wang et al. (2019)	–	82.8	86.2	78.1	92.9	–	1.8	2.4
NeurIPS'22	PointNeXt Qian et al. (2022)	–	–	–	87.7	92.9	–	1.4	3.6
JAS'23	PointConT Liu et al. (2023)	–	–	–	88.0	93.5	–	–	–
NeurIPS'24	PointMamba Liang et al. (2024)	–	88.30	87.78	82.48	–	–	12.3	3.6
ACM MM'25	HydraMamba Qu et al. (2025)	–	–	–	88.3	**94.0**	–	–	–
CVPR'25	SI-Mamba Bahri et al. (2025)	–	92.25	91.39	87.30	92.7	–	12.3	3.6
AAAI'25	PCM Zhang et al. (2025a)	–	–	–	88.1	93.4	–	34.2	45.0
–	**Ours**	–	**93.98**	**92.77**	**91.29**	93.7	–	17.1	4.0
Pre-training using Single-Modal Information									
ECCV'22	MaskPoint Liu et al. (2022)	Transformer	89.30	88.10	84.30	93.8	93.2	22.1	–
CVPR'22	Point-BERT Yu et al. (2022)	Transformer	87.43	88.12	83.07	92.7	93.2	23.8	4.8
ECCV'22	Point-MAE Pang et al. (2022)	Transformer	92.77	91.22	89.04	92.7	93.8	23.8	4.8
NeurIPS'23	PointGPT-S Chen et al. (2023)	Transformer	93.39	92.43	89.17	93.3	94.0	29.2	5.7
AAAI'24	Point-FEMAE Zha et al. (2024)	Transformer	95.18	93.29	90.22	94.0	94.5	27.4	3.6
ICLR'24	MaskFeat3D Yan et al. (2024)	Transformer	92.7	92.0	88.6	–	–	–	–
ICCV'25	Point-PQAE Zhang et al. (2025b)	Transformer	95.0	93.6	89.6	94.0	–	22.1	–
NeurIPS'24	PointMamba Liang et al. (2024)	Mamba	94.32	92.60	89.31	93.6	94.1	12.3	3.6
ACM MM'24	Mamba3D Han et al. (2024)	Mamba	93.12	92.08	92.05	94.7	95.1	16.9	3.9
CVPR'25	SI-Mamba Bahri et al. (2025)	Mamba	94.32	91.91	89.10	93.4	–	12.3	3.6
ICCV'25	StruMamba3D Wang et al. (2025)	Structural Mamba	**95.18**	93.63	92.75	95.1	95.4	15.8	4.0
AAAI'26	CloudMamba Qu et al. (2026)	Mamba	–	–	88.30	93.7	–	10.0	1.2
–	**Ours**	Mamba	**95.18**	**94.84**	**94.55**	**95.7**	**96.0**	17.1	4.0

Table 2: **Few-Shot Classification on ModelNet40**. The best results are highlighted in bold.

Method	5-way		10-way	
	10-shot	20-shot	10-shot	20-shot
Supervised Learning Only				
PointNet	52.0 ± 3.8	57.8 ± 4.9	46.6 ± 4.3	35.2 ± 4.8
DGCNN	31.6 ± 2.8	40.8 ± 4.6	19.9 ± 2.1	16.9 ± 1.5
Transformer	87.8 ± 5.2	93.3 ± 4.3	84.6 ± 5.5	89.4 ± 6.3
With Self-supervised Pre-training				
Point-BERT	94.6 ± 3.1	93.9 ± 3.1	86.4 ± 5.4	91.3 ± 4.6
MaskPoint	95.0 ± 3.7	97.2 ± 1.7	91.4 ± 4.0	92.7 ± 5.1
Point-MAE	96.3 ± 2.5	97.8 ± 1.8	92.6 ± 4.1	95.0 ± 3.0
Point-M2AE	96.8 ± 1.8	98.3 ± 1.4	92.3 ± 4.5	95.0 ± 3.0
MaskFeat3D	97.1 ± 2.1	98.4 ± 1.6	93.4 ± 3.8	95.7 ± 3.4
PointMamba	96.9 ± 2.0	$\mathbf{99.0 \pm 1.1}$	93.0 ± 4.4	95.6 ± 3.2
Mamba3D	96.4 ± 2.2	98.2 ± 1.2	92.4 ± 4.1	95.2 ± 2.9
SI-Mamba	96.4 ± 2.7	98.5 ± 1.5	92.0 ± 5.1	95.1 ± 3.6
Ours	$\mathbf{97.5 \pm 2.4}$	$\mathbf{99.0 \pm 0.9}$	$\mathbf{93.6 \pm 4.1}$	$\mathbf{95.9 \pm 2.9}$

Table 3: **Part Segmentation on ShapeNetPart.** We report the mean IoU over all classes (mIoU_c) and all instances (mIoU_i).

Method	Cls. mIoU (%)	Ins. mIoU (%)
Supervised Learning Only		
PointNet	80.4	83.7
PointNet++	81.9	85.1
DGCNN	82.3	85.2
With Self-supervised Pre-training		
Point-MAE	84.2	86.1
Point-BERT	84.1	85.6
PointGPT-S	84.1	86.2
PointMamba	84.4	86.0
Mamba3D	83.6	85.6
Point-PQAE	84.6	86.1
SI-PointMamba	–	86.1
CloudMamba	–	**86.6**
Ours	**84.9**	86.5

achieves 95.18%, 94.84%, and 94.55% on the three ScanObjectNN variants, exceeding Mamba3D by 2.06, 2.76, and 2.50 percentage points, respectively. On ModelNet40, it reaches 95.7% without voting and 96.0% with voting, yielding the highest or tied-highest accuracy among the methods reported in Table 1. Overall, these results show that AdaLoMamba3D maintains strong classification performance across both synthetic and real-world point-cloud benchmarks.

Few-shot Learning on ModelNet40. We conduct few-shot classification on ModelNet40 under the standard n-way, m-shot protocol, using 10 independent runs for each setting and reporting the mean accuracy and standard deviation. Under the four few-shot settings in Table 2, our method improves over Mamba3D by 1.1, 0.8, 1.2, and 0.7 percentage points under the 5-way 10-shot, 5-way 20-shot, 10-way 10-shot, and 10-way 20-shot settings, respectively. These results indicate consistent improvements across different few-shot configurations.

Part Segmentation on ShapeNetPart. We evaluate our method on ShapeNetPart Yi et al. (2016) using class-average and instance-average mIoU. As shown in Table 3, our method achieves 84.9% Cls. mIoU and 86.5% Ins. mIoU, improving over Mamba3D by 1.3 and 0.9 percentage points, respectively. Compared with PointMamba, it obtains gains of 0.5 percentage points on both metrics.

Among the methods reported in Table 3, our method achieves the highest Cls. mIoU, while its Ins. mIoU remains close to the best reported result.

4.3 Ablation Studies

We conduct ablation studies on the ScanObjectNN PB_T50_RS split to analyze the contribution of each proposed component. Specifically, we investigate the architectural design, neighborhood scheduling strategy, and bidirectional fusion mechanism.

Architecture Ablation. We evaluate the contribution of Progressive Locality Expansion (PLE), Local Statistical Adaptation (LSA), and State-Adaptive Mamba Fusion (SAF) individually and in different combinations. The results are reported in Table 4. When introduced individually, PLE, LSA, and SAF achieve 93.68%, 93.75%, and 93.89% OA, respectively, compared with 93.19% for the baseline. Combining these components leads to further improvements, with the complete model reaching 94.55% OA, an increase of 1.36 percentage points over the baseline. The results also show that the three components provide complementary gains when used together.

Table 4: Ablation on architecture components.

PLE	LSA	SAF	OA(%)
×	×	×	93.19
✓	×	×	93.68
×	✓	×	93.75
×	×	✓	93.89
✓	✓	×	94.17
×	✓	✓	94.45
✓	×	✓	94.24
✓	✓	✓	**94.55**

Effect of Progressive Locality Expansion. We analyze the neighborhood scheduling strategy of PLE by comparing different K configurations. Table 5 compares four neighborhood schedules, with all fixed schedules yielding lower OA than the progressive one. Specifically, the $[4, 8, 12]$ configuration achieves 94.55% OA, compared with 93.72%, 94.24%, and 94.09% for the fixed $[4, 4, 4]$, $[8, 8, 8]$, and $[12, 12, 12]$ schedules, respectively.

Table 5: Ablation on neighborhood schedule.

Schedule	OA(%)
[4,4,4]	93.72
[8,8,8]	94.24
[12,12,12]	94.09
[4,8,12]	**94.55**

Table 6: Ablation on state-adaptive fusion.

Fusion	OA(%)
Base	94.17
Fixed	94.06
SAF	**94.55**

Effect of State-Adaptive Fusion. We further compare different fusion schemes for bidirectional Mamba features. The baseline directly combines the forward and reverse state-space features, while the fixed strategy assigns a constant coefficient of 0.5 to the reverse branch across all tokens and channels. The fusion ablation in Table 6 shows that fixed scaling slightly decreases the OA from 94.17% to 94.06%, whereas SAF achieves 94.55%, improving over the baseline by 0.38 percentage points. These results indicate that input-dependent modulation is more effective than the tested fixed fusion strategy for bidirectional state-space interaction.

5 Conclusion

In this paper, we present AdaLoMamba3D, a locality-adaptive state-space framework for 3D point cloud representation learning. By introducing adaptive mechanisms into local feature aggregation and state-space interaction, our approach enables point cloud models to better accommodate diverse geometric structures and feature representations while maintaining the efficiency of the Mamba-based backbone. Extensive experiments across diverse point-cloud benchmarks and downstream settings demonstrate the effectiveness of the proposed framework, and ablation studies further validate the contribution of each component. Future work will explore more flexible adaptive modeling strategies for large-scale point cloud understanding and investigate the extension of adaptive state-space architectures to broader 3D perception scenarios.

AI USE STATEMENT

In this work, we used generative AI tools to assist with language polishing and the retrieval and discovery of relevant literature. All AI-assisted content and retrieved references were reviewed and verified by the authors. We take responsibility for the final content of this work.

REFERENCES

Ali Bahri, Moslem Yazdanpanah, Mehrdad Noori, Sahar Dastani, Milad Cheraghalikhani, Gustavo Adolfo Vargas Hakim, David Osowiechi, Farzad Beizaee, Ismail Ben Ayed, and Christian Desrosiers. Spectral informed mamba for robust point cloud processing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 11799–11809, 2025.

Angel X Chang, Thomas Funkhouser, Leonidas Guibas, Pat Hanrahan, Qixing Huang, Zimo Li, Silvio Savarese, Manolis Savva, Shuran Song, Hao Su, Jianxiong Xiao, Li Yi, and Fisher Yu. Shapenet: An information-rich 3d model repository. *arXiv preprint arXiv:1512.03012*, 2015.

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*, pp. 29667–29679, 2023.

Albert Gu and Tri Dao. Mamba: Linear-time sequence modeling with selective state spaces. In *First Conference on Language Modeling*, 2024.

Xu Han, Yuan Tang, Zhaoxuan Wang, and Xianzhi Li. Mamba3d: Enhancing local features for 3d point cloud analysis via state space model. In *ACM Multimedia 2024*, pp. 4995–5004, 2024.

Li Jiang, Hengshuang Zhao, Shaoshuai Shi, Shu Liu, Chi-Wing Fu, and Jiaya Jia. PointGroup: Dual-set point grouping for 3d instance segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4866–4875, 2020.

Alex H. Lang, Sourabh Vora, Holger Caesar, Lubing Zhou, Jiong Yang, and Oscar Beijbom. Pointpillars: Fast encoders for object detection from point clouds. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 12689–12697, 2019.

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. In *Advances in Neural Information Processing Systems*, volume 37, 2024.

Hongzhuo Liang, Xiaojian Ma, Shuang Li, Michael Görner, Song Tang, Bin Fang, Fuchun Sun, and Jianwei Zhang. PointNetGPD: Detecting grasp configurations from point sets. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pp. 3629–3635, 2019.

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

Yahui Liu, Bin Tian, Yisheng Lv, Lingxi Li, and Fei-Yue Wang. Point cloud classification using content-based transformer via clustering in feature space. *IEEE/CAA Journal of Automatica Sinica*, pp. 231–239, 2023.

Arsalan Mousavian, Clemens Eppner, and Dieter Fox. 6-DOF GraspNet: Variational grasp generation for object manipulation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 2901–2910, 2019.

Yatian Pang, Wenxiao Wang, Francis EH Tay, Wei Liu, Yonghong Tian, and Li 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*, pp. 604–621, 2022.

Charles R. Qi, Hao Su, Kaichun Mo, and Leonidas J. Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 77–85, 2017a.

Charles R. Qi, Li Yi, Hao Su, and Leonidas 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*, pp. 5105–5114, 2017b.

Charles R. Qi, Or Litany, Kaiming He, and Leonidas J. Guibas. Deep hough voting for 3d object detection in point clouds. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 9277–9286, 2019.

Guocheng Qian, Yuchen Li, Houwen Peng, Jinjie Mai, Hasan Hammoud, Mohamed Elhoseiny, and Bernard Ghanem. Pointnext: Revisiting pointnet++ with improved training and scaling strategies. In *Advances in Neural Information Processing Systems (NeurIPS)*, pp. 23192–23204, 2022.

Kanglin Qu, Pan Gao, Qun Dai, and Yuanhao Sun. Hydramamba: Multi-head state space model for global point cloud learning. In *Proceedings of the 33rd ACM International Conference on Multimedia*, pp. 333–342, 2025.

Kanglin Qu, Pan Gao, Qun Dai, Zhanzhi Ye, Rui Ye, and Yuanhao Sun. Cloudmamba: Grouped selective state spaces for point cloud analysis. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pp. 8659–8667, 2026.

Shaoshuai Shi, Chaoxu Guo, Li Jiang, Zhe Wang, Jianping Shi, Xiaogang Wang, and Hongsheng Li. PV-RCNN: Point-voxel feature set abstraction for 3d object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 10526–10535, 2020.

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 (ICCV)*, pp. 1588–1597, 2019.

Chuxin Wang, Yixin Zha, Wenfei Yang, and Tianzhu Zhang. Strumamba3d: Exploring structural mamba for self-supervised point cloud representation learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 28546–28555, 2025.

Yue Wang, Yongbin Sun, Ziwei Liu, Sanjay E. Sarma, Michael M. Bronstein, and Justin M. Solomon. Dynamic graph cnn for learning on point clouds. *ACM Transactions on Graphics*, 38(5):1–12, 2019.

Zhirong Wu, Shuran Song, Aditya Khosla, Fisher Yu, Linguang Zhang, Xiaoou Tang, and Jianxiong Xiao. 3d shapenets: A deep representation for volumetric shapes. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1912–1920, 2015.

Siming Yan, Yuqi Yang, Yuxiao Guo, Hao Pan, Peng-shuai Wang, Xin Tong, Yang Liu, and Qixing Huang. 3d feature prediction for masked-autoencoder-based point cloud pretraining. In *International Conference on Learning Representations*, 2024.

Li Yi, Vladimir G. Kim, Duygu Ceylan, I-Chao Shen, Mengyan Yan, Hao Su, Cewu Lu, Qixing Huang, Alla Sheffer, and Leonidas Guibas. A scalable active framework for region annotation in 3d shape collections. *ACM Transactions on Graphics*, 35(6):1–12, 2016.

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 Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 19313–19322, 2022.

Yaohua Zha, Huizhen Ji, Jinmin Li, Rongsheng Li, Tao Dai, Bin Chen, Zhi Wang, and Shu-Tao Xia. Towards compact 3d representations via point feature enhancement masked autoencoders. In *Proceedings of the AAAI conference on artificial intelligence*, pp. 6962–6970, 2024.

Renrui Zhang, Ziyu Guo, Rongyao Fang, Bin Zhao, Dong Wang, Yu Qiao, Hongsheng Li, and Peng Gao. Point-M2AE: Multi-scale masked autoencoders for hierarchical point cloud pre-training. In *Advances in Neural Information Processing Systems*, pp. 27071–27084, 2022.

Tao Zhang, Haobo Yuan, Lu Qi, Jiangning Zhang, Qianyu Zhou, Shunping Ji, Shuicheng Yan, and Xiangtai Li. Point cloud mamba: Point cloud learning via state space model. In *Proceedings of the AAAI conference on artificial intelligence*, pp. 10121–10130, 2025a.

Xiangdong Zhang, Shaofeng Zhang, and Junchi Yan. Towards more diverse and challenging pre-training for point cloud learning: Self-supervised cross reconstruction with decoupled views. In *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 28696–28706, 2025b.