GEOMETRY-AWARE STATE SPACE MODELING FOR POINT CLOUD LENGTH EXTRAPOLATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Dense point clouds provide rich geometric details for 3D perception, but most point cloud models are trained with a fixed number of input points and struggle when test density changes. Downsampling high-density inputs to the training size may remove important structures, while retraining for each density is costly and impractical. In this paper, we study point cloud length extrapolation, where a model trained on short serialized point-token sequences is directly tested on much longer sequences with fixed parameters. We reframe point cloud density variation as token-length variation after serialization, without test-time raw-point downsampling or high-density retraining. This setting is challenging because dense 3D inputs contain redundant nearby tokens, geometry-critical regions, and token-channel interactions tied to spatial structure. To address these challenges, we propose PointLEX-Mamba, a geometry-aware state space modeling framework. It introduces Geometry-Aware Channel Routing to assign token-channel features to local and global branches, and Geo-Score Token Filtering to select critical global tokens using temporal response, geometric saliency, and spatial diversity. Experiments on ModelNet40 and ScanObjectNN show that PointLEX-Mamba reduces accuracy decay under point cloud length extrapolation relative to baseline and direct-transfer methods as input density and serialized length grow. Ablation studies further verify complementary contributions from both modules.

1 INTRODUCTION

Point clouds represent objects and scenes as spatial samples and are widely used in 3D applications, while modern sensors increasingly capture dense inputs with rich geometric detail Li et al. (2021); Guo et al. (2021b); Zollhöfer et al. (2018); Altuntas (2023). Point-wise, convolutional, and graph-based methods learn from points or local neighborhoods Qi et al. (2017a;b); Li et al. (2018); Wang et al. (2019); Transformer methods capture global context but incur quadratic cost Guo et al. (2021a); Zhao et al. (2021); Wu et al. (2022; 2024); Lai et al. (2022). State space models offer linear-complexity sequence modeling, and Mamba adds input-dependent selective scanning Gu et al. (2022); Gu & Dao (2024). Recent point-cloud Mamba models therefore serialize 3D inputs into one-dimensional token sequences for efficient global modeling Liang et al. (2024); Zhang et al. (2025); Han et al. (2024); Wang et al. (2025).

Most current point cloud models are trained with a fixed number of input points, such as 1024, which induces a fixed token length under a fixed grouping configuration. Denser inputs produce longer serialized point-token sequences than those observed during training. Downsampling to the training size may remove fine geometric details (Figure 1(a)) Lang et al. (2020), whereas retraining on dense inputs adds cost and cannot cover every possible test density (Figure 1(b)). We ask whether a low-density-trained model can directly process much denser inputs.

We reframe point cloud density variation as token-length variation after serialization, defining point cloud length extrapolation (Figure 1(c)). Under fixed serialization and grouping rules, more points from the same object or scene produce more groups and a longer serialized sequence. A model trained with 1024 points is directly tested with up to 32768 points using the same checkpoint; its architecture and learned parameters remain fixed. The goal is short-to-long generalization without test-time raw-point downsampling Lang et al. (2020) or high-density retraining Hu et al. (2020).

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.

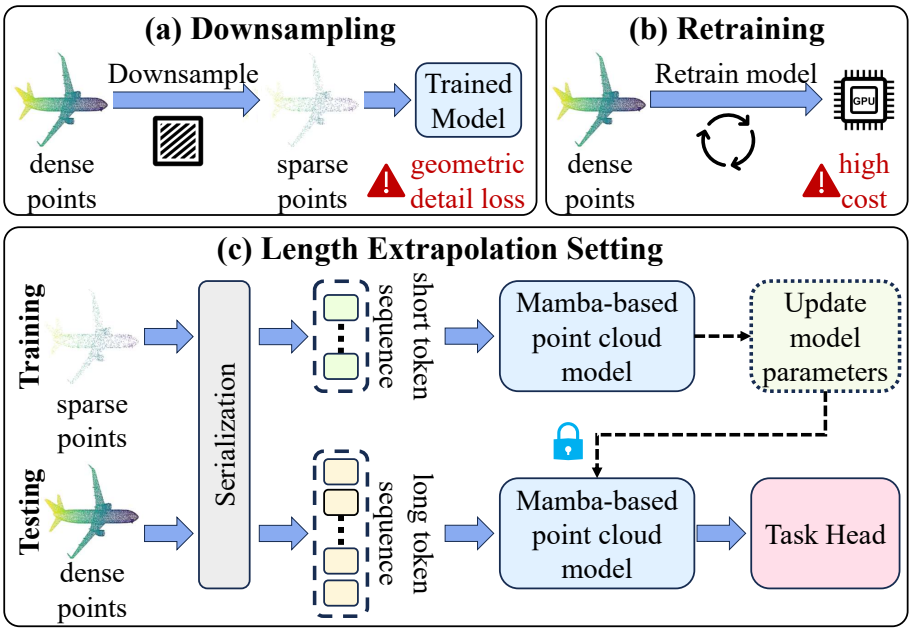

Figure 1: **Illustration of point cloud length extrapolation.** (a) Downsampling reduces dense inputs to the training size but may lose geometric details. (b) Retraining adapts the model to dense inputs but is costly. (c) Our point cloud length extrapolation setting aims to train on short serialized point-token sequences and test on much longer ones with fixed model parameters.

To achieve this goal, point cloud length extrapolation needs to address two main challenges caused by long serialized sequences and 3D geometric structure. *Firstly*, dense point clouds contain substantial token redundancy. Many nearby points describe similar local surfaces, so scanning all tokens may amplify repeated responses and dilute discriminative structures such as edges, corners, and curved regions. *Secondly*, token-channel features do not require the same amount of global modeling. Structurally complex regions may need longer-range context, while flat or redundant regions can be handled more locally. Therefore, point cloud length extrapolation should decide which tokens are worth scanning globally and which token-channel features should enter global or local modeling under fixed parameters.

To address the aforementioned challenges in point cloud length extrapolation, we propose PointLEX-Mamba, a geometry-aware state space modeling framework with two modules in each Mamba block. *Firstly*, Geometry-Aware Channel Routing assigns token-channel features to local or global branches according to learned channel decay and local geometric complexity. *Secondly*, Geo-Score Token Filtering selects critical global tokens using temporal response, geometric saliency, and spatial diversity. Selected tokens undergo global Mamba scanning, while the dense local branch preserves a path for every token. This design reduces redundant global scanning while retaining geometry-aware information. Experiments, ablations, and efficiency studies evaluate the extrapolation performance, contribution, and cost of both modules.

In summary, our contributions are three-fold:

- We study point cloud length extrapolation: a model trained on short serialized point-token sequences is directly tested on much longer ones using the same checkpoint, without test-time raw-point downsampling or high-density retraining.
- We propose PointLEX-Mamba, a geometry-aware state space modeling framework with Geometry-Aware Channel Routing for adaptive local/global modeling and Geo-Score Token Filtering for parameter-free critical-token selection.
- Experiments on synthetic and real point clouds demonstrate improved point cloud length extrapolation over baseline and direct-transfer methods; ablation and efficiency studies verify both modules with essentially unchanged parameter count and modest overhead.

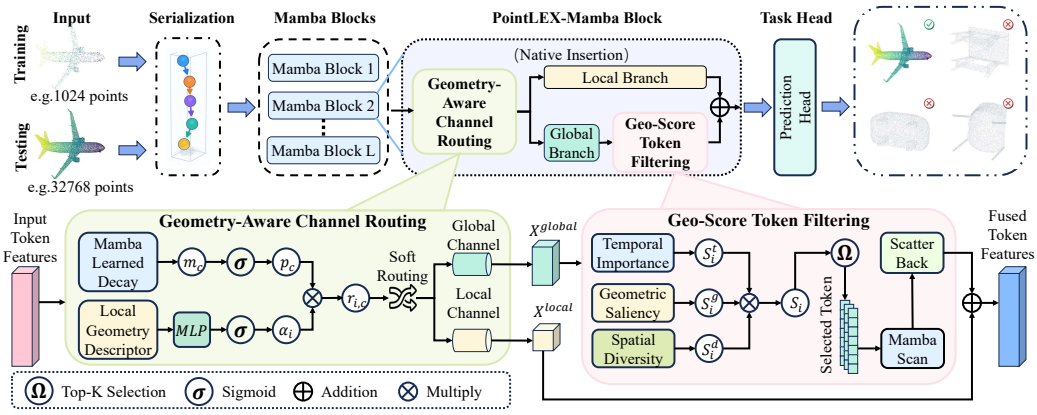

Figure 2: Overview of PointLEX-Mamba. The point cloud is serialized into a one-dimensional token sequence and processed by stacked Mamba blocks. Inside a PointLEX-Mamba block, Geometry-Aware Channel Routing softly routes token-channel features into local and global branches. Geo-Score Token Filtering then selects critical global tokens for sparse Mamba scanning. The scanned global features are scattered back to the original sequence positions and added to the dense local branch before the prediction head.

2 Related Work

3D Point Cloud Analysis. Point cloud analysis learns from unordered, irregular 3D data Guo et al. (2021b); Graham et al. (2018); Choy et al. (2019). Point-wise and hierarchical networks Qi et al. (2017a;b), CNN- and graph-based methods Li et al. (2018); Wang et al. (2019); Wu et al. (2019); Thomas et al. (2019); Tatarchenko et al. (2018); Komarichev et al. (2019); Liu et al. (2019); Yan et al. (2020); Xu et al. (2021); Qian et al. (2022), and Transformer/pre-training methods Guo et al. (2021a); Zhao et al. (2021); Wu et al. (2022; 2024); Lai et al. (2022); Yu et al. (2022); Pang et al. (2022); Zhang et al. (2022a;b); Zhu et al. (2023); Chen et al. (2023) use local aggregation, geometric operators, attention, or representation learning, but struggle on dense inputs.

State Space Models for Point Clouds. Structured and selective SSMs use recurrent transitions and input-dependent updates for linear-complexity sequence modeling Gu et al. (2022); Gu & Dao (2024). Point-cloud variants serialize 3D points before scanning and study scan order, local features, and structural representations Liang et al. (2024); Zhang et al. (2025); Han et al. (2024); Wang et al. (2025), but are usually evaluated with similar training and test lengths.

Length Extrapolation for State Space Models. Length extrapolation tests short-trained models on longer inputs. SSM approaches use token filtering Ben-Kish et al. (2025); Ye et al. (2025a), channel division Ye et al. (2025b), or changes to discretization, state evolution, and state matrices Azizi et al. (2025); Yuan et al. (2025); Lu et al. (2025); Wang (2024); related ideas address visual-token resolution variation Song et al. (2024). Unlike language, longer point-cloud sequences also change spatial density, geometry, and coverage.

3 Method

3.1 Overview

For N raw points, the tokenizer samples $G = N/16$ centers and uses forward and transposed Hilbert orders to produce $M = 2G = N/8$ tokens. Thus, training with 1024 points uses 128 tokens, while testing with 2048–32768 points uses 256–4096 tokens with the same frozen checkpoint.

As shown in Figure 2, PointLEX-Mamba modifies each PointMamba block. Geometry-Aware Channel Routing splits the projected content into dense local and global branches; Geo-Score Token Filtering selects a subset Ω of the global branch for Mamba scanning. The scanned features are scattered to their original positions and fused with the dense local branch.

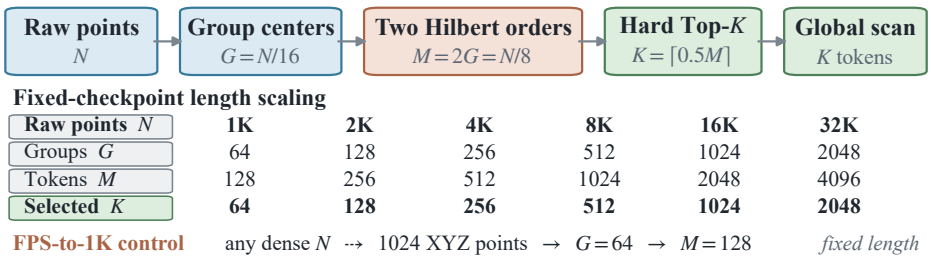

Figure 3: **Protocol of point cloud length extrapolation.** Under a frozen checkpoint, increasing raw points increases group centers, serialized tokens, and the retained PointLEX recurrence; FPS-to-1K downsamples XYZ points before grouping and fixes the sequence at 128 tokens.

For coordinates p_i and token features x_i, Hilbert serialization Skilling (2004) gives $\pi = \mathrm{argsort}_i(F_{\mathrm{sfc}}(p_i))$ and $X = [x_{\pi(1)}, \ldots, x_{\pi(M)}] \in \mathbb{R}^{M\times d}$. For compactness, we denote the Mamba block's projection and local input preprocessing Gu & Dao (2024) by

$$z, x, B, C, \Delta = \mathrm{Proj}(\mathrm{Norm}(X)). \tag{1}$$

Here, x is routed, z gates the output, and B, C, and Δ parameterize the selective scan. PointLEX scans only the selected global features and their corresponding B, C, and Δ; unselected tokens remain represented by the local branch.

3.2 Geometry-Aware Channel Routing

Geometry-Aware Channel Routing is introduced to decide where global state-space modeling is actually needed in dense serialized point clouds. A pure channel prior can identify channels with longer memory, but it is shared by all tokens and cannot distinguish a flat surface from an edge or a corner. Conversely, token geometry indicates which spatial regions are structurally complex, but does not tell which channels are suitable for long-range propagation. We therefore combine token-level geometric complexity with channel-level memory capacity, so that global modeling is assigned to both suitable channels and geometry-critical tokens under fixed parameters.

For each token, we first compute a local geometry descriptor $g_i = [\rho_i, \kappa_i, lin_i, pla_i, sct_i, anis_i]$ from its neighboring 3D coordinates Weinmann et al. (2015); Guo et al. (2021b). Here, ρ_i measures local point density, κ_i describes surface curvature, lin_i, pla_i, and sct_i indicate whether the neighborhood is line-like, plane-like, or scattered, and $anis_i$ measures directional anisotropy. These descriptors are used because geometrically complex regions usually require more contextual reasoning than smooth or densely sampled flat regions. A lightweight routing network maps g_i to a token-level geometric complexity score:

$$\alpha_i = \sigma(\mathrm{MLP}_{\mathrm{route}}(g_i)). \tag{2}$$

Here, a larger α_i indicates that token i lies in a region that is more likely to benefit from global modeling. For Mamba head h, let $a_h = \exp(A_{\log,h})$ be its learned positive decay magnitude and let $\bar{\Delta}_h$ be its mean positive time-step bias. We estimate the remaining memory after a sequence of length M and convert it to a routing prior:

$$m_h = \exp(-\bar{\Delta}_h a_h M), \qquad p_h = \sigma(\beta(m_h - \theta)). \tag{3}$$

We use fixed $\beta = 8$ and $\theta = 0.5$ in all layers and broadcast the head prior to its channels to obtain p_c. All exponent arguments are evaluated in floating point and clamped to the implementation's stable exponential range. A larger p_c indicates a channel with slower estimated memory decay. Full tensor shapes and normalization details are provided in the supplement.

The final routing should be high only when the token is geometrically important and the channel has long-range memory. We therefore combine the two cues by a token-channel routing weight:

$$r_{i,c} = \alpha_i p_c. \tag{4}$$

Instead of using a hard split, we softly decompose the projected feature into global and local branches. This avoids abruptly discarding local information and keeps the block differentiable:

$$X_{i,c}^{\text{global}} = r_{i,c}x_{i,c}, \quad X_{i,c}^{\text{local}} = (1 - r_{i,c})x_{i,c}. \tag{5}$$

In this way, global channels are emphasized when the token is geometrically complex, while simple regions and local-memory channels remain mainly in the local branch.

The local branch is not an identity copy of the raw input: the content feature has already passed through the block's depthwise local convolution before routing. It therefore provides dense local context at every token, while only the selected global subsequence enters the state-space recurrence.

Overall, Geometry-Aware Channel Routing replaces uniform scanning with geometry-conditioned allocation. It directs long-range capacity to informative regions while retaining dense local responses elsewhere, supporting point cloud length extrapolation across densities.

3.3 Geo-Score Token Filtering

Geo-Score Token Filtering is introduced because even the global branch still contains many redundant tokens in dense point clouds. We rank tokens using their state-space response, local geometric saliency, and a local-spacing coverage score, retain a fixed fraction $r = K/M$, and restore the retained indices to serialized order before scanning. Keeping a ratio rather than a fixed K ensures that the scanned recurrence itself grows from training to test time.

Following the token response idea in DeciMamba Ben-Kish et al. (2025), temporal importance is computed from the projected time-step vector Δ_i:

$$s_i^t = \|\Delta_i\|_2. \tag{6}$$

A larger s_i^t means that the token may cause a stronger state update, but this internal response alone may select spatially redundant points. We therefore add a parameter-free geometric saliency score from normalized curvature, scattering, distance to the object center, and the descriptor change between adjacent serialized tokens Weinmann et al. (2015); Ma et al. (2022):

$$s_i^g = \frac{1}{4}\left[\text{Norm}(\kappa_i) + \text{Norm}(sct_i) + \text{Norm}(dist_i) + \text{Norm}(\delta_{geo,i})\right]. \tag{7}$$

Here, Norm divides a nonnegative cue by its maximum within the sequence, $dist_i$ is the distance to the object center, and $\delta_{geo,i}$ is the descriptor difference between adjacent serialized tokens. High curvature, high scattering, large center distance, or a large local geometric change can therefore increase a token's score without introducing a second trainable scoring network.

To discourage repeatedly favoring densely sampled regions, we use normalized inverse density as a bounded spatial-coverage cue:

$$s_i^d = \text{Norm}\big((\rho_i + \epsilon)^{-1}\big), \qquad S_i = s_i^t(1 + \lambda_g s_i^g)(1 + \lambda_d s_i^d). \tag{8}$$

Following the training-free decimation principle of DeciMamba, we retain the $K = \lceil rM \rceil$ largest scores with hard Top-K. The selector has no learnable parameters and therefore requires no relaxation, gradient estimator, or auxiliary selection loss. The routing MLP is separate from this discrete selector and receives ordinary task gradients through the continuous local/global feature decomposition defined above. Selected indices are sorted into their original serialized order before the order-sensitive Mamba scan. Only the selected global features and their corresponding B, C, and Δ enter the recurrence; unselected tokens retain the dense local branch.

This sparse recurrence is an approximation to the dense SSM rather than an algebraically equivalent acceleration: skipped tokens do not contribute state transitions. The supplement reports the exact recurrence and the length-dependent scan budget induced by the fixed retention ratio.

Overall, Geo-Score Token Filtering changes global scanning from dense token processing into a parameter-free geometry-aware hard subsequence, with a Δ-based temporal score following DeciMamba. Separately, Geometry-Aware Channel Routing adapts LongMamba's cumulative-decay view into a channel-memory prior and combines it with point-cloud geometry.

Algorithm 1 PointLEX-Mamba selective SSM block

Require: $X \in \mathbb{R}^{\mathcal{B}\times M\times D}$, descriptors g, retention r
$z, x, B, C, \Delta \leftarrow \mathrm{Proj}(\mathrm{Norm}(X))$
$m_h \leftarrow \exp(-\bar{\Delta}_h \exp(A_{\log,h})M)$; $p_h \leftarrow \sigma(\beta(m_h - \theta))$
$\alpha_i \leftarrow \sigma(\mathrm{MLP}_{\mathrm{route}}(g_i))$; $R_{i,c} \leftarrow \alpha_i p_c$
$x^g \leftarrow R \odot x$; $x^l \leftarrow (1-R)\odot x$
Compute s_i^t, s_i^g, s_i^d, and S_i by Eqs. (6)–(8)
$K \leftarrow \lceil rM \rceil$; $\Omega \leftarrow \mathrm{sort}(\mathrm{TopK}(S, K).\mathrm{indices})$
$x_\Omega^g, B_\Omega, C_\Omega, \Delta_\Omega \leftarrow \mathrm{Gather}(x^g, B, C, \Delta; \Omega)$
$(\bar{A}_\Omega, \bar{B}_\Omega) \leftarrow \mathrm{discretize}(\Delta_\Omega, A, B_\Omega)$; $y_\Omega \leftarrow \mathrm{SSM}(\bar{A}_\Omega, \bar{B}_\Omega, C_\Omega)(x_\Omega^g)$
return $Y \leftarrow \mathrm{Proj}_{\mathrm{out}}(z \odot [x^l + \mathrm{Scatter}(y_\Omega; \Omega)])$

4 EXPERIMENTS

4.1 EXPERIMENT SETTINGS

Datasets. We evaluate on ModelNet40 Wu et al. (2015), which contains clean synthetic objects from 40 classes, and the real, cluttered ScanObjectNN Uy et al. (2019). For ScanObjectNN, the main comparison uses the challenging PB-T50-RS split; matched OBJ_ONLY and OBJ_BG cohort results are reported in supplementary Tables 8 and 9. Following PointMamba Liang et al. (2024), ShapeNet Chang et al. (2015) is used for pre-training when required. ModelNet40 density variants preserve the official test split through its id-to-file mapping. For ScanObjectNN, we geometry-match official test samples to raw objects and resample matched test objects only to avoid object-level leakage. All PB-T50-RS densities contain the same 2756 identities in the same label order.

Implementation Details. Experiments use RTX 4090 D GPUs, AdamW with weight decay 0.05, and 300 cosine-scheduled epochs with 10 warm-up epochs. Fine-tuning uses learning rate 3×10^{-4} and batch size 32; ShapeNet pre-training uses 10^{-3} and 128. Every method trains at 1024 points and is tested at 1024–32768 points using the same frozen checkpoint, with no high-density parameter updates. Figure 3 maps $G = 64$–2048 centers to $M = 128$–4096 serialized tokens. PointLEX retains $K = \lceil 0.5M \rceil = 64$–$2048$, so its selected global recurrence also extrapolates rather than remaining at the training length. In contrast, FPS-to-1K samples exactly 1024 XYZ points before grouping, fixes $G = 64$ and $M = 128$, and uses the dense baseline's frozen PointMamba checkpoint. The supplement specifies the repeated-run protocol and fixed retention-ratio setting.

4.2 MAIN RESULTS

Comparison Methods. Table 1 compares fixed-checkpoint PointMamba Liang et al. (2024), PointMamba with FPS, PointLEX, and PointMamba adaptations of representative one-dimensional length-extrapolation methods Ye et al. (2025b); Ben-Kish et al. (2025); Azizi et al. (2025); Lu et al. (2025); Ye et al. (2025a); Wang (2024); Yuan et al. (2025). This tests whether language-oriented token filtering, channel division, or state modifications transfer to serialized 3D inputs. All non-FPS settings increase both G and M as in Figure 3, while FPS provides a downsampling reference.

Results of ScanObjectNN Dataset. On PB-T50-RS, PointLEX matches PointMamba at the 1K training length (89.33%) and remains higher throughout extrapolation, including a gain from 26.05% to 34.87% at 32K. It has the highest values in Table 1 at 2K–8K. Direct LongMamba and DeciMamba adaptations do not consistently improve PointMamba, indicating that one-dimensional strategies do not automatically transfer to geometry-structured tokens. ReMamba is strongest at 16K–32K, so PointLEX's advantage is concentrated at short-to-medium extrapolation on this split.

Results of ModelNet40 Dataset. On ModelNet40, PointMamba falls from 89.79% at 1K to 5.63% at 32K. PointLEX matches the 1K result, reaches 85.09% at 4K and 68.84% at 8K, and remains higher at 16K and 32K, where it obtains 37.52% and 13.25%. LAMB and LongSSM equal PointMamba in this implementation, whereas PointLEX has lower 1K-to-8K and 1K-to-32K decay.

Table 1: Point cloud length extrapolation OA (%, three-run means). From 1K–32K points, $G = 64$–2048 and $M = 128$–4096; the last columns give relative OA decay.

Method	Overall Accuracy (OA, %)						Decay Rate (%)	
	1024 (train)	2048	4096	8192	16384	32768	$(OA_{1024} - OA_{8192})$ $/OA_{1024}$	$(OA_{1024} - OA_{32768})$ $/OA_{1024}$
ModelNet40								
PointMamba Liang et al. (2024)	89.79	87.68	79.94	58.06	23.83	5.63	35.34	93.73
PointMamba + FPS-to-1K	89.79	86.71	70.22	52.88	23.10	8.24	41.10	90.82
LongMamba Ye et al. (2025b)	89.79	87.52	77.91	57.85	23.51	6.47	35.57	92.79
DeciMamba Ben-Kish et al. (2025)	89.79	86.83	78.53	56.40	23.95	6.12	37.19	93.18
MambaExtend Azizi et al. (2025)	89.79	88.21	80.06	54.21	29.78	8.75	39.63	90.26
Mamba Modulation Lu et al. (2025)	89.79	88.01	79.86	47.41	9.36	5.23	47.20	94.18
LAMB Ye et al. (2025a)	89.79	87.68	79.94	58.06	23.83	5.63	35.34	93.73
LongSSM Wang (2024)	89.79	87.68	79.94	58.06	23.83	5.63	35.34	93.73
ReMamba Yuan et al. (2025)	89.79	87.44	78.24	55.43	22.53	7.21	38.27	91.97
PointLEX-Mamba (ours)	89.79	**88.28**	**85.09**	**68.84**	**37.52**	**13.25**	**23.33**	**85.24**
ScanObjectNN PB-T50-RS								
PointMamba Liang et al. (2024)	89.33	88.24	81.68	70.10	48.44	26.05	21.53	70.84
PointMamba + FPS-to-1K	89.33	88.64	76.75	66.07	48.50	23.32	26.04	73.89
LongMamba Ye et al. (2025b)	89.33	86.72	78.05	66.25	44.19	23.04	25.84	74.21
DeciMamba Ben-Kish et al. (2025)	89.33	87.19	79.28	66.62	45.39	23.91	25.42	73.23
MambaExtend Azizi et al. (2025)	89.33	88.17	81.75	69.45	46.84	25.80	22.25	71.12
Mamba Modulation Lu et al. (2025)	89.33	88.24	82.04	69.70	47.02	25.07	21.97	71.94
LAMB Ye et al. (2025a)	89.33	88.24	81.68	70.10	48.44	26.05	21.53	70.84
LongSSM Wang (2024)	89.33	88.24	81.68	70.10	48.44	26.05	21.53	70.84
ReMamba Yuan et al. (2025)	89.33	88.03	83.74	73.80	**58.64**	**36.21**	17.38	**59.46**
PointLEX-Mamba (ours)	89.33	**89.17**	**85.99**	**75.69**	56.17	34.87	**15.27**	60.96

Evaluation protocol. Table 1 reports three-run means under our fixed-checkpoint point cloud length extrapolation protocol. Dense PointMamba measures robustness as the raw input and recurrence grow, while FPS-to-1K measures the effect of returning inputs to the training size. These controls distinguish extrapolation from raw-point downsampling. At every test density, the checkpoint, parameters, and optimization state remain fixed; only the input point count and resulting group and token lengths change. Thus, gains cannot come from length-specific fine-tuning or density-specific weights. Intermediate lengths and the 1K-to-8K and 1K-to-32K decay rates expose both the onset and accumulated severity of degradation.

4.3 Ablation Study

Tables 2 and 3 report 1K-trained ablations. Table 2 shows that token filtering provides the larger individual long-length gain, while routing adds complementary improvements in the full model. At 32K, full PointLEX reaches 13.25% on ModelNet40 and 34.87% on PB-T50-RS, above either isolated module. The combined advantage is most visible beyond the training density, indicating that the two modules cooperate as sequence length grows rather than merely improving the in-distribution operating point. Table 3 then separates the cues used by both modules.

Effectiveness of Geometry-Aware Channel Routing. The right half of Table 3 separates the channel decay prior p_c and local geometry g_i. Neither cue is consistently stable alone: p_c identifies channels with long memory but cannot distinguish geometric regions, whereas g_i identifies informative tokens without modeling channel-specific memory. Their product $r_{i,c}$ conditions routing on token geometry and channel decay, giving the strongest 4K–32K ModelNet40 results and the best PB-T50-RS profile. This supports allocating global modeling capacity to token-channel pairs that are geometrically informative and suitable for long-range propagation.

Effectiveness of Geo-Score Token Filtering. The left half of Table 3 separates temporal response s_i^t, geometric saliency s_i^g, and diversity s_i^d. Complete S_i is strongest at 4K–32K on ModelNet40 and at every extrapolated PB-T50-RS length, consistent with complementary response, structure, and coverage cues. Geometric saliency is particularly useful at intermediate lengths, whereas diversity becomes increasingly competitive at the longest PB-T50-RS setting; combining them with temporal response yields the most consistent profile across densities and datasets.

Table 2: Coarse OA (%) ablation at 2K–32K for Routing, Filtering, and their PointLEX combination.

Variant	2K	4K	8K	16K	32K
ModelNet40					
PointMamba	87.68	79.94	58.06	23.83	5.63
Channel Routing	87.64	80.36	60.04	29.69	6.88
Token Filtering	86.82	82.00	63.35	34.88	10.52
PointLEX	**88.28**	**85.09**	**68.84**	**37.52**	**13.25**
ScanObjectNN PB-T50-RS					
PointMamba	88.24	81.68	70.10	48.44	26.05
Channel Routing	88.00	83.76	71.44	52.32	26.25
Token Filtering	89.71	83.52	74.11	53.63	30.77
PointLEX	**89.17**	**85.99**	**75.69**	**56.17**	**34.87**

Table 3: Fine-grained OA (%) ablations for token filtering (left) and channel routing (right).

Geo-Score Token Filtering						Geometry-Aware Channel Routing					
Variant	**2K**	**4K**	**8K**	**16K**	**32K**	**Variant**	**2K**	**4K**	**8K**	**16K**	**32K**
ModelNet40						**ModelNet40**					
s_i^t only	86.83	78.53	56.40	23.95	6.12	p_c only	87.52	77.91	57.85	23.51	6.47
s_i^g only	**88.04**	81.36	60.25	28.87	8.69	g_i only	**87.71**	79.05	59.13	28.35	5.94
s_i^d only	87.04	78.85	58.25	23.59	6.69	$r_{i,c}$	87.64	**80.36**	**60.04**	**29.69**	**6.88**
S_i	86.82	**82.00**	**63.35**	**34.88**	**10.52**						
ScanObjectNN PB-T50-RS						**ScanObjectNN PB-T50-RS**					
s_i^t only	87.19	79.28	66.62	45.39	23.91	p_c only	86.72	78.05	66.25	44.19	23.04
s_i^g only	88.13	82.73	70.51	51.41	27.44	g_i only	85.41	81.51	71.16	51.44	24.00
s_i^d only	87.32	79.28	68.16	48.90	29.98	$r_{i,c}$	**88.00**	**83.76**	**71.44**	**52.32**	**26.25**
S_i	**89.71**	**83.52**	**74.11**	**53.63**	**30.77**						

4.4 Analysis

Figure 4 visualizes the length-wise results in Table 1. PointLEX follows a smoother curve than the non-FPS baselines on ModelNet40 and remains above PointMamba, LongMamba, and DeciMamba across PB-T50-RS extrapolation lengths. This trend is consistent with the dense local branch preserving every token while the selected global branch limits redundant state transitions as density and serialized sequence length grow.

Length-dependent gains. Table 2 shows that PointLEX exceeds PointMamba by 0.60–13.69 points on ModelNet40 and 0.93–8.82 points on PB-T50-RS from 2K to 32K. The generally larger gaps beyond the training density suggest that geometry-aware routing and filtering become more useful as redundancy and sequence length increase. Absolute OA still falls at 32K, indicating mitigation rather than complete resolution of length-induced degradation.

Cross-dataset behavior. Although 1K OA is similar, PointMamba and PointLEX retain substantially higher 32K OA on PB-T50-RS than on ModelNet40 (26.05 versus 5.63 and 34.87 versus 13.25, respectively). The degradation profile is therefore dataset dependent, so the controlled within-dataset gaps are more informative than absolute OA comparisons across benchmarks.

Table 4 shows essentially unchanged parameter count, confirming that the gains do not come from increasing model size. PointLEX adds only modest measured time and estimated GFLOPs overhead while improving robustness at extrapolated lengths.

Supplementary Table 6 evaluates transfer to PCM Zhang et al. (2025) and Mamba3D Han et al. (2024). PointLEX improves 4K–32K accuracy across both backbones and datasets, showing that its routing and filtering mechanisms extend beyond the primary PointMamba backbone.

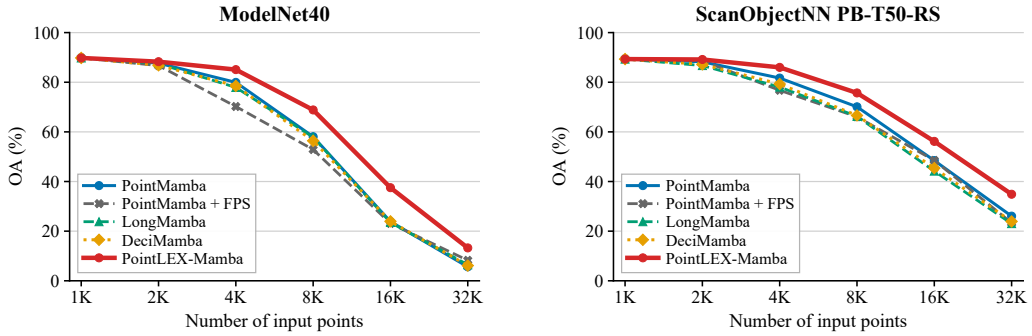

Figure 4: Point cloud length extrapolation curves on ModelNet40 (left) and ScanObjectNN PB-T50-RS (right). PointLEX-Mamba maintains a smoother accuracy profile than PointMamba, LongMamba, and DeciMamba as input density and serialized sequence length increase.

Table 4: Efficiency comparison; GFLOPs are estimates derived from measured overhead.

Metric	**Setting**	**PointMamba**	**PointLEX**
Time (s)	ModelNet40	31.70	34.07
	PB-T50-RS	29.81	32.44
Params (M)	ModelNet40	12.30	12.30
	PB-T50-RS	12.30	12.29
GFLOPs	ModelNet40	1.50	1.61
	PB-T50-RS	3.10	3.37

Discussion. As N increases, G and M grow together, so the benchmark couples density with recurrence length and measures point cloud length extrapolation rather than pure sequence extension. Dense PointMamba controls architecture and input; FPS-to-1K isolates downsampling.

PointLEX uses $r = 0.5$ as an effective accuracy–cost operating point: the selected global recurrence grows with input length, while the dense local branch preserves a feature path for every token.

Relative to PointMamba, PointLEX adds approximately 7–9% measured time and estimated GFLOPs with unchanged parameter count, so greater capacity does not explain the gains. FPS-to-1K generally trails dense PointMamba at intermediate lengths; PointLEX instead retains dense input and controls the global recurrent path at modest additional cost.

Using the same identities and label order at every density controls test-set composition, so the reported changes primarily reflect increasing point density and serialized sequence length.

Results across synthetic and real point clouds and three SSM backbones support the proposed routing and filtering strategy as density and sequence length increase.

5 Conclusion

This paper presents PointLEX-Mamba, a geometry-aware state space modeling framework for point cloud length extrapolation. Under fixed serialization, density growth becomes token-length growth, allowing a checkpoint trained with 128 tokens to process up to 4096 tokens without raw-point downsampling or high-density retraining. Geometry-Aware Channel Routing combines decay with local geometry, while Geo-Score Token Filtering selects global tokens for serialized scanning. Together, dense local and selected global paths address redundancy and geometry-dependent token-channel interactions. On ModelNet40 and ScanObjectNN, PointLEX-Mamba reduces accuracy decay as density and sequence length grow. Ablations verify both modules, with essentially unchanged parameters and modest time and estimated GFLOPs overhead.

AI USE STATEMENT

Generative AI tools assisted manuscript writing, translation, document formatting, literature search and organization, and feedback on the clarity and consistency of the method and experimental protocol. They were not used to implement the reported model, preprocess datasets, train models, analyze raw experimental data, or generate reported results. The authors manually checked AI-assisted text against the cited literature, the documented method, and the reported tables, and take responsibility for the final manuscript, technical claims, and results.

REPRODUCIBILITY STATEMENT

The main paper describes the model and experimental setup. Additional inference pseudocode and method details are provided in Sections A and B of the supplement. Supplement Section C describes the high-density test-set construction procedure, Section D lists implementation settings, and Section E contains length curves, completed protocol checks, evaluation protocols, and additional ScanObjectNN tables. The anonymous supplement also records the point-to-token mapping, object-cohort audit, and parameter-free selector audit. No executable source code, checkpoints, private paths, or raw datasets are included in the review material.

ETHICS STATEMENT

This work studies point-cloud classification and length extrapolation using the datasets and evaluation procedure described in the paper and Supplement Section C. Downstream uses of 3D perception can raise privacy and safety concerns. Deployment should assess data provenance, privacy, and reliability under density shifts and across relevant populations and environments. Use and redistribution of third-party datasets remain subject to their applicable terms and licenses. The reported benchmark results alone do not establish deployment safety.

REFERENCES

Cengiz Altuntas. Review of scanning and pixel array-based LiDAR point-cloud measurement techniques to capture 3d shape or motion. *Applied Sciences*, 13(11):6488, 2023.

Seyedarmin Azizi, Souvik Kundu, Mohammad Erfan Sadeghi, and Massoud Pedram. MambaExtend: A training-free approach to improve long context extension of mamba. In *International Conference on Learning Representations*, 2025.

Assaf Ben-Kish, Itamar Zimerman, Shady Abu-Hussein, Nadav Cohen, Amir Globerson, Lior Wolf, and Raja Giryes. DeciMamba: Exploring the length extrapolation potential of mamba. In *International Conference on Learning Representations*, 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, 2015.

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

Christopher Choy, JunYoung Gwak, and Silvio Savarese. 4d spatio-temporal convnets: Minkowski convolutional neural networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 3075–3084, 2019.

Benjamin Graham, Martin Engelcke, and Laurens van der Maaten. 3d semantic segmentation with submanifold sparse convolutional networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 9224–9232, 2018.

Albert Gu and Tri Dao. Mamba: Linear-time sequence modeling with selective state spaces. In *International Conference on Learning Representations*, 2024.

Albert Gu, Karan Goel, and Christopher Ré. Efficiently modeling long sequences with structured state spaces. In *International Conference on Learning Representations*, 2022.

Meng-Hao Guo, Jun-Xiong Cai, Zheng-Ning Liu, Tai-Jiang Mu, Ralph R. Martin, and Shi-Min Hu. PCT: Point cloud transformer. *Computational Visual Media*, 7(2):187–199, 2021a.

Yulan Guo, Hanyun Wang, Qingyong Hu, Hao Liu, Li Liu, and Mohammed Bennamoun. Deep learning for 3d point clouds: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 43(12):4338–4364, 2021b.

Xu Han, Yuan Tang, Zhaoxuan Wang, and Xianzhi Li. Mamba3D: Enhancing local features for 3d point cloud analysis via state space model. In *Proceedings of the 32nd ACM International Conference on Multimedia*, pp. 1023–1031, 2024.

Qingyong Hu, Bo Yang, Linhai Xie, Stefano Rosa, Yulan Guo, Zhihua Wang, Niki Trigoni, and Andrew Markham. RandLA-Net: Efficient semantic segmentation of large-scale point clouds. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 11108–11117, 2020.

Artem Komarichev, Zichun Zhong, and Jing Hua. A-CNN: Annularly convolutional neural networks on point clouds. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 7421–7430, 2019.

Xin Lai, Jiakai Liu, Li Jiang, Liwei Wang, Hengshuang Zhao, Shu Liu, Xiaojuan Qi, and Jiaya Jia. Stratified transformer for 3d point cloud segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 8500–8509, 2022.

Itai Lang, Asaf Manor, and Shai Avidan. SampleNet: Differentiable point cloud sampling. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 7578–7588, 2020.

Yangyan Li, Rui Bu, Mingchao Sun, Wei Wu, Xinhan Di, and Baoquan Chen. PointCNN: Convolution on x-transformed points. In *Advances in Neural Information Processing Systems*, volume 31, pp. 820–830, 2018.

Ying Li, Lingfei Ma, Zilong Zhong, Fei Liu, Michael A. Chapman, Dongpu Cao, and Jonathan Li. Deep learning for LiDAR point clouds in autonomous driving: A review. *IEEE Transactions on Neural Networks and Learning Systems*, 32(8):3412–3432, 2021.

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.

Yongcheng Liu, Bin Fan, Shiming Xiang, and Chunhong Pan. Relation-shape convolutional neural network for point cloud analysis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 8895–8904, 2019.

Peng Lu, Jerry Huang, Qiuhao Zeng, Xinyu Wang, Boxing Chen, Philippe Langlais, and Yufei Cui. Mamba modulation: On the length generalization of mamba models. In *Advances in Neural Information Processing Systems*, volume 38, pp. 22019–22064, 2025.

Xu Ma, Can Qin, Haoxuan You, Haoxi Ran, and Yun Fu. Rethinking network design and local geometry in point cloud: A simple residual MLP framework. In *International Conference on Learning Representations*, 2022.

Yatian Pang, Wenxiao Wang, Francis E. H. Tay, Wei Liu, Yonghong Tian, and Li Yuan. Masked autoencoders for point cloud self-supervised learning. In *Proceedings of the European Conference on Computer Vision*, 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 *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 77–85, 2017a.

Charles Ruizhongtai Qi, Li Yi, Hao Su, and Leonidas J. Guibas. PointNet++: Deep hierarchical feature learning on point sets in a metric space. In *Advances in Neural Information Processing Systems*, volume 30, pp. 5105–5114, 2017b.

Guocheng Qian, Yuchen Li, Houwen Peng, Jinjie Mai, Hasan Abed Al Kader Hammoud, Mohamed Elhoseiny, and Bernard Ghanem. PointNeXt: Revisiting PointNet++ with improved training and scaling strategies. In *Advances in Neural Information Processing Systems*, volume 35, pp. 23192–23204, 2022.

John Skilling. Programming the Hilbert curve. In *AIP Conference Proceedings*, volume 707, pp. 381–387, 2004.

Yiran Song, Qianyu Zhou, Xiangtai Li, Deng-Ping Fan, Xuequan Lu, and Lizhuang Ma. BA-SAM: Scalable bias-mode attention mask for segment anything model, 2024.

Maxim Tatarchenko, Jaesik Park, Vladlen Koltun, and Qian-Yi Zhou. Tangent convolutions for dense prediction in 3d. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3887–3896, 2018.

Hugues Thomas, Charles R. Qi, Jean-Emmanuel Deschaud, Beatriz Marcotegui, François Goulette, and Leonidas J. Guibas. KPConv: Flexible and deformable convolution for point clouds. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 6411–6420, 2019.

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*, 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*, pp. 28546–28555, 2025.

Shida Wang. LongSSM: On the length extension of state-space models in language modelling, 2024.

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):146:1–146:12, 2019.

Martin Weinmann, Boris Jutzi, and Clément Mallet. Semantic point cloud interpretation based on optimal neighborhoods, relevant features and efficient classifiers. *ISPRS Journal of Photogrammetry and Remote Sensing*, 105:286–304, 2015.

Wenxuan Wu, Zhongang Qi, and Li Fuxin. PointConv: Deep convolutional networks on 3d point clouds. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 9621–9630, 2019.

Xiaoyang Wu, Yixing Lao, Li Jiang, Xihui Liu, and Hengshuang Zhao. Point transformer V2: Grouped vector attention and partition-based pooling. In *Advances in Neural Information Processing Systems*, volume 35, pp. 33330–33342, 2022.

Xiaoyang Wu, Li Jiang, Peng-Shuai Wang, Zhijian Liu, Xihui Liu, Yu Qiao, Wanli Ouyang, Tong He, and Hengshuang Zhao. Point transformer V3: Simpler, faster, stronger. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 4840–4851, 2024.

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*, pp. 1912–1920, 2015.

Mutian Xu, Runyu Ding, Hengshuang Zhao, and Xiaojuan Qi. PAConv: Position adaptive convolution with dynamic kernel assembling on point clouds. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 3173–3182, 2021.

Xu Yan, Chaoda Zheng, Zhen Li, Sheng Wang, and Shuguang Cui. PointASNL: Robust point clouds processing using nonlocal neural networks with adaptive sampling. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 5589–5598, 2020.

Zhifan Ye, Zheng Wang, Kejing Xia, Jihoon Hong, Leshu Li, Lexington Whalen, Cheng Wan, Yonggan Fu, Yingyan Celine Lin, and Souvik Kundu. LAMB: A training-free method to enhance the long-context understanding of SSMs via attention-guided token filtering. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pp. 1200–1209, 2025a.

Zhifan Ye, Kejing Xia, Yonggan Fu, Xin Dong, Jihoon Hong, Xiangchi Yuan, Shizhe Diao, Jan Kautz, Pavlo Molchanov, and Yingyan Celine Lin. LongMamba: Enhancing mamba's long context capabilities via training-free receptive field enlargement. In *International Conference on Learning Representations*, 2025b.

Xumin Yu, Lulu Tang, Yongming Rao, Tiejun Huang, Jie Zhou, and Jiwen Lu. Point-BERT: Pre-training 3d point cloud transformers with masked point modeling. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 19313–19322, 2022.

Danlong Yuan, Jiahao Liu, Bei Li, Huishuai Zhang, Jingang Wang, Xunliang Cai, and Dongyan Zhao. ReMamba: Equip mamba with effective long-sequence modeling. In *Findings of the Association for Computational Linguistics: EMNLP 2025*, pp. 6830–6840, 2025.

Renrui Zhang, Ziyu Guo, Peng Gao, Rongyao Fang, Bin Zhao, Dong Wang, Yu Qiao, and Hongsheng Li. Point-M2AE: Multi-scale masked autoencoders for hierarchical point cloud pre-training. In *Advances in Neural Information Processing Systems*, volume 35, pp. 27061–27074, 2022a.

Renrui Zhang, Ziyu Guo, Wei Zhang, Kunchang Li, Xupeng Miao, Bin Cui, Yu Qiao, Peng Gao, and Hongsheng Li. PointCLIP: Point cloud understanding by CLIP. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 8552–8562, 2022b.

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*, volume 39, pp. 10121–10130, 2025.

Hengshuang Zhao, Li Jiang, Jiaya Jia, Philip H. S. Torr, and Vladlen Koltun. Point transformer. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 16259–16268, 2021.

Xiangyang Zhu, Renrui Zhang, Bowei He, Ziyu Guo, Ziyao Zeng, Zipeng Qin, Shanghang Zhang, and Peng Gao. PointCLIP V2: Prompting CLIP and GPT for powerful 3d open-world learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 2639–2650, 2023.

Michael Zollhöfer, Patrick Stotko, Andreas Görlitz, Christian Theobalt, Matthias Nießner, Reinhard Klein, and Andreas Kolb. State of the art on 3d reconstruction with RGB-D cameras. *Computer Graphics Forum*, 37(2):625–652, 2018.

TECHNICAL SUPPLEMENT

This supplement is organized as follows:

- **A. Inference Algorithm:** summarizes the test-time execution flow of PointLEX-Mamba under fixed model parameters.
- **B. Additional Method Details:** defines the local geometry descriptor, the channel-memory prior, and the complete PointLEX-Mamba block execution.
- **C. High-Density Test Set Construction:** describes how we build length-specific ModelNet40 and ScanObjectNN test sets while preserving the official test split and avoiding leakage.
- **D. Implementation Details:** reports the dataset settings, training setup, hardware, and PointLEX-Mamba hyperparameters used in the experiments.
- **E. Additional Experimental Results:** provides length curves, completed protocol checks, evaluation protocols, and additional OBJ_ONLY and OBJ_BG results.

A INFERENCE ALGORITHM

Algorithm S1 summarizes inference. We use a fixed retention ratio r and set $K = \lceil rM \rceil$ in each block. The implementation logs both M and K for every evaluation, and selected indices are restored to serialized order before scanning.

Algorithm S1: Inference Stage of PointLEX-Mamba

Input: point cloud $P \in \mathbb{R}^{N\times 3}$
Input: model parameters θ, retention ratio r
$(p, X_{\text{seq}}) \leftarrow \text{TokenizeAndSerialize}(P)$; record M
$X \leftarrow \text{Embed}(X_{\text{seq}})$
for each PointLEX-Mamba block **do**
  Project X, locally convolve content/BC features, form positive Δ
  $g_i \leftarrow \text{Geom}(p_i)$, $p_c \leftarrow \text{Mem}(c)$
  $r_{i,c} \leftarrow \alpha_i p_c$, $X^{\text{local}}, X^{\text{global}} \leftarrow x$
  $S_i \leftarrow \text{GeoScore}(x_i, g_i, \Delta_i)$
  $K \leftarrow \lceil rM \rceil$, $\Omega \leftarrow \text{SortIndices}(\text{TopK}(S, K))$
  $(X_\Omega^{\text{global}}, B_\Omega, C_\Omega, \Delta_\Omega) \leftarrow (X^{\text{global}}, B, C, \Delta)[\Omega]$
  $Y_\Omega \leftarrow \text{Scan}(X_\Omega^{\text{global}}, B_\Omega, C_\Omega, \Delta_\Omega)$
  $Y^{\text{global}} \leftarrow \text{Scatter}(Y_\Omega, \Omega)$
  $Y \leftarrow X^{\text{local}} + Y^{\text{global}}$
  $X \leftarrow X + \text{OutProj}(\text{GatedRMSNorm}(Y, z))$
end for
Return: classification prediction

B ADDITIONAL METHOD DETAILS

Local Geometry Descriptor. For each serialized token center p_i, we compute a local descriptor from its k nearest token centers. In our implementation, $k = 8$. Let $\mathcal{N}_i$ denote the neighbor set and let $\bar{p}_i$ be its mean coordinate. We form the local covariance matrix

$$\Sigma_i = \frac{1}{k-1} \sum_{p_j \in \mathcal{N}_i} (p_j - \bar{p}_i)(p_j - \bar{p}_i)^\top. \tag{9}$$

Let $\lambda_{i,1} \geq \lambda_{i,2} \geq \lambda_{i,3} \geq 0$ be the eigenvalues of Σ_i. The descriptor used in the main paper is

$$g_i = [\rho_i, \kappa_i, lin_i, pla_i, sct_i, anis_i], \tag{10}$$

where

$$\rho_i = \left(\frac{1}{k-1} \sum_{p_j \in \mathcal{N}_i \setminus p_i} \|p_i - p_j\|_2 + \epsilon \right)^{-1}, \tag{11}$$

$$\kappa_i = \frac{\lambda_{i,3}}{\lambda_{i,1} + \lambda_{i,2} + \lambda_{i,3} + \epsilon}, \tag{12}$$

$$lin_i = \frac{\lambda_{i,1} - \lambda_{i,2}}{\lambda_{i,1} + \epsilon}, \quad pla_i = \frac{\lambda_{i,2} - \lambda_{i,3}}{\lambda_{i,1} + \epsilon}, \tag{13}$$

$$sct_i = \frac{\lambda_{i,3}}{\lambda_{i,1} + \epsilon}, \quad anis_i = \frac{\lambda_{i,1} - \lambda_{i,3}}{\lambda_{i,1} + \epsilon}. \tag{14}$$

The six descriptor dimensions are divided by their maximum absolute values within the sequence, with an epsilon floor. The available implementation computes these coordinate descriptors without gradients, although the MLP consuming them can still receive gradients through continuous routing. This normalization is a scale-control heuristic, not a proof of robustness to outliers or coordinate noise.

Geometry-Preprocessing Complexity. Neighbor search uses chunked `torch.cdist`: for chunk width b, it uses $O(bM)$ temporary distance memory and $O(M^2)$ total distance work, followed by $O(Mk^3)$ covariance eigendecomposition for fixed $k = 8$. The sparse Mamba recurrence is linear in the retained length, and the efficiency protocol accounts for the complete geometric preprocessing and scanning path.

Channel Memory Prior. The routing module combines token-level geometry with a channel-level long-memory prior. For a Mamba head h, let $a_h = \exp(A_{\log,h})$ denote the learned positive decay magnitude, and let $\bar{\Delta}$ be the mean softplus time-step bias. For a sequence of length M, we estimate the head memory score by

$$m_h = \exp(-\bar{\Delta} a_h M). \tag{15}$$

The corresponding channel prior is

$$p_h = \sigma(\beta(m_h - \theta)), \tag{16}$$

where $\beta = 8.0$ and $\theta = 0.5$ in our implementation. The head-level prior is repeated over the channels belonging to the same head to obtain p_c. This prior favors channels whose learned decay suggests stronger long-range information preservation.

Geometry-Aware Channel Routing. The descriptor g_i is passed to a lightweight two-layer MLP with hidden dimension 64, producing the token-level geometry score α_i. The final routing weight is

$$r_{i,c} = \alpha_i p_c. \tag{17}$$

The content feature x is softly decomposed as

$$X^{\text{global}}_{i,c} = r_{i,c} x_{i,c}, \quad X^{\text{local}}_{i,c} = (1 - r_{i,c}) x_{i,c}. \tag{18}$$

This soft routing avoids a hard irreversible split and keeps local features available for all tokens.

Geo-Score Token Filtering. Token filtering is applied to the global branch. The temporal score is computed from the projected Mamba time-step vector:

$$s^t_i = \|\Delta_i\|_2. \tag{19}$$

The parameter-free geometric score is

$$s^g_i = \frac{1}{4} \left[\text{Norm}(\kappa_i) + \text{Norm}(sct_i) + \text{Norm}(dist_i) + \text{Norm}(\delta_{geo,i}) \right], \tag{20}$$

where $dist_i$ is the distance to the object center, $\delta_{geo,i}$ is the descriptor difference between adjacent serialized tokens, and each nonnegative cue is divided by its within-sequence maximum with an epsilon floor. This deterministic score adapts DeciMamba's training-free Δ-based decimation to

serialized point clouds without introducing a scoring MLP behind a non-differentiable index operation. The geometry-score weight is $\lambda_g = 0.5$. We compute a bounded inverse-density coverage term $s_i^d = \mathrm{Norm}((\rho_i + \epsilon)^{-1})$ and combine the three cues as

$$S_i = s_i^t(1 + \lambda_g s_i^g)(1 + \lambda_d s_i^d), \qquad \lambda_d = 0.1. \tag{21}$$

We select the largest $K = \lceil rM \rceil$ scores with hard Top-K. As in DeciMamba, this selection rule is training-free: it has no learnable parameters and needs no gradient estimator or auxiliary loss. The separate routing MLP remains trainable because $r_{i,c}$ continuously multiplies the local and global feature branches before selection. The selected indices are sorted back into sequence order before Mamba scanning.

Block Execution. For completeness, the PointLEX-Mamba block follows these steps:

1. Serialize point tokens, project the normalized sequence, apply the two local convolutions to the content/BC branch, and form positive Δ by softplus.
2. Compute g_i from token centers and obtain the routing score α_i.
3. Compute the channel memory prior p_c and the routing weight $r_{i,c}$.
4. Split the content feature into X^{local} and X^{global}.
5. Compute the parameter-free Geo-Score values and use hard Top-K to select the global token subset Ω.
6. Gather $X_\Omega^{\mathrm{global}}$, B_Ω, C_Ω, and Δ_Ω, and perform Mamba scanning only on the selected subsequence.
7. Scatter the scanned global output back to the original token positions and add it to the dense local branch.
8. Apply gated normalization and the output projection, keeping the original block interface unchanged.

C High-Density Test Set Construction

ModelNet40. ModelNet40 (Wu et al., 2015) contains clean synthetic CAD objects from 40 categories. For length extrapolation, we keep the official train/test split fixed. High-density test sets are generated from the original test shapes according to the official id-to-file mapping. Therefore, every extrapolated ModelNet40 test set contains only official test objects; the point number changes, but the object identity and split membership remain unchanged.

ScanObjectNN. ScanObjectNN (Uy et al., 2019) contains real scanned objects and is more sensitive to object-level leakage because the official processed h5 files do not provide a direct raw-object identifier. If high-density test sets are generated by directly sampling from the whole raw-data pool, a raw object corresponding to an official training sample may be incorrectly placed into the extrapolation test set. We therefore use a two-stage construction procedure to first match official processed h5 test samples to raw objects and then generate high-density test sets from the matched test-split objects.

First, we match each official processed h5 test sample to one raw object with the same class label. The raw candidates are indexed from ScanObjectNN raw `.bin` files, where the first three attributes are used as xyz coordinates. For OBJ_ONLY, an optional foreground index file is applied before matching when available. Each processed h5 sample and each raw candidate are normalized to the unit sphere and represented by a compact geometric fingerprint, including radial histograms, coordinate histograms, bounding-box extent, and radius quantiles. For each h5 test sample, the script retrieves the closest fingerprint candidates within the same class and then selects the best raw object by one-way Chamfer distance. By default, one raw object is used at most once, and the output manifest records the raw path, official h5 split, label, h5 index, fingerprint distance, Chamfer distance, and raw point count.

Second, the reported generator samples matched test objects without replacement and excludes any object with fewer points than the target. A read-only inspection of the currently accessible matched

files found 2756 PB-T50-RS objects, 576 OBJ_BG objects, and 62 OBJ_ONLY objects at every density from 1024 to 32768. Each split's manifest has the same number of unique raw paths, all marked test, and minimum raw point counts of 32778, 32796, and 33421, respectively. The h5 label vector at every density matches its manifest's label order. This supports a fixed eligible cohort in those files. Together, the unique raw paths, official test markers, fixed cohort size, and label-order agreement provide a consistent object-level construction across all evaluated densities.

Evaluation Lengths. All models are trained at 1024 input points and evaluated at 1024, 2048, 4096, 8192, 16384, and 32768 raw points. The corresponding group counts are 64, 128, 256, 512, 1024, and 2048; two serialization orders produce 128, 256, 512, 1024, 2048, and 4096 state-space tokens. With $r = 0.5$, PointLEX scans 64, 128, 256, 512, 1024, and 2048 global tokens. The same checkpoint is used at every test length.

Sparse-state semantics. Let the sorted retained indices be $i_1 < \cdots < i_K$. The sparse recurrence is

$$h_r = \bar{A}_{i_r} h_{r-1} + \bar{B}_{i_r} X^{\text{global}}_{i_r}, \qquad r = 1, \ldots, K. \tag{22}$$

Here, r indexes recurrence steps in the compact subsequence, whereas i_r denotes the corresponding position in the original length-M sequence. Sorting the retained indices preserves the serialized spatial order. Unselected positions introduce no global state transition, so the global recurrence has exactly K steps rather than M. At each retained position, the global output is computed as $Y^{\text{global}}_{i_r} = C_{i_r} h_r$ and then scattered back to position i_r before branch fusion.

The retained Δ, B, and C values therefore define a compact, ordered global subsequence. In parallel, the local convolution is computed before selection and preserves context at every original token position, including positions omitted from the global recurrence. The two branches combine efficient long-range propagation over selected tokens with dense local coverage over the complete sequence.

Implementation audit. The revised classifier explicitly activates the PointLEX block when requested, scales the group count with the raw point count, and logs raw points, model-input points, serialized tokens, selected tokens, and sample count. The FPS-to-1K path applies iterative farthest point sampling in Euclidean XYZ space to obtain exactly 1024 points before grouping and serialization; it therefore uses $G = 64$ and remains at 128 tokens for every raw input density. Forward and backward audits pass for both one-block and 12-block classifiers, confirming execution of the PointLEX path and gradient connectivity through the trainable routing branch.

D Implementation Details

The implementation uses one checkpoint for the complete length sweep. Training always uses 1024 raw points, 64 groups, and 128 serialized tokens. At evaluation, only the input-dependent group count, serialized length, and resulting Top-K budget change with density; all learned weights, normalization layers, and classification parameters remain fixed. No test-time optimization or density-specific fine-tuning is applied. Table 5 lists the common training configuration and the PointLEX-specific settings used by this protocol.

Length-dependent quantities are derived deterministically rather than tuned for each density: $G = N/16$, $M = 2G$ for the two serialization orders, and $K = \lceil rM \rceil$. Consequently, the global scan budget grows from 64 to 2048 tokens across the reported sweep instead of imposing a fixed-token bottleneck. Reporting N, G, M, and K together separates raw-point density from the effective SSM length and makes the protocol auditable.

Table 6: Transfer OA (%). Ours replaces each backbone's Mamba block with PointLEX-Mamba.

Method	1K	2K	4K	8K	16K	32K
ModelNet40						
PCM	91.53	90.40	85.66	68.68	45.26	26.26
PCM (Ours)	91.53	90.84	86.91	75.32	56.73	43.80
Mamba3D	91.17	86.06	76.46	51.62	23.54	10.49
Mamba3D (Ours)	91.17	88.90	82.29	59.40	32.74	16.00
ScanObjectNN PB-T50-RS						
PCM	91.04	90.31	81.68	66.62	52.29	39.84
PCM (Ours)	91.04	90.46	83.02	69.56	55.81	42.83
Mamba3D	90.97	90.86	77.32	58.09	38.90	27.32
Mamba3D (Ours)	90.97	88.53	88.64	75.29	53.66	36.36

Table 5: Implementation settings and PointLEX-Mamba-specific hyperparameters.

Item	Value
General Settings	
Backbone	PointMamba-style serialized Mamba backbone
Token dimension	384
Number of Mamba blocks	12
Number of groups	$N/16$: 64 at training, 64–2048 at evaluation
Group size	32
Classification head	Average pooling head
Optimizer	AdamW
Weight decay	0.05
Training epochs	300
Warm-up epochs	10
Fine-tuning learning rate	3×10^{-4}
ShapeNet pre-training learning rate	10^{-3}
Fine-tuning batch size	32
Pre-training batch size	128
Gradient clipping	10
Hardware	NVIDIA GeForce RTX 4090 D GPUs
PointLEX-Mamba Settings	
Geometry neighbors k	8
Routing MLP hidden dimension	64
Routing slope β	8.0
Routing threshold θ	0.5
Geometry-score weight λ_g	0.5
Coverage weight λ_d	0.1
Global scan budget K	$\lceil rM \rceil$, default $r = 0.5$
Token selector training	None (parameter-free hard Top-K)
Mamba local-awareness kernel size	7
Drop path rate	0.1
Dropout	0.0

Metrics. We report overall accuracy (OA, %) for classification. To measure extrapolation degradation, we compute the relative decay rates from the 1024-point result to the 8192-point and 32768-point results:

$$\text{Decay}_{1024\to L} = \frac{OA_{1024} - OA_L}{OA_{1024}} \times 100\%, \quad L \in \{8192, 32768\}. \tag{23}$$

E Additional Experimental Results

Backbone transfer. Supplementary Table 6 applies PointLEX to PCM (Zhang et al., 2025) and Mamba3D (Han et al., 2024). On ModelNet40, PointLEX raises 32K OA from 26.26% to 43.80%

for PCM and from 10.49% to 16.00% for Mamba3D. On PB-T50-RS, PCM improves at every extrapolated length, while Mamba3D gains strongly from 4K onward, including 27.32% to 36.36% at 32K. These results show that Geometry-Aware Channel Routing and Geo-Score Token Filtering transfer consistently across point-cloud state-space backbones.

Repeated Runs and FPS Protocol. The main table increases the actual state-space length from 128 to 4096 tokens and reports three-run means. The FPS-to-1K control applies farthest point sampling before grouping and presents 1024 points (128 tokens) to the frozen dense baseline, whereas PointMamba receives every token and PointLEX retains half of the growing sequence. This separates fixed-length downsampling from genuine growth of the state-space sequence.

Retention-Ratio Ablation. The default retention ratio $r = 0.5$ makes the selected global length grow from 64 to 2048 tokens across the density sweep. This preserves genuine short-to-long extrapolation while halving the global recurrence relative to the full serialized sequence.

Protocol and Implementation Audits. The PB-T50-RS audit verifies the same fixed cohort at every density. The implementation checks distinguish the parameter-free selector from the continuously trained routing branch.

Table 7: Completed protocol and implementation checks.

Check	Result
PB-T50-RS identities at all six lengths	2756, identical order
Duplicate manifest identities	0
Token-selector trainable parameters	0
Routing branch gradient path	continuous, task-supervised
12-block PointLEX forward/backward	pass

End-to-End Efficiency Protocol. The end-to-end protocol covers grouping, neighborhood search, eigendecomposition, scoring, selection, and Mamba scanning at 8K and 32K. It uses batch size one on a single RTX 4090 D, 20 warm-up iterations, and 100 timed iterations, with CUDA synchronization around each interval and peak memory reset before evaluation. This setup measures latency and memory over the complete PointLEX pipeline.

Additional ScanObjectNN Results. The main paper reports the main comparison on ModelNet40 and ScanObjectNN PB-T50-RS. To provide a more complete view of ScanObjectNN, Tables 8 and 9 report results on OBJ_ONLY and OBJ_BG, respectively. The compared methods follow the main paper, including PointMamba (Liang et al., 2024), LongMamba (Ye et al., 2025b), DeciMamba (Ben-Kish et al., 2025), MambaExtend (Azizi et al., 2025), Mamba Modulation (Lu et al., 2025), LAMB (Ye et al., 2025a), LongSSM (Wang, 2024), and ReMamba (Yuan et al., 2025). The supplementary tables include the methods with results available for each split. MambaExt. and Mamba Mod. denote MambaExtend and Mamba Modulation.

Table 8: Length extrapolation OA (%) on ScanObjectNN OBJ_ONLY.

Method	1K	2K	4K	8K	16K	32K
PointMamba	87.26	85.89	81.24	68.85	49.91	24.27
LongMamba	87.26	84.76	80.59	65.52	48.44	23.33
DeciMamba	87.26	84.85	80.38	67.99	47.33	22.20
MambaExt.	87.26	85.89	81.07	69.02	50.60	25.65
Mamba Mod.	87.26	85.20	81.07	68.67	48.36	23.24
LAMB	87.26	85.89	81.24	68.85	49.91	24.27
LongSSM	87.26	85.89	81.24	68.85	49.91	24.27
ReMamba	87.26	86.40	80.90	69.54	54.91	29.78
PointLEX	87.26	**89.50**	**84.62**	**71.29**	**58.90**	**39.07**

Table 9: Length extrapolation OA (%) on ScanObjectNN OBJ_BG.

Method	1K	2K	4K	8K	16K	32K
PointMamba	89.33	88.12	86.75	76.42	57.66	30.64
LongMamba	89.33	86.23	85.01	73.36	54.97	30.12
DeciMamba	89.33	88.12	85.71	75.56	54.56	30.64
MambaExt.	89.33	88.12	87.78	76.42	55.94	30.81
Mamba Mod.	89.33	88.30	87.44	76.25	57.83	28.92
LAMB	89.33	88.12	86.75	76.42	57.66	30.64
LongSSM	89.33	88.12	86.75	76.42	57.66	30.64
ReMamba	89.33	**91.22**	87.26	76.76	**63.86**	**40.96**
PointLEX	89.33	90.31	**88.37**	**79.48**	62.25	38.02

Discussion. The density-wise OA tables complement the PB-T50-RS results and show consistent PointLEX robustness across the OBJ_ONLY and OBJ_BG variants as input length increases.