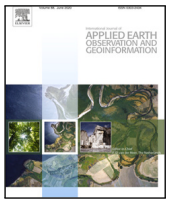
International Journal of Applied Earth Observation and Geoinformation 136 (2025) 104401

Contents lists available at ScienceDirect

International Journal of Applied Earth Observation and Geoinformation

journal homepage: www.elsevier.com/locate/jag



3D-UMamba: 3D U-Net with state space model for semantic segmentation of multi-source LiDAR point clouds

Dening Lu [a], Linlin Xu [a], Jun Zhou [b], Kyle Gao [a], Zheng Gong [c], Dedong Zhang [a,*]

[a] Department of Systems Design Engineering, University of Waterloo, Waterloo, Ontario, N2L 3G1, Canada
[b] School of Nursing, The Hong Kong Polytechnic University, TU428, Hong Kong, China
[c] School of Computer Engineering, Jimei University, Xiamen, 361021, China

ARTICLE INFO

Dataset link: https://github.com/d62lu/3D-UMamba

Keywords:
State space model
Mamba
Multi-source LiDAR data processing
Point cloud semantic segmentation

ABSTRACT

Segmentation of point clouds is foundational to numerous remote sensing applications. Recently, the development of Transformers has further improved segmentation techniques thanks to their great long-range context modeling capability. However, Transformers have quadratic complexity in inference time and memory, which both limits the input size and poses a strict hardware requirement. This paper presents a novel 3D-UMamba network with linear complexity, which is the earliest to introduce the Selective State Space Model (i.e., Mamba) to multi-source LiDAR point cloud processing. 3D-UMamba integrates Mamba into the classic U-Net architecture, presenting outstanding global context modeling with high efficiency and achieving an effective combination of local and global information. In addition, we propose a simple yet efficient 3D-token serialization approach (Voxel-based Token Serialization, i.e., VTS) for Mamba, where the Bi-Scanning strategy enables the model to collect features from all input points in different directions effectively. The performance of 3D-UMamba on three challenging LiDAR point cloud datasets (airborne MultiSpectral LiDAR (MS-LiDAR), aerial DALES, and vehicle-mounted Toronto-3D) demonstrated its superiority in multi-source LiDAR point cloud semantic segmentation, as well as the strong adaptability of Mamba to different types of LiDAR data, exceeding current state-of-the-art models. Ablation studies demonstrated the higher efficiency and lower memory costs of 3D-UMamba than its Transformer-based counterparts.

1. Introduction

LiDAR point cloud scene segmentation plays a crucial role in remote sensing due to its capability to provide high-resolution and accurate three-dimensional data of the Earth's surface. This technology allows for precise mapping and analysis of terrain, vegetation, and urban structures, facilitating various applications such as environmental monitoring (Pasternak et al., 2023), forestry management (Xiang et al., 2024), urban planning (Xiao et al., 2023; Stilla and Xu, 2023), and disaster response (Nikoohemat et al., 2020). By extracting valuable spatial information, LiDAR point cloud scene segmentation significantly enhances the analysis and interpretation of complex geographic data, leading to more informed decision-making and resource management.

The 3D Transformer technique has achieved great success in LiDAR point cloud processing (Cheng et al., 2023; Zhang et al., 2023a; Lu et al., 2024a). The self-attention mechanism plays a crucial role in capturing long-range dependency and contextual information, which allows the model to capture complex spatial dependencies within point cloud data. Specifically, given an input point cloud, the self-attention algorithm enables each point to establish a connection with every other point in the set by calculating a similarity matrix. By aggregating information from all points, the self-attention mechanism captures the global context and intricate relationships within the point cloud, leading to more comprehensive and accurate feature representations.

However, since the self-attention mechanism inherently involves calculating attention scores between every pair of points in the set, it results in quadratic complexity denoted as $\mathcal{O}(N^2D)$. Consequently, as N increases, the computational cost and memory requirements rise rapidly, which imposes strong hardware requirements. Therefore, designing an elegant model with linear complexity is necessary and meaningful while maintaining an excellent performance comparable to the Transformer.

Mamba, an improved Structured State Space Model (SSM), recently emerged as a highly promising method with great performance in long-range context dependency modeling (Gu and Dao, 2023). Moreover,

* Corresponding author.
E-mail addresses: d62lu@uwaterloo.ca (D. Lu), l44xu@uwaterloo.ca (L. Xu), zachary-jun.zhou@connect.polyu.hk (J. Zhou), y56gao@uwaterloo.ca (K. Gao), zheng.gong@jmu.edu.cn (Z. Gong), dedong.zhang@uwaterloo.ca (D. Zhang).

https://doi.org/10.1016/j.jag.2025.104401
Received 13 September 2024; Received in revised form 13 December 2024; Accepted 30 January 2025
Available online 13 February 2025

it achieves linear complexity due to its inherent recurrence relations. Compared to NLP, point cloud data presents more challenging characteristics, such as sparsity, uneven spatial distribution, and disorder. Therefore, it is essential to discuss Mamba's adaptability to point cloud processing before its application in this domain. First, the sparsity and uneven spatial distribution of point clouds pose significant challenges for local and global feature extraction. The selective state space model in Mamba addresses this by dynamically adjusting the system's state-space parameters based on the input sequence. This dynamic mechanism enables the model to analyze each input point and decide whether to propagate or discard past information. Such flexibility allows Mamba to effectively model long-range dependencies, capturing global contextual information even in sparse and unevenly distributed point clouds, resulting in robust feature representation capabilities. Second, the unordered nature of point clouds prevents them from being directly processed by Mamba. Therefore, serialization of point clouds is essential. Point cloud serialization typically leverages their geometric spatial attributes for ordering, ensuring that the original spatial topology and geometric relationships are preserved during serialization. Third, Mamba's selective state space model effectively incorporates contextual information while filtering irrelevant or redundant data. This capability is particularly beneficial for handling noise and outliers in point clouds, such as LiDAR scans, making Mamba suitable for noisy data processing. Finally, compared to Transformers, Mamba adopts a recurrence-based computation approach. The recurrence relation updates the state using only the previous state and the current input, resulting in linear complexity. This allows Mamba to achieve comparable global context modeling capability to Transformers while being significantly more efficient, making it well-suited for large-scale LiDAR point cloud processing.

Therefore, in this work, we introduced the Mamba technique to the field of remote sensing and explored its effectiveness in different types of LiDAR data processing. This is the earliest work to apply Mamba to multi-source LiDAR point cloud processing to our knowledge. Specifically, we integrate the Mamba block into the classic U-Net framework (Ronneberger et al., 2015), forming a novel network named 3D-UMamba. It has a hierarchical encoder–decoder structure and integrates both local and global feature learning effectively. The local feature learning is achieved by the common feature grouping and pooling operations, while the global feature learning is achieved by the Mamba blocks operated on progressively downsampled point sets. To transfer the unordered point clouds into 1D-sequence data, we proposed a simple yet effective serialization approach, named Voxel-based Token Serialization (VTS). The proposed 3D-UMamba achieved SOTA segmentation results across various LiDAR datasets and demonstrated superior algorithm efficiency compared to its Transformer-based counterpart.

We highlight the key contributions of the paper as follows:

- We proposed a novel Mamba-based LiDAR point cloud semantic segmentation framework, named 3D-UMamba, which achieves global context information capturing with linear complexity. The local information aggregation is integrated with global context modeling in 3D-UMamba for strong point cloud feature representation.
- We designed an effective token serialization approach, named Voxel-based Token Serialization (VTS), where the Bi-Scanning strategy allows each input point to effectively gather information from all other points in different directions. It enhances the model's generalization to sequential data by introducing random ordering of tokens within each voxel, which is especially suitable for LiDAR point cloud processing.
- 3D-UMamba achieves SOTA segmentation performance on multi-source LiDAR point cloud datasets (MS-LiDAR with 84.5% of mIoU, aerial DALES with 78.1% of mIoU, and vehicle-mounted Toronto-3D with 79.4% of mIoU), surpassing other recent Mamba-based methods such as PointMamba and PointCloud-Mamba.

2. Related work

2.1. LiDAR point cloud segmentation

Existing LiDAR point cloud segmentation methods can be broadly divided into four categories: Volume-based methods, Projection-based methods, Point-based methods, and Transformer-based methods.

Inspired by image processing, VoxNet (Maturana and Scherer, 2015) proposed the 3D voxelization method for point cloud processing. It used the regular volumetric grid to represent unstructured point clouds. After that, 3D convolutions can be directly performed for feature extraction. OctNet (Riegler et al., 2017) introduced an unbalanced grid-octree algorithm, enabling the efficient representation of higher-resolution input data ($256 \times 256 \times 256$) compared to VoxNet. Subsequently, advancements such as 4D Spatio-temporal ConvNets (Choy et al., 2019) adopted sparse convolution algorithms, which significantly improved efficiency by avoiding computations in unoccupied voxel regions. These techniques effectively mitigated the computational and memory overheads associated with voxelization.

Projection-based methods transform 3D point clouds into 2D representations, allowing the use of well-established convolutional neural network (CNN) architectures. These methods, such as SqueezeSeg (Wu et al., 2018) and its improved versions (SqueezeSegV2 (Wu et al., 2019b) and SqueezeSegV3 (Xu et al., 2020)), employ spherical or range projections to map point clouds into grids for semantic segmentation. For instance, RangeNet++ (Milioto et al., 2019) converts point clouds into range images, applies 2D fully convolutional networks for segmentation, and reconstructs the 3D structure post-segmentation. Similarly, PolarNet (Zhang et al., 2020) and Multi-Projection Fusion (MPF) (Alnaggar et al., 2021) utilize alternative projection schemes like polar or multi-view projections to enhance segmentation accuracy. While projection-based methods are computationally efficient and offer faster inference times, they face challenges such as discretization errors and difficulties in segmenting complex scenarios.

Point-based methods take the raw point cloud data with/without normals as input, dealing with unordered point clouds directly. PointNet (Qi et al., 2017a) first used shared Multi-Layer Perceptrons (MLPs) to achieve point cloud feature learning. After that, PointNet++ (Qi et al., 2017b) introduced the local feature aggregation into PointNet, making the network aware of the local information. Inspired by the strong local feature extraction capability of CNNs, some further variants, such as PointCNN (Li et al., 2018), PointConv (Wu et al., 2019a), and DGCNN (Wang et al., 2019b), proposed to design the 3D convolutional kernels and Graph Convolution algorithm to improve point cloud processing and analysis. PointNext (Qian et al., 2022) further explored the potential of point-based methods on point cloud segmentation, and fully investigated various data augmentation approaches for performance improvement. Unlike earlier methods such as PointNet++, PointMLP (Ma et al., 2022) employs a fully-connected MLP-based architecture with a lightweight design that efficiently extracts both local and global features through hierarchical grouping and feature aggregation. By optimizing feature learning and leveraging residual connections, PointMLP achieves excellent results in point cloud segmentation.

In recent years, inspired by the application of Transformers in the image processing field, many Transformer-based point cloud segmentation methods have been proposed. The unique mechanism of Transformer, self-attention, achieves excellent results thanks to its strong capability of global context modeling. Point Cloud Transformer (PCT) (Guo et al., 2021) replaced the shared MLPs in PointNet architecture with Transformer blocks, making the network aware of long-range context dependencies. Point Transformer (PT) (Zhao et al., 2021a) applied the Transformer to local feature extraction, which has shown its superiority in feature aggregation. Instead of using raw point clouds, Stratified Transformer (Lai et al., 2022) took 3D voxels as input to the segmentation network. It applied Transformers in predefined local windows, following Swin Transformer (Liu et al., 2021). Furthermore,

there also are many efficient point cloud Transformers (Hui et al., 2021; Zhang et al., 2022; Park et al., 2022; Sun et al., 2023; Robert et al., 2023; Wang et al., 2023; Liu et al., 2023) proposed to reduce computation and memory costs and improve processing efficiency. PPT-Net (Hui et al., 2021) proposed a hierarchical encoder–decoder network to reduce the number of points gradually. Instead of using a pure Transformer architecture, it combined graph convolution-based (Wang et al., 2019b) local feature embedding and Transformer-based global feature learning, which not only enhances long-term dependencies among points but also reduces the computational cost. PatchFromer (Zhang et al., 2022), SPFormer (Sun et al., 2023), and SPT (Robert et al., 2023) integrated superpoint-based local feature aggregation methods with Transformers, which are able to generate geometrically-homogeneous point clusters for further local feature extraction, and only need to be calculated once, as a pre-processing step. Since the number of superpoints is much less than that of raw input points, superpoint-based Transformers achieved great process in model efficiency improvement.

2.2. State Space Models (SSMs)

Structured State Space Sequence models (S4), recently proposed by Gu et al. (2022b) have achieved great success in NLP (Özçelik et al., 2024; Shi and Xiang, 2024; Gu et al., 2022a; Fu et al., 2023). These models transform the structured state matrices within SSMs to allow for stable diagonalization. This approach simplifies SSM computations to the calculation of a Cauchy kernel. They have shown strong performance in modeling long-range contexts. They also are highly efficient due to the advantage of linear complexity. On this basis, a series of S4 variants (Gu et al., 2022a, 2023; Gupta et al., 2022) were proposed for further efficient improvement. Mamba (Gu and Dao, 2023) was proposed to solve further the limitation of the inability to perform content-based reasoning in S4. Its system parameters are determined by the input, which enables the model to adaptively select information along the sequence length dimension depending on the current token.

Thanks to its excellent performance in NLP, Mamba has been expanded into the vision domain Wang et al. (2024a), Ma et al. (2024), Liu et al. (2024). VMamba (Liu et al., 2024) proposed a novel vision backbone with a 2D Selective Scan (SS2D) module. It addressed the problem of non-sequentiality of 2D vision data, facilitating the use of Mamba in vision perception tasks. U-Mamba (Ma et al., 2024) proposed a combined CNN-SSM block, combining both Convolutional Neural Networks (CNNs) and SSMs for local and global feature extraction. It achieved SOTA performance in diverse biomedical tasks.

2.3. 3D Mamba works

The great success of Mamba in the NLP and 2D vision domains attracted increasing attention to 3D Mamba development. PointMamba (Liang et al., 2024) introduced an innovative approach by utilizing Hilbert curves for point serialization and developed a streamlined, non-hierarchical 3D Mamba architecture tailored for point cloud processing. However, the lack of hierarchical structure in PointMamba limits its ability to effectively combine local and global features, leading to a relatively weak feature learning and representation ability in point cloud segmentation. PointCloudMamba (Zhang et al., 2024) integrated the geometry affine blocks proposed in PointMLP (Ma et al., 2022) and Mamba blocks for point cloud processing, with a novel Consistent Traverse Serialization technique. Its experiments on various tasks demonstrate the significant potential of the Mamba framework in handling point cloud data. PointTramba (Wang et al., 2024b) explored the combination of Mamba and Transformer, where the Transformer was applied for intra-group information gathering, and the Mamba was utilized for inter-group information learning. Its experiments demonstrated the effectiveness of the combination, facilitating a new research direction in 3D Mamba research.

To explore the effectiveness of Mamba for LiDAR point cloud processing, this paper proposes a new 3D Mamba backbone for LiDAR point cloud scene segmentation, named 3D-UMamba. Dense experiments on various types of LiDAR data demonstrated the SOTA performance of 3D-UMamba in terms of both accuracy and efficiency.

3. Methodology

This section introduces details of the 3D-UMamba network. Thanks to its linear complexity and excellent global modeling capability, Mamba has achieved great success in NLP. To investigate the capabilities of Mamba for LiDAR point cloud processing, we integrated Mamba blocks into the classical U-Net framework (Ronneberger et al., 2015). Firstly, we introduced the overall framework of 3D-UMamba, shown in Fig. 1. Secondly, we presented the novel token serialization strategy. Lastly, we introduced the key Mamba blocks in our 3D-UMamba network.

3.1. Overview

As shown in Fig. 1, 3D-UMamba is designed as a hierarchical encoder–decoder network, taking the raw LiDAR point set $\mathbb{P} = \{p_i\}_{i \in N} \in R^{N \times C}$ as input, where C is the feature dimension of input points.

Encoder Architecture. $\mathbb{P}$ is first fed into a sparse convolution layer named Point-Voxel CNN (PVCNN) (Liu et al., 2019b) for token embedding. The embedded tokens are denoted as $\mathbb{T} = \{t_i\}_{i \in N} \in R^{N \times D}$, where D is the feature dimension of the tokens. $\mathbb{T}$ is then fed into a series of cascading modules for hierarchical deep feature extraction.

Module 1, for example, contains a token grouping and aggregating (TGA) block, a voxel-based token serialization (VTS) block, and a vanilla Mamba block. In Fig. 2, the TGA block performs the Farthest Point Sampling (FPS) and multi-scale ball-query grouping method (Qi et al., 2017b) for point cloud sampling and clustering, followed by achieving local feature aggregation by an MLP layer and average pooling layer. We denote the aggregated tokens as $\mathbb{S} = \{s_i\}_{i \in S} \in R^{S \times D_1}$, where S is the number of downsampled tokens ($S = N/4$ in our experiments), and D_1 represents the feature dimension after aggregation.

Since Mamba requires the 1D-sequence tokens as input, the VTS block is designed to serialize $\mathbb{S}$ by a novel Bidirectional Scanning strategy (i.e., Bi-Scanning) based on their 3D coordinates. A more detailed introduction to the VTS block is provided in Section 3.2. The serialized token sets from different scanning directions are taken as input to the Mamba block independently for global information modeling.

Subsequently, all resultant series are reordered and merged as the output of Mamba blocks, denoted as $\mathbb{M} = \{m_i\}_{i \in S} \in R^{S \times D_1}$. The Bi-Scanning strategy allows each token to have bidirectional perception of all other tokens, which leads to effective global information modeling. A more detailed introduction to the Mamba block is provided in Section 3.3.

Finally, we combine the locally aggregated features $\mathbb{S}$ from the TGA block and $\mathbb{M}$ from the Mamba block by a residual connection as the final output of Module 1, which can be expressed as:

$$\mathbb{F} = \mathbb{M} + MLP(\mathbb{S}), \tag{1}$$

where $MLP(\mathbb{S})$ represents an MLP layer operated on $\mathbb{S}$, + means the element-wise addition. This combination enables 3D-UMamba to consider both local and global information of the target point cloud, thus achieving a more comprehensive semantic information analysis.

Decoder Architecture. Shown in Fig. 1, the decoder contains the same number of modules as the encoder. Moreover, each module includes a point cloud upsampling block achieved by the trilinear interpolation method and an MLP block for feature extraction. In addition, the residual connection is used between the connecting feature maps of the encoder to those of the decoder.

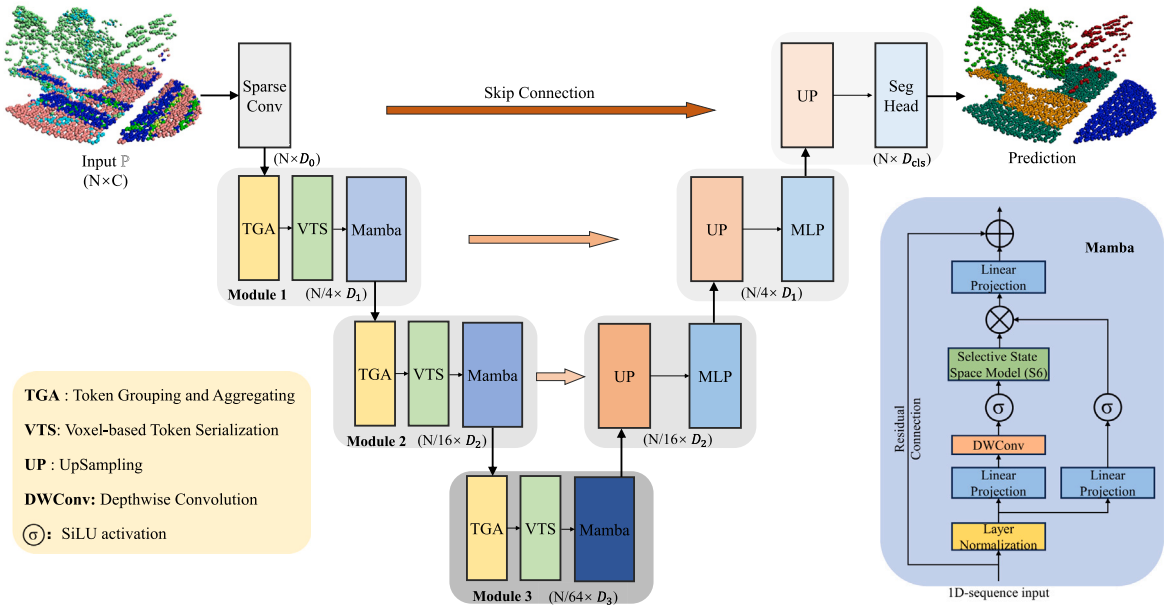


Fig. 1. The pipeline of our 3D-UMamba. It is a hierarchical encoder–decoder framework, that integrates Mamba into the U-Net architecture for both local and global information modeling. It includes three main components: Token Grouping and Aggregating (TGA), Voxel-based Token Serialization (VTS), and Mamba. Mamba is implemented by integrating the S6 model into a simple neural network architecture with linear projection and Depth-Wise Convolution (DWConv). Specifically, the dimensions of input features to each module are $D_0 = 64$, $D_1 = 320$, $D_2 = 640$, and $D_3 = 1280$.

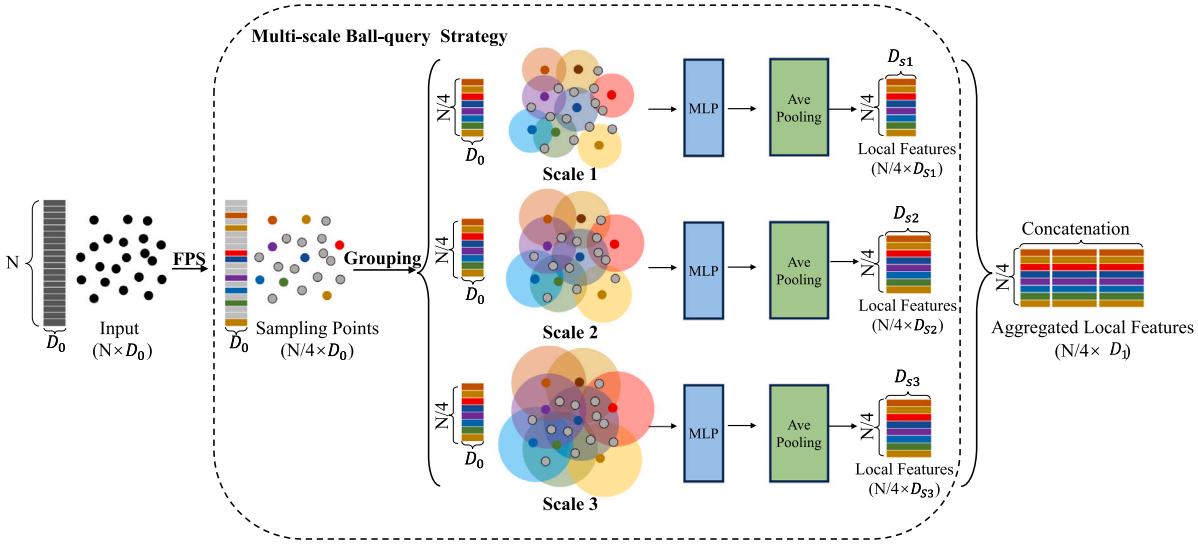


Fig. 2. Illustration of the TGA block. The Farthest Point Sampling (FPS) and multi-scale ball-query clustering methods are used for point cloud sampling and clustering, followed by local feature aggregation through MLP and average pooling layers. Specifically, in Module 1, $D_{s1} = 64$, $D_{s2} = 128$, $D_{s3} = 128$, and $D_1 = D_{s1} + D_{s2} + D_{s3} = 320$.

3.2. Voxel-based token serialization block

We propose a simple and efficient serialization method (VTS) to generate 1D-sequence tokens as input to the Mamba block. Visualized in Fig. 3, we first voxelize the output tokens $\mathbb{S}$ from the TGA block based on their 3D coordinates.

Secondly, inspired by the 2D-Selective-Scan (SS2D) (Liu et al., 2024), the voxels are traversed along two different scanning directions (forward and backward directions), to generate two sets of voxel sequences in different orders, which is called Bi-Scanning. Specifically, in the forward direction, voxels are ordered sequentially along the X-axis in ascending order, followed by the Y-axis in ascending order for voxels with identical X-coordinate values, and finally the Z-axis in ascending order for voxels with identical Y-coordinate values. Conversely, in the backward direction, voxels are ordered sequentially along the X-axis in descending order, followed by the Y-axis in descending order, and then the Z-axis in descending order. Due to the varying spatial contexts of each token across different directions, the Bi-Scanning strategy allows each token in $\mathbb{S}$ to effectively gather information from all other tokens in different directions. This multidirectional aggregation captures intricate spatial relationships, providing a richer global context crucial for point cloud representation. Unlike Transformers, which capture global dependencies through parallel attention mechanisms, Mamba relies on sequential processing that requires input data to follow a specific order. This dependency on ordered input tokens poses challenges for unordered point cloud data, as a single-directional scan struggles to effectively aggregate information between distant tokens in the sequence. Specifically, tokens at the beginning of the sequence may fail to integrate meaningful context information from those at the

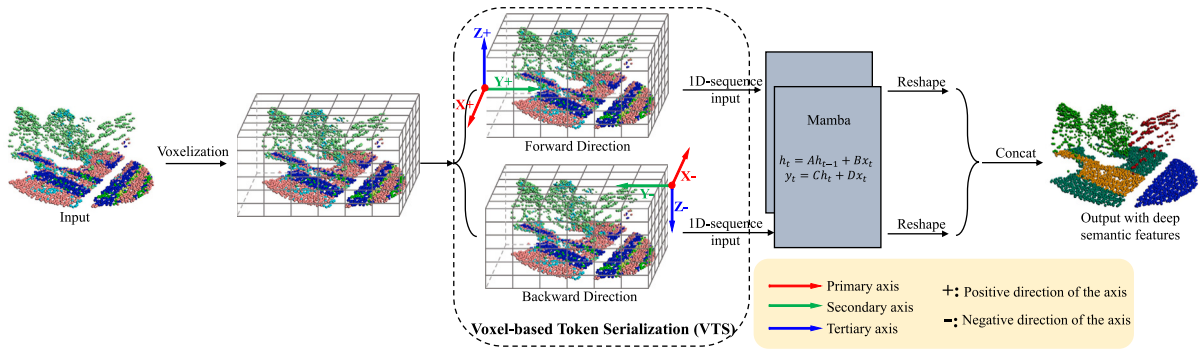


Fig. 3. Illustration of the VTS block. The VTS block serializes the 3D tokens into 1D-sequence tokens, as input to the Mamba block, where the novel Bi-Scanning strategy allows each token to effectively gather global information in different directions. For example, in the forward direction, voxels are ordered sequentially along the X–Y–Z axis in ascending order. + means the positive direction of the axis. To express it clearly, the X-, Y-, and Z-axis are designated as the primary, secondary, and tertiary axes, respectively.

end. By introducing Bi-scanning, Mamba processes the point cloud from two opposing directions, enabling more comprehensive and balanced information aggregation. This approach facilitates the establishment of global receptive fields, allowing Mamba to better capture long-range dependencies and model global context in unordered point clouds.

Lastly, the tokens in $\mathbb{S}$ are serialized according to their voxel indices, where tokens within the same voxel are randomly ordered to make the model more robust to new data. The VTS strategy effectively preserves the spatial structure of the point cloud during the serialization process. The Bi-Scanning process traverses the voxels sequentially along the X, Y, and Z axes in both forward and backward directions. This ensures that the spatial topological and geometric relationships of the original point cloud are preserved during serialization. Tokens that are spatially adjacent in the original 3D space remain close to each other in the serialized 1D sequence. Conversely, tokens that are spatially distant are also far apart in the serialized order. Furthermore, the random ordering of tokens within the same voxel ensures that the context in the serialized order still contains the corresponding spatial neighborhood information. Therefore, the global spatial continuity and topology of the original point cloud can be effectively preserved by the VTS method. In addition, the hierarchical design of 3D-UMamba incorporates TGA blocks, which aggregate features within local neighborhoods, further enhancing the spatial coherence of the serialized data.

3.3. Mamba block

As illustrated in Fig. 3, the Mamba block receives each set of serialized tokens from the VTS block as input. Mamba is built based on the S4 model (Gu et al., 2022b), which is described in detail as follows.

Structured State Space Sequence Model (S4). The S4 architecture was recently developed based on State Space Models (SSMs) (Gu et al., 2022b). Continuous-time SSMs are designed to map a sequence $x(t) \in \mathbb{R}$ to $y(t) \in \mathbb{R}$ by an implicit latent state $h(t) \in \mathbb{R}^N$. The SSM sequence-to-sequence transformation is defined as follows

$$\begin{aligned} h'(t) &= Ah(t) + Bx(t), \\ y(t) &= Ch(t). \end{aligned} \tag{2}$$

In this equation, $A \in \mathbb{R}^{N\times N}$, $B \in \mathbb{R}^{N\times 1}$, and $C \in \mathbb{R}^{1\times N}$. Furthermore, to integrate SSMs into deep learning models, the parameters (A, B) in continuous-time SSMs are discretized into $(\bar{A}, \bar{B})$ by the zero-order hold (ZOH) discretization rule as follows:

$$\begin{aligned} \bar{A} &= exp(\Delta A) \\ \bar{B} &= (\Delta A)^{-1}(exp(\Delta A) - I) \cdot \Delta B, \end{aligned} \tag{3}$$

where Δ is the time-scale parameter for discretization. Therefore, the new parameters $(\Delta, \bar{A}, \bar{B}, C, D)$ define the S4 model, achieving discretized sequence transformation from $x(t)$ to $y(t)$, which can be expressed as:

$$\begin{aligned} h_t &= \bar{A}h_{t-1} + \bar{B}x_t \\ y_t &= Ch_t + Dx_t, \end{aligned} \tag{4}$$

where $D \in \mathbb{R}^{1\times N}$ represents a residual connection.

Selective State Space Models (S6). Since the parameters in SSMs are fixed across the entire sequence processing, which is also called the Linear Time-Invariant (LTI) property, the content-based reasoning capability of SSMs is limited. To tackle this limitation, Gu and Dao (2023) proposed a novel parameterization method for SSMs. It enables dynamic adjustment of parameters (Δ, B, C) according to the input data, allowing the model to dynamically transfer information across the sequence length axis. Therefore, the proposed novel time-varying model named Selective SSM (S6) has linear complexity and strong context-based reasoning ability. Due to the recurrence-based computation approach in S6, it requires the sequence data as input. This is why the VTS block is proposed to serialize the unordered point clouds into the 1D-sequence data as input to the Mamba block. Besides, considering the sparsity and uneven spatial distribution of point clouds, a Depth-Wise Convolution (DWConv) (Chollet, 2017) is introduced into the Mamba block for local feature enhancement, as shown in Fig. 1.

4. Experiments

This section begins with an overview of the 3D-UMamba implementation. Following this, we assessed the effectiveness of 3D-UMamba in segmentation tasks using three challenging LiDAR point cloud datasets: the Airborne MS-LiDAR dataset (Zhao et al., 2021b), DALES (Varney et al., 2020), and Toronto −3D (Tan et al., 2020). These datasets, which represent different types of point cloud data (airborne MS-LiDAR, aerial LiDAR, and vehicle-mounted LiDAR), provide a comprehensive assessment of 3D-UMamba's performance in LiDAR data processing. At last, we presented related ablation experiments on the main components in 3D-UMamba and analyzed the method's sensitivity to token serialization approaches.

4.1. Implementation details

3D-UMamba was developed using PyTorch, with the training procedure employing an initial learning rate of 0.01 and adjusting the learning rate through a cosine annealing schedule. The batch size was configured to 16 for the Airborne MS-LiDAR and Toronto-3D datasets, and 8 for the DALES dataset, to accommodate varying input sizes. A weighted Cross-Entropy (CE) loss function was utilized during the training phase. All training parameters were selected based on empirical results to enhance performance.

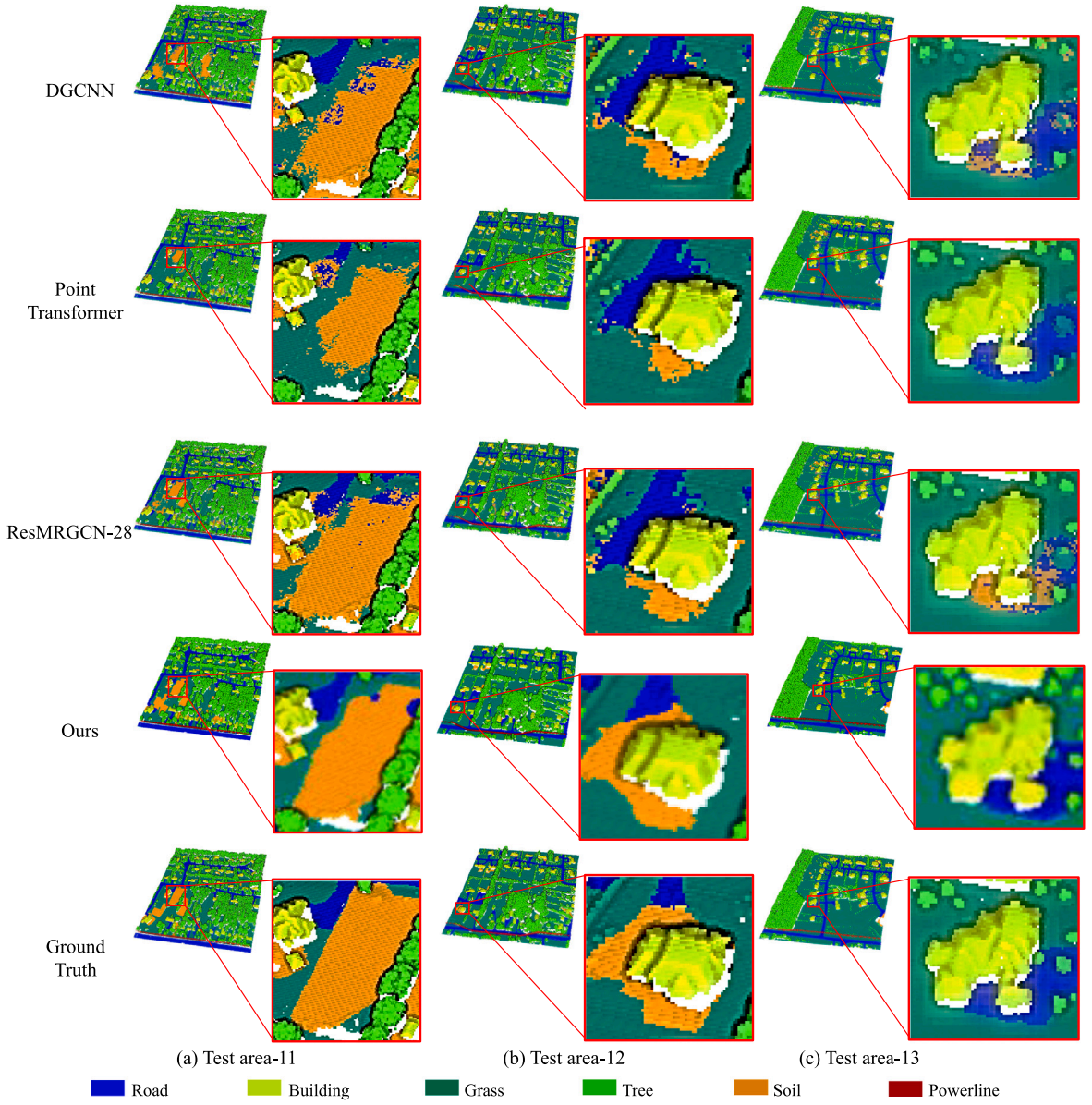


Fig. 4. Visualization result comparison of different methods on the testing areas 11–13 in the airborne MS-LiDAR dataset.

4.2. Airborne MS-LiDAR dataset

Dataset and Metrics. The airborne MS-LiDAR dataset, introduced by Zhao et al. (2021b), originates from the Teledyne Optech Titan LiDAR system. The data spans 13 areas located in a Canadian town, with Areas 1–10 designated for training and Areas 11–13 for testing. Specifically, each area spans over 15,000 m^2 and has an average point density of approximately 3.6 points/m^2. Each point includes six attributes: XYZ coordinates, MIR (1550 nm wavelength), NIR (1064 nm wavelength), and Green (532 nm wavelength).

The dataset includes over 10 million labeled points, distributed across six semantic categories, as shown in Table 2. It poses a challenge due to its imbalanced class distribution. As shown in Table 1, the quantity of road points exceeds the number of powerline points by more than 30 times. The MS-LiDAR dataset serves as a valuable resource for benchmarking deep learning algorithms in applications such as land cover classification, urban planning, and environmental monitoring. We adopted the data pre-processing method in Zhao et al. (2021b) for training/testing sample generation in our experiments, to ensure a fair comparison. Each area was divided into a group of local blocks using k-Nearest Neighbor (kNN) searching, with k set to 4096.

$$
\begin{aligned}
OA &= \frac{T}{N}, \\
mIoU &= \frac{\sum_{i=1}^{Cls} mIoU_i}{Cls}, \\
mIoU_i &= \frac{\sum TP_i}{\sum TP_i + \sum FP_i + \sum FN_i}, \\
AverageF_1 &= \frac{\sum_{i=1}^{Cls} F_{1i}}{Cls}, \\
F_{1i} &= \frac{Precision_i * Recall_i}{Precision_i + Recall_i}, \\
Precision_i &= \frac{\sum TP_i}{\sum TP_i + \sum FP_i}, \\
Recall_i &= \frac{\sum TP_i}{\sum TP_i + \sum FN_i}.
\end{aligned}
\tag{5}
$$

Performance evaluation metrics include average F_1 score, Mean Intersection over Union (mIoU), and Overall Accuracy (OA) to process the given input data. Additionally, we provided Precision, Recall, and

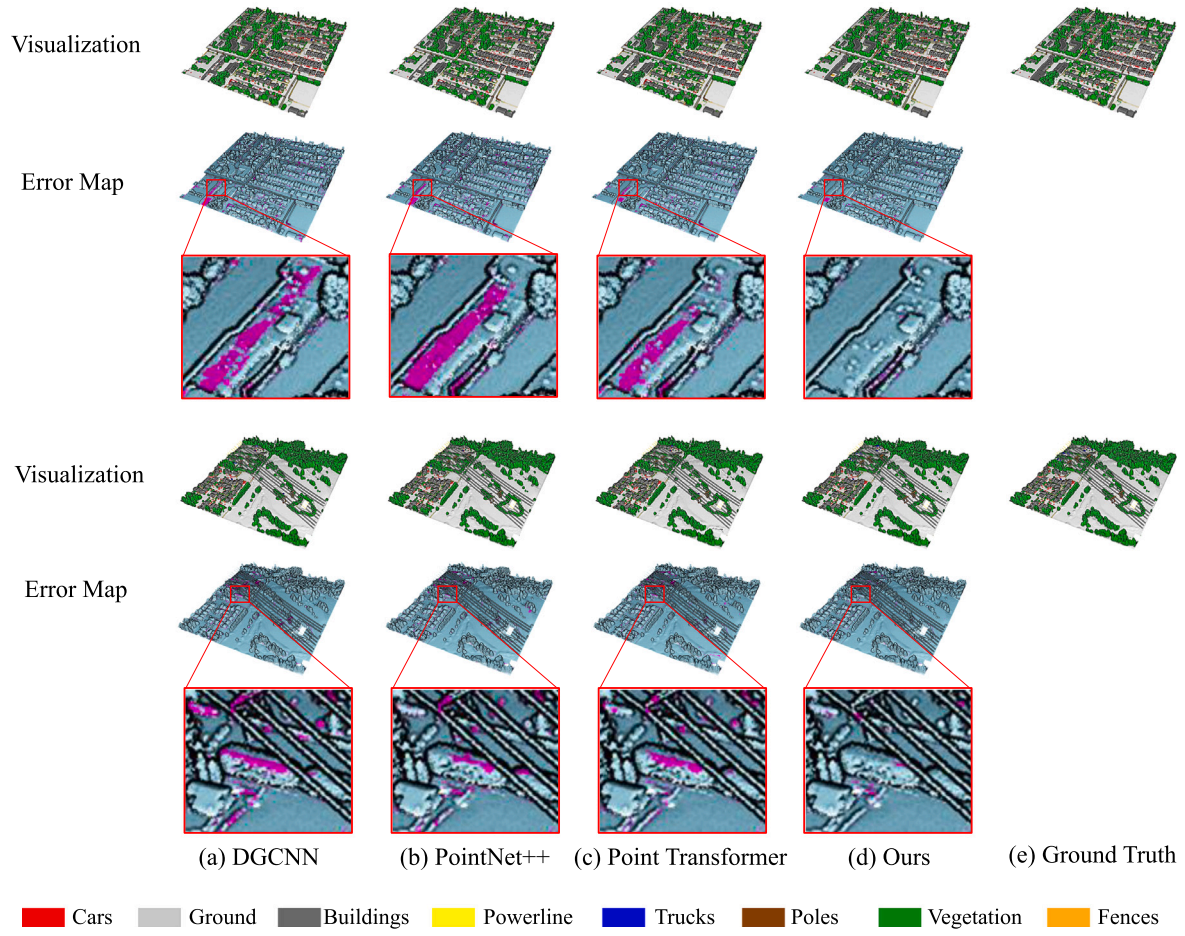


Fig. 5. Visualization result comparison of different methods on the DALES dataset. To compare model performance more clearly, we also show error maps of the segmentation results compared to the ground truth, where the misclassified points are colored in red. From the visualization results, our segmentation predictions are precise and closely match the ground truth. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

Table 1
Comparison of the point numbers in each category in the MS-LiDAR dataset.

Class	Training set (No. 1–10)	Test set (No. 11–13)
Road	586,987	258,942
Building	415,763	116,085
Grass	2,577,194	983,724
Tree	2,623,317	863,519
Soil	124,566	33,214
Powerline	12,155	7545

F_1 scores for each category. The definitions of metrics are shown in Eq. (5), where T is the number of all correctly predicted points, $T = \sum_{i=1}^{K} T_i$, T_i is the number of correctly predicted points in class i, N is the number of all points in the dataset, $N = \sum_{i=1}^{K} N_i$ and N_i is the number of points in class i, Cls is the number of categories in the dataset, and $\sum TP_i$ represents the sum of TP samples in class i.

Performance Evaluation. Table 2 shows a detailed evaluation results of 3D-UMamba on MS-LiDAR data. Specifically, it provides the confusion matrix of 3D-UMamba on the dataset, as well as the Precision, Recall, and F_1 score for each category. From the table, we can see that the F_1 scores of all categories, except for the soil, exceed 90%, demonstrating the strong segmentation performance of our method on the airborne MS-LiDAR data. Due to the similar geometric characteristics and unbalanced class distribution, over 25% soil points were misclassified as road and building points, leading to a relatively low F_1 score of 69.2% for the soil class.

Table 2
Confusion matrix (%) of 3D-UMamba on the airborne MS-LiDAR dataset.

Categories	Road	Building	Grass	Tree	Soil	Powerline
Road	97.2	5.2	0.1	0.0	11.6	0.0
Building	1.4	89.6	0.9	0.0	15.2	0.0
Grass	0.0	0.3	98.7	0.2	0.0	3.3
Tree	0.0	0.3	0.3	99.8	0.0	0.0
Soil	1.3	4.6	0.0	0.0	73.2	0.0
Powerline	0.0	0.0	0.0	0.0	0.0	96.7
Precision	83.9	95.5	99.8	95.6	65.6	95.4
Recall	97.2	89.6	98.7	99.8	73.2	96.7
F_1	90.1	92.5	99.2	97.7	69.2	96.0

The performance comparison result is demonstrated in Table 3. From the results, 3D-UMamba achieves the best segmentation performance in terms of all accuracy metrics. Fig. 4 shows the visualization of the segmentation result comparison, where zoom-in parts suggest that 3D-UMamba outperforms other benchmarked methods. Overall, both the performance evaluation and comparison demonstrate the superiority of 3D-UMamba in airborne MS-LiDAR point cloud semantic segmentation.

4.3. DALES

Dataset and Metrics. The DALES dataset is a comprehensive aerial LiDAR dataset designed for semantic segmentation, introduced by Varney et al. (2020). This dataset was collected using a Riegl Q1560

Table 3
Performance comparison (%) of different methods on the airborne MS-LiDAR dataset.

Methods	Input Points	Average F_1 score	mIoU	OA	Latency(ms)
Other Deep Learning Models					
PointNet++ (Qi et al., 2017b)	4096	72.1	58.6	90.1	322.6
KPConv (Thomas et al., 2019)	4096	76.8	64.4	92.2	83.7
DGCNN (Wang et al., 2019b)	4096	71.6	51.0	91.4	86.2
RSCNN (Liu et al., 2019a)	4096	73.9	56.1	91.0	158.7
GACNet (Wang et al., 2019a)	4096	67.7	51.0	90.0	277.8
AGConv (Zhou et al., 2021)	4096	76.9	71.2	93.3	312.5
SE-PointNet++ (Jing et al., 2021)	4096	75.9	60.2	91.2	–
FR-GCNet (Zhao et al., 2021b)	4096	78.6	65.8	93.6	–
ResMRGCN-28 (Li et al., 2023)	4096	81.1	74.0	93.3	45.7
GCNAS (Zhang et al., 2023b)	4096	88.1	82.3	95.2	–
Transformer-based Models					
PointTransformer (Zhao et al., 2021a)	4096	80.5	73.6	93.1	285.7
Xiao et al. (Xiao et al., 2022)	4096	83.3	79.3	94.0	–
PatchFormer (Zhang et al., 2022)	4096	82.4	77.8	93.1	62.9
3DGTN (Lu et al., 2024a)	4096	88.6	82.1	95.2	217.4
DCTNet (Lu et al., 2024b)	4096	86.0	80.2	95.0	**23.3**
Mamba-based Models					
PointMamba (Liang et al., 2024)	4096	86.3	77.6	94.2	47.4
PointCloudMamba (Zhang et al., 2024)	4096	86.1	80.5	94.8	52.7
Ours	4096	**90.8**	**84.5**	**95.9**	66.1

Table 4
Performance comparison (%) of different methods on the DALES dataset, including OA, mIoU, and latency(ms).

Methods	input points	OA	mIoU	Latency(ms)
PointNet++ (Qi et al., 2017b)	8192	95.7	68.3	487.1
KPConv (Thomas et al., 2019)	8192	96.9	72.4	125.3
DGCNN (Wang et al., 2019b)	8192	96.1	66.4	136.2
PointCNN (Li et al., 2018)	8192	97.2	58.4	–
SPG (Landrieu and Simonovsky, 2018)	8192	95.5	60.6	–
ConvPoint (Boulch, 2020)	8192	97.2	67.4	–
PointTransformer (Zhao et al., 2021a)	8192	97.1	74.9	468.4
SuperCluster (Robert et al., 2024)	8192	–	77.3	–
PReFormer (Akwensi et al., 2024)	8192	92.9	70.9	–
PointTransformerV3 (Wu et al., 2024)	8192	96.9	77.4	**41.9**
PointMamba (Liang et al., 2024)	8192	96.3	73.3	60.8
PointCloudMamba (Zhang et al., 2024)	8192	97.0	74.7	77.5
Ours	8192	**97.2**	**78.1**	79.2

dual-channel system mounted on a Piper PA31 Panther Navajo aircraft. The collection covered 330 km^2 over Surrey, British Columbia, Canada, with DALES focusing on a 10 km^2 subset comprising 40 unique tiles, each spanning 0.5 km^2. The dataset features an average resolution of 50 points per square meter and maintains high accuracy through rigorous calibration and noise filtering. These characteristics make DALES particularly suitable for applications like 3D urban modeling, environmental monitoring, and deep learning algorithm benchmarking.

Each point in the DALES dataset includes 4 attributes: X, Y, Z coordinates, and intensity. The dataset encompasses eight semantic classes — Buildings, Cars, Trucks, Poles, Power lines, Fences, Ground, and Vegetation — meticulously hand-labeled with the aid of satellite imagery and a digital elevation model. DALES represents one of the largest publicly available aerial LiDAR datasets, providing over 500 million annotated points. It is split into training and testing subsets (70% and 30%, respectively) to facilitate robust machine learning evaluations.

The dataset was subsampled by using a 10 cm grid (Robert et al., 2023) and further divided into groups of 20 m × 20 m blocks as training/testing samples, with each block containing 8192 points after sampling. The performance evaluation metrics used were mIoU and OA.

Performance Evaluation. Detailed performance comparisons on DALES are shown in Table 4. From the table, 3D-UMamba got the best mIoU (78.1%) and OA (97.2%) values, exceeding other SOTA methods such as PointTransformerV3 (Wu et al., 2024). The visualization results comparison is shown in Fig. 5. All segmentation predictions are precise and closely match the ground truth, demonstrating the superiority of 3D-UMamba in aerial LiDAR point cloud semantic segmentation.

4.4. Toronto-3D

Dataset and Metrics. Toronto-3D (Tan et al., 2020) is a high-resolution urban LiDAR dataset collected using a Velodyne HDL-32E laser scanner mounted on a vehicle. The dataset spans approximately 1 km of roadway in Toronto, Canada, and is divided into four segments (L001-L004), each covering around 250 meters of urban roadway. With a total of 78.3 million points, Toronto-3D offers a dense point cloud with an average point density of approximately 1000 points per square meter. This dataset was created to support urban scene understanding and semantic segmentation, featuring diverse urban elements such as roadways, sidewalks, and vehicles.

Each point in the dataset is labeled with eight semantic classes: road, road markings, natural (e.g., trees and shrubs), building, utility lines, poles, cars, and fences. The point attributes include XYZ coordinates, RGB, and intensity values, providing rich spatial and radiometric information for analysis. The dataset is divided into training (L001, L003, L004) and testing (L002) sets to facilitate benchmarking. Toronto-3D serves as a valuable resource for developing and evaluating algorithms in urban LiDAR data processing, including applications like autonomous driving and urban infrastructure analysis. We subsampled each area by using a 6 cm grid and applied a coordinate offset as described in Tan et al. (2020). Further, each area was split into groups of 5 m × 5 m blocks, each containing 4096 points after sampling. Each

Table 5
Performance comparisons (%) on the Toronto-3D dataset, including OA, mIoU, and the IoU value for each category.

Methods	OA	mIoU	IoU for Each Category							
			Road	Rd mrk.	Natural	Building	Util. line	Pole	Car	Fence
PointNet++ (Qi et al., 2017b)	92.6	59.5	92.9	0.0	86.1	82.2	61.0	62.8	76.4	14.4
KPConv (Thomas et al., 2019)	95.3	69.1	94.6	0.0	**96.0**	91.5	**87.6**	**81.5**	85.6	15.7
DGCNN (Wang et al., 2019b)	94.2	61.8	93.8	0.0	91.2	80.3	62.4	62.3	88.2	15.8
MS-PCNN (Ma et al., 2019)	90.0	65.9	93.8	3.8	93.4	82.5	67.8	71.9	91.1	22.5
TGNet (Li et al., 2019)	94.0	61.3	93.5	0.0	90.8	81.5	65.2	62.9	88.7	7.8
MS-TGNet (Tan et al., 2020)	95.7	70.5	94.4	17.1	95.7	88.8	76.0	73.9	**94.2**	23.6
PointTransformer (Zhao et al., 2021a)	**97.1**	69.7	93.9	**65.8**	88.8	77.8	69.8	62.7	86.3	12.3
Han et al., (Han et al., 2021)	93.6	70.8	92.2	53.8	92.8	86.0	72.2	72.5	75.7	21.2
GAANet (Wan et al., 2023)	92.7	65.1	92.7	39.3	92.9	88.4	78.0	68.7	75.1	24.1
PReFormer (Akwensi et al., 2024)	96.1	75.8	96.8	65.4	92.4	84.6	82.0	68.3	85.5	31.2
PointMamba (Liang et al., 2024)	95.6	72.2	96.2	65.0	92.7	87.5	71.7	60.4	93.5	10.6
PointCloudMamba (Zhang et al., 2024)	96.1	77.4	96.5	61.9	95.2	89.8	82.4	71.0	92.8	37.4
Ours	96.5	**79.4**	**97.0**	65.6	95.9	**91.5**	79.7	72.8	93.0	**39.9**

Table 6
Ablation study for local feature aggregation in 3D-UMamba.

Grouping Strategy	OA (%)	mIoU (%)	latency (ms)
Multi-scale ball-query	**97.2**	**78.1**	79.2
Single-scale ball-query	96.7	74.6	**49.7**

Table 7
Ablation study for VTS in 3D-UMamba, where PTS represents point-based token serialization without point cloud voxelization. In addition, the impact of voxel size (measured by the edge length of the voxel: 0.04, 0.06, 0.10, 0.20, and 0.40) are also explored.

Serialization Methods		OA (%)	mIoU (%)
PTS		96.8	77.0
VTS	0.04	96.9	76.5
	0.06	97.1	77.6
	0.10	**97.2**	**78.1**
	0.20	96.9	77.0
	0.40	96.4	75.9

Table 8
Ablation study for Bi-Scanning in 3D-UMamba, where H represents the number of the scanning directions.

H	OA (%)	mIoU (%)	latency (ms)
1	96.1	73.3	**76.6**
2	**97.2**	**78.1**	79.2
4	96.9	78.0	92.1
8	97.2	78.0	109.4

Table 9
Ablation study for the Bi-Scanning strategy. Four sets of Bi-Scanning strategies in different directions were applied to verify its effectiveness and robustness.

Bi-Scanning	OA (%)	mIoU (%)	latency (ms)
Fig. 7 (a)	**97.2**	**78.1**	79.2
Fig. 7 (b)	97.1	77.8	79.2
Fig. 7 (c)	96.8	78.0	79.2
Fig. 7 (d)	97.1	78.1	79.2

input point has 7 attributes: XYZ, RGB, and intensity. The mIoU and OA metrics were used for model evaluation, and IoU for each class was also given.

Performance Evaluation. Table 5 shows the detailed performance comparisons on the Toronto-3D dataset. The results demonstrate the excellent performance of 3D-UMamba in vehicle-mounted MLS point cloud semantic segmentation. Its outstanding performance in roadway infrastructure categories such as Road (mIoU of 97.0%), Building (mIoU of 91.5%), Natural (mIoU of 95.9%), and Car (mIoU of 93.0%) confirms its potential in traffic management, autonomous driving, and infrastructure maintenance.

4.5. Ablation study

This section conducted ablation studies for each main block in 3D-UMamba on the DALES dataset. In addition, we also discussed the sensitivities of the model to the token serialization approach. To evaluate the efficiency of 3D-UMamba compared with its Transformer-based counterpart, we introduced the latency, Model Parameters, FLOPs, and VRAM (i.e., Video Random Access Memory) usage metrics, where the latency is used to measure the time (ms) it takes the model to process a single sample.[1]

Local Feature Aggregation. As shown in Section 3.1, the local feature aggregation in 3D-UMamba is achieved by the ball-query grouping and feature average pooling methods. Specifically, we performed both multi-scale and single-scale ball-query grouping strategies for local neighborhood construction. As shown in Table 6, both the mIoU and OA of 3D-UMamba with multi-scale ball-query grouping are higher than those of 3D-UMamba with the single-scale strategy. This is because the multi-scale grouping strategy enables the model to capture richer context information of the target point cloud than the single-scale one, which results in better feature representation.

Voxel-based Token Serialization. We investigated a series of variants of the proposed VTS strategy, and further discussed the sensitivities of the model to the scanning directions. Table 7 shows the performance of 3D-UMamba with VTS and the vanilla Point-based Token Serialization (PTS) method, as well as explores the impact of voxel size on model performance. PTS is implemented by directly ordering points along X, Y, Z axes sequentially. From the results, VTS (78.1% of mIoU) achieves better results than PTS (77.0% of mIoU). Compared with PTS, VTS prevents over-reliance on a fixed input sequence and encourages the extraction of global structural features due to the random ordering within each voxel. This enhances the model's generalization ability to sequential tokens. In addition, this randomness reduces the impact of noise in LiDAR data by varying the positions of noisy points during training, thereby improving the model's robustness to noise. These advantages make VTS a robust method for feature extraction in noisy and complex LiDAR datasets. In addition, we also conducted ablation studies on the selection of the voxel size. As shown in the table, the model accuracy gradually increases with the increase of voxel size until the edge length of the voxel equals 0.1. As the size continues to increase, there is a drop in model accuracy. This is because an over-large voxel size tends to increase the randomness of the token serialization, thus hindering the Mamba's ability of global context modeling.

Table 8 shows the segmentation performance of 3D-UMamba under different numbers of the scanning directions (H represents the number

[1] The batch size is set to 1 when calculating FLOPs and VRAM usage in our experiments.

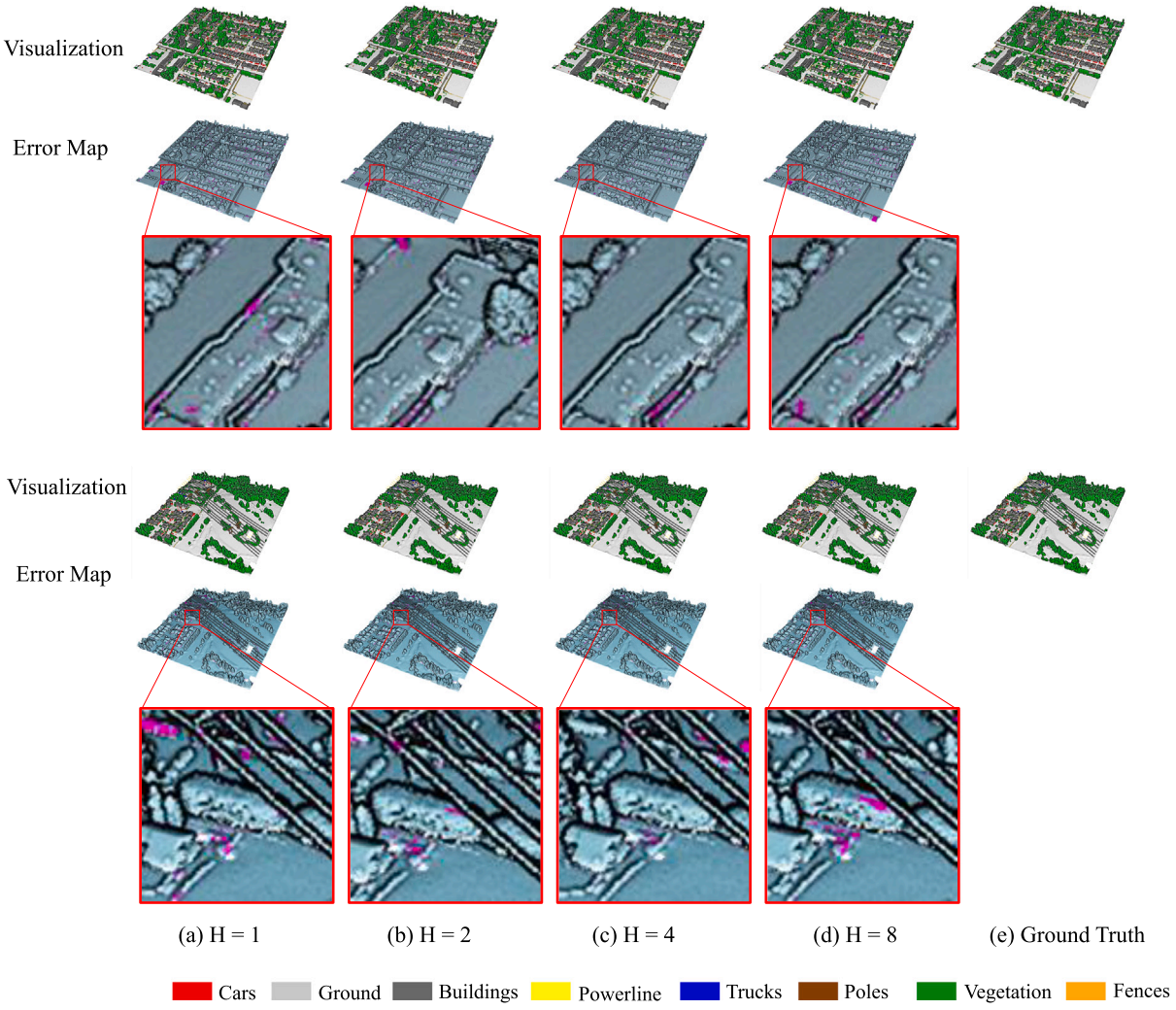


Fig. 6. Visualization result comparison of 3D-UMamba under different numbers (H) of the scanning directions on the DALES dataset, where our Bi-scanning strategy ($H = 2$) got the best performance. To compare model performance under different H more clearly, we also show error maps of the segmentation results compared to the ground truth, where the misclassified points are colored in red. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

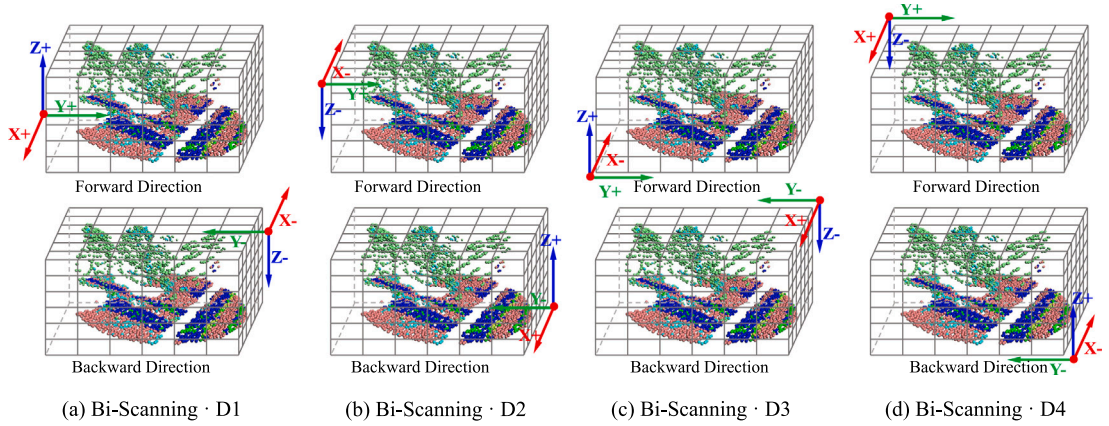


Fig. 7. Illustrations of the Bi-Scanning strategy in different directions. $D1 - D4$ represents four different bidirectional combinations.

of scanning directions, which is set to 1, 2, 4, and 8 respectively). Our Bi-scanning strategy ($H = 2$) achieves the best performance on the DALES dataset, with the mIoU of 78.1% and OA of 97.2%. The performance of 3D-UMamba with $H = 1$ is much lower than that with $H = 2$, which confirms the effectiveness of the Bi-Scanning strategy. As H increases, the model's performance approaches stability, while its efficiency gradually declines. Fig. 6 illustrates the visualization results. In addition, we also investigated the effectiveness and robustness of the Bi-Scanning strategy in different directions, which is illustrated in Fig. 7. The results are presented in Table 9. All four sets of Bi-Scanning strategies in different directions achieved excellent and stable results. Moreover, considering the unordered nature of point clouds,

Table 10
Ablation study for Mamba blocks in 3D-UMamba. The latency (ms), Params (MB), and FLOPs (GB) are used to measure models' efficiency and sizes.

Network	OA (%)	mIoU (%)	latency (ms)	Params (MB)	FLOPs (GB)
U-Net Baseline	96.2	69.8	**68.7**	**8.9**	**15.4**
U-Net + Transformer	96.7	74.2	81.4	21.5	16.8
U-Net + Mamba (Ours)	**97.2**	**78.1**	79.2	16.1	16.0

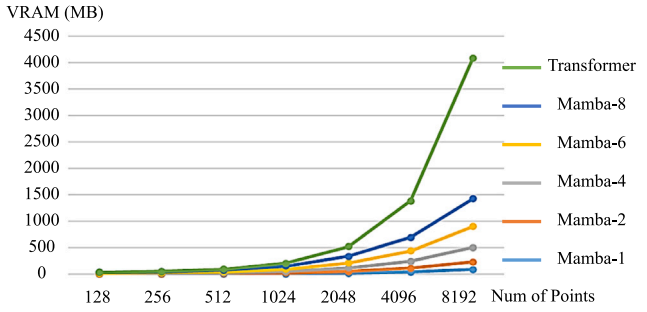


Fig. 8. VRAM usage comparison of Mamba with different scanning paths and Transformer blocks on different numbers of input points, where Mamba-2 represents the Mamba block with Bi-Scanning paths. The Transformer block shows a steeper increase in VRAM usage to the number of input points than the Mamba block, which confirms that Transformers have quadratic computational complexity, while Mamba operates with linear computational complexity.

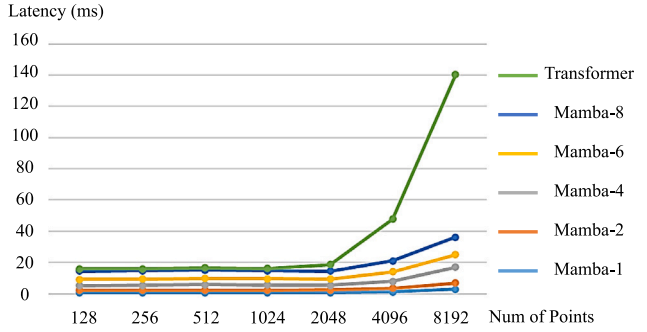


Fig. 9. Latency comparison of Mamba with different scanning paths and Transformer blocks on different numbers of input points.

we believe that 3D Mamba models still have room for improvement by optimizing the token serialization methods. This will be a potential research direction for our future work.

Mamba Block. We conducted ablation studies for the Mamba blocks, to demonstrate its superiority in LiDAR point cloud processing. As shown in Table 10 Row 1, we first removed the Mamba blocks from the 3D-UMamba. In this situation, a significant drop in mIoU (8.3 absolute percentage points) was observed, demonstrating Mamba's excellent performance in LiDAR point cloud learning. Secondly, we replaced the Mamba blocks with vanilla Transformer blocks, to compare their performance. By comparison, Mamba achieved slightly better performance than Transformer in terms of segmentation accuracy (78.1% vs. 74.2% in mIoU), and the former (79.2 ms, 16.1 MB) is faster and more lightweight than the latter (81.4 ms, 21.5 MB) as measured by latency and model parameters.

Computational Requirement Scaling. It is well understood that the Transformer block has quadratic complexity and the Mamba block has linear complexity. We conducted comparison experiments between the Mamba and Transformer block in terms of latency and VRAM usage, with the same input dimensions ($D = 320$). As shown in Figs. 8 and 9, we tested different numbers of input points from 128 to 8192 on the Mamba block and Transformer block. From the results, the VRAM usage and latency comparison of the blocks have similar trends. Specifically, the Mamba block with the Bi-Scanning strategy requires the fewer VRAM usage and lower latency than the Transformer block when the input number $N \geq 1024$. As N increases, the Mamba block outperforms the Transformer block by a larger and larger margin. Overall The VRAM and latency curves of the Transformer block exhibit quadratic growth as the number of input points increases, whereas Mamba demonstrates linear growth, which confirms that Transformers have quadratic computational complexity, while Mamba operates with linear computational complexity. These results demonstrate the superiority of Mamba in processing larger-scale point data.

Interestingly, we further evaluated Mamba blocks with different scanning strategies and observed that Mamba blocks with more scanning paths require higher VRAM usage and operate at slower speeds. This is reasonable, as an increased number of scanning paths necessitates more information processing and aggregating. Therefore, exploring an scanning strategy tailored for point cloud representation is meaningful to develop efficient Mamba networks.

4.6. Discussion

This paper conducted extensive experiments to evaluate the segmentation performance of 3D-Umamba on LiDAR point cloud data. As shown in Table 11, to fully explore the effectiveness of 3D-UMamba on multi-source LiDAR data, we used three datasets with different types of LiDAR data, i.e., airborne multi-spectral LiDAR data, aerial LiDAR data, and vehicle-mounted LiDAR data. Additionally, originating from different regions, these datasets encompass varied point cloud categories and scene complexities. This diversity highlights the model's ability to process and generalize across different LiDAR application scenarios, proving its versatility and robustness. Therefore, the selection of these three datasets underscores the strong generalization capability of 3D-UMamba as a general Mamba-based framework for LiDAR point cloud segmentation, addressing the challenges posed by data heterogeneity and varying scene complexities.

Extensive experiments on the aforementioned datasets demonstrate the superiority of 3D-Umamba on LiDAR data processing, compared to other benchmarked methods. Firstly, the comparison results on all three datasets show that 3D-Umamba significantly outperforms the traditional deep-learning methods, such as PointNet++ (Qi et al., 2017b) and KPConv (Thomas et al., 2019), in terms of both model accuracy and efficiency. This mainly lies in the ability of 3D-Umamba to integrate both local and global contextual information. While PointNet++ and KPConv focus primarily on local feature extraction, they neglect the critical aspect of capturing global context, which limits their representational capacity for complex point cloud data. In contrast, 3D-UMamba leverages Mamba, an efficient global context modeling technique, and incorporates it into a hierarchical network design. Each module of the 3D-UMamba framework effectively combines the Token Grouping and Aggregation (TGA) block for local feature learning with the Mamba block for global feature extraction. This comprehensive architecture enables 3D-UMamba to achieve robust and accurate point cloud feature extraction and representation, addressing the inductive bias of locality in traditional models and leading to superior segmentation performance.

Secondly, 3D-Umamba also outperforms Transformer-based point cloud processing methods, and achieves higher model efficiency and lower model size than its Transformer-based counterparts. Transformers rely on a self-attention mechanism that computes pairwise attention scores between all points, leading to a quadratic complexity of $\mathcal{O}(N^2 D)$. This makes Transformers computationally expensive, particularly for

Table 11
Comparison of the airborne MS-LiDAR, DALES, and Toronto-3D Datasets.

Aspect	MS-LiDAR	DALES	Toronto-3D
Collection Area	Canadian town	Urban and suburban areas in Canada	1 km roadway in Toronto, Canada
Resolution	~3.6 points/m^2	~50 points/m^2	~1,000 points/m^2
Classes	6 (road, building, grass, tree, soil, powerline)	8 (ground, vegetation, buildings, cars, trucks, poles, power lines, fences)	8 (road, road markings, natural, building, utility line, pole, car, fence)
Data Scale	13 areas, 15,000 m^2/area ~10 million points total	40 tiles, 0.5 km^2/tile ~500 million points total	4 sections, 250 m roadway/section 78.3 million points total
LiDAR Type	Multispectral	Aerial	Vehicle-mounted
Point Attributes	XYZ, intensity (MIR, NIR, Green)	XYZ, intensity	XYZ, intensity, RGB, GPS time, scan angle
Applications	Land cover classification, Urban planning, and Environmental monitoring	Urban modeling, Environmental monitoring	Roadway analysis, Autonomous driving, and HD mapping

large-scale point clouds, and imposes huge memory requirements. In contrast, Mamba is built on Selective State Space Models (S6), which achieves linear complexity $\mathcal{O}(N D)$ by leveraging recurrence relations to aggregate sequential information. This efficiency enables Mamba to handle larger input scales with reduced computational and memory costs. Additionally, the Selective Scan mechanism enables Mamba to adaptively determine the relevance of input tokens, selectively retaining essential information and discarding extraneous data. This selective processing allows Mamba to effectively model long-range dependencies without the computational burden associated with Transformer's self-attention. Consequently, Mamba achieves superior performance in processing long-sequence data, due to its ability to efficiently capture global context while maintaining linear computational complexity.

Finally, we compared 3D-Umamba with the most recent Mamba-based point cloud processing methods such as PointMamba (Liang et al., 2024) and PointCloudMamba (Zhang et al., 2024). The comparison results show that 3D-Umamba achieves better results than other Mamba-based methods. Compared to PointMamba, 3D-UMamba has a hierarchical framework for point cloud feature learning that integrates modules for both local and global contextual information extraction. In contrast, PointMamba employs a simpler design by cascading Mamba modules without leveraging hierarchical processing. This lack of hierarchical structure in PointMamba limits its ability to effectively combine local and global features, leading to a relatively weak feature learning and representation ability compared to 3D-Umamba. Since the hierarchical design of 3D-UMamba includes FPS operations, its processing efficiency is lower than PointMamba. When compared to PointCloudMamba, both networks use a hierarchical structure. However, on the one hand, 3D-UMamba incorporates TGA modules that support multi-scale feature extraction. This design enhances 3D-UMamba's adaptability to the complex scenarios in LiDAR data. On the other hand, the VTS method enhances the model's generalization to sequential data by introducing random ordering of points within each voxel. This randomization disrupts any fixed spatial short-range dependencies, preventing the network from overfitting to specific point orders and encouraging it to focus on learning global structures. The ablation studies in Section 4.5 also confirms it.

Based on the aforementioned excellent results of 3D-UMamba on multi-source LiDAR dataset, it holds significant potential for a variety of advanced remote sensing applications. For example, its accurate semantic segmentation results can be utilized for urban vegetation coverage assessment, aiding urban planning and environmental monitoring efforts. In addition, the method can also be applied to forest biomass estimation by segmenting key tree components such as trunks and branches, providing critical data for ecological and forestry studies. Furthermore, its ability to handle complex and large-scale point clouds makes it highly suitable for high-precision urban map modeling, supporting tasks such as infrastructure management and the development of navigation systems. These potential applications highlight the versatility and practicality of the proposed method in addressing diverse challenges within the field of remote sensing.

Section 4.5 provides a detailed comparison between Mamba and Transformer blocks in terms of VRAM usage and latency, which confirms the superiority of Mamba in large-scale point cloud processing. However, the 3D-Umamba has only a slight advantage over its Transformer-based counterpart in terms of model size and speed. This is mainly because point cloud sampling, local information extraction, and other operations in the framework limit the further improvement of the model efficiency. In addition, the ablation experiments demonstrate that the designs of token serialization and scanning strategies also play a critical role in Mamba-based methods, in terms of both model accuracy and efficiency. Therefore, it is necessary and meaningful to explore an elegant Mamba-based framework with a suitable token serialization and scanning method. In addition, since Mamba has an advantage over Transformer in processing large-scale token sequences, developing novel dataset preprocessing algorithms customized for Mamba is also a direction worth exploring.

5. Conclusion

This paper proposed a novel Mamba-based LiDAR point cloud semantic segmentation network with linear space complexity, named 3D-UMamba. It integrates the Mamba block into the classic U-Net architecture, forming a hierarchical encoder–decoder framework. Specifically, 3D-UMamba considers both local and global feature learning. Local feature learning is achieved by the ball-query grouping and neighbor feature pooling, while global feature learning is achieved by applying Mamba to the serialized points. To adapt the input point cloud to fit the Mamba block, we designed a simple yet effective serialization method (i.e., VTS) for transferring the unordered 3D points to 1D-sequence data along different directions. The Bi-Scanning strategy in the VTS block enables the model to capture global contexts from different directions. VTS reorders the tokens within the same voxels randomly during model training, which enhances the generalization capability of the model to sequential data while maintaining its strong performance of global context modeling. Dense experiments of three challenging LiDAR datasets (MS-LiDAR with 84.5% of mIoU, aerial DALES with 78.1% of mIoU, and vehicle-mounted Toronto-3D with 79.4% of mIoU) demonstrate the SOTA performance of 3D-UMamba in multi-source LiDAR point cloud semantic segmentation. Ablation studies show that 3D-UMamba also achieved comparable accuracy with its Transformer-based counterparts, with higher efficiency and lower memory costs as measured by latency and model parameters.

CRediT authorship contribution statement

Dening Lu: Writing – original draft, Visualization, Validation, Methodology, Investigation, Formal analysis, Data curation, Conceptualization. **Linlin Xu:** Writing – review & editing, Supervision, Resources, Methodology, Conceptualization. **Jun Zhou:** Writing – review & editing, Visualization, Methodology, Investigation, Data curation. **Kyle Gao:** Writing – review & editing, Investigation. **Zheng Gong:** Writing – review & editing, Investigation. **Dedong Zhang:** Writing – review & editing, Supervision, Conceptualization.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This work was supported in part by the Natural Sciences and Engineering Research Council of Canada (NSERC) Discovery Grant (No. RGPIN-2022-03741) and the Chinese Scholarship Council under Grant 202106830030.

Data availability

Our code has been released at https://github.com/d62lu/3D-UMamba.

References

Akwensi, P.H., Wang, R., Guo, B., 2024. Preformer: A memory-efficient transformer for point cloud semantic segmentation. Int. J. Appl. Earth Obs. Geoinf. 128, 103730. http://dx.doi.org/10.1016/j.jag.2024.103730.

Alnaggar, Y.A., Afifi, M., Amer, K., ElHelw, M., 2021. Multi projection fusion for real-time semantic segmentation of 3d lidar point clouds. In: Proc. IEEE/CVF Winter Conf. Appl. Comput. Vis.. pp. 1800–1809.

Boulch, A., 2020. ConvPoint: continuous convolutions for point cloud processing. Comput. Graph..

Cheng, H.X., Han, X.F., Xiao, G.Q., 2023. TransRVNet: LiDAR semantic segmentation with transformer. IEEE Trans. Intell. Transp. Syst. 24, 5895–5907.

Chollet, F., 2017. Xception: deep learning with depthwise separable convolutions. In: Proc. IEEE Conf. Comput. Vis. Pattern Recognit.. pp. 1251–1258.

Choy, C., Gwak, J., Savarese, S., Jun, 2019. 4D spatio-temporal convnets: Minkowski convolutional neural networks. In: Proc. IEEE Conf. Comput. Vis. Pattern Recognit.. pp. 3075–3084.

Fu, D.Y., Dao, T., Saab, K.K., Thomas, A.W., Rudra, A., Ré, C., 2023. Hungry hungry HIPPOs: towards language modeling with state space models. In: Proc. Int. Conf. Learn. Represent. https://openreview.net/forum?id=COZDy0WYGg.

Gu, A., Dao, T., 2023. Mamba: linear-time sequence modeling with selective state spaces. arXiv preprint arXiv:2312.00752.

Gu, A., Goel, K., Gupta, A., Ré, C., 2022a. On the parameterization and initialization of diagonal state space models. In: Proc. Adv. Neural Inf. Process. Syst. http://papers.nips.cc/paper_files/paper/2022/hash/e9a32fade47b906de908431991440f7c-Abstract-Conference.html.

Gu, A., Goel, K., Ré, C., 2022b. Efficiently modeling long sequences with structured state spaces. In: Proc. Int. Conf. Learn. Represent. https://openreview.net/forum?id=uYLFoz1vlAC.

Gu, A., Johnson, I., Timalsina, A., Rudra, A., Ré, C., 2023. How to train your HIPPO: state space models with generalized orthogonal basis projections. In: Proc. Int. Conf. Learn. Represent. https://openreview.net/forum?id=klK17OQ3KB.

Guo, M.H., Cai, J.X., Liu, Z.N., Mu, T.J., Martin, R.R., Hu, S.M., 2021. Pct: Point cloud transformer. Comput. Vis. Media 7, 187–199.

Gupta, A., Gu, A., Berant, J., 2022. Diagonal state spaces are as effective as structured state spaces. In: Proc. Adv. Neural Inf. Process. Syst.. http://papers.nips.cc/paper_files/paper/2022/hash/9156b0f6dfa9bbd18c79cc459ef5d61c-Abstract-Conference.html.

Han, X., Dong, Z., Yang, B., 2021. A point-based deep learning network for semantic segmentation of MLS point clouds. ISPRS J. Photogramm. Remote Sens. 175, 199–214.

Hui, L., Yang, H., Cheng, M., Xie, J., Yang, J., 2021. Pyramid point cloud transformer for large-scale place recognition. In: Proc. IEEE Int. Conf. Comput. Vis.. pp. 6098–6107.

Jing, Z., Guan, H., Zhao, P., Li, D., Yu, Y., Zang, Y., Wang, H., Li, J., 2021. Multispectral lidar point cloud classification using SE-PointNet++. Remote. Sens. 13 (2516), http://dx.doi.org/10.3390/RS13132516.

Lai, X., Liu, J., Jiang, L., Wang, L., Zhao, H., Liu, S., Qi, X., Jia, J., 2022. Stratified transformer for 3D point cloud segmentation. In: Proc. IEEE Conf. Comput. Vis. Pattern Recognit.. pp. 8500–8509.

Landrieu, L., Simonovsky, M., 2018. Large-scale point cloud semantic segmentation with superpoint graphs. In: Proc. IEEE Conf. Comput. Vis. Pattern Recognit.. pp. 4558–4567.

Li, Y., Bu, R., Sun, M., Wu, W., Di, X., Chen, B., 2018. PointCNN: convolution on X-transformed points. In: Proc. Adv. Neural Inf. Process. Syst.. pp. 828–838.

Li, Y., Ma, L., Zhong, Z., Cao, D., Li, J., 2019. TGNet: Geometric graph CNN on 3-D point cloud segmentation. IEEE Trans. Geosci. Remote Sens. 58, 3588–3600.

Li, G., Müller, M., Qian, G., Delgadillo, I.C., Abualshour, A., Thabet, A., Ghanem, B., 2023. Deepgcns: making GCNs go as deep as CNNs. IEEE Trans. Pattern Anal. Mach. Intell. 45, 6923–6939. http://dx.doi.org/10.1109/TPAMI.2021.3074057.

Liang, D., Zhou, X., Wang, X., Zhu, X., Xu, W., Zou, Z., Ye, X., Bai, X., 2024. Pointmamba: a simple state space model for point cloud analysis. arXiv preprint arXiv:2402.10739.

Liu, Y., Fan, B., Xiang, S., Pan, C., 2019a. Relation-shape convolutional neural network for point cloud analysis. In: Proc. IEEE Conf. Comput. Vis. Pattern Recognit.. pp. 8895–8904.

Liu, Z., Lin, Y., Cao, Y., Hu, H., Wei, Y., Zhang, Z., Lin, S., Guo, B., 2021. Swin transformer: Hierarchical vision transformer using shifted windows. In: Proc. IEEE/CVF Int. Conf. Comput. Vis.. pp. 10012–10022.

Liu, Z., Tang, H., Lin, Y., Han, S., 2019b. Point-voxel CNN for efficient 3D deep learning. In: Proc. Adv. Neural Inf. Process. Syst.. pp. 963–973.

Liu, Y., Tian, Y., Zhao, Y., Yu, H., Xie, L., Wang, Y., Ye, Q., Liu, Y., 2024. Vmamba: visual state space model. arXiv preprint arXiv:2401.10166.

Liu, Z., Yang, X., Tang, H., Yang, S., Han, S., 2023. FlatFormer: Flattened window attention for efficient point cloud transformer. In: Proc. IEEE Conf. Comput. Vis. Pattern Recognit.. pp. 1200–1211.

Lu, D., Gao, K., Xie, Q., Xu, L., Li, J., 2024a. 3DGTN: 3D dual-attention glocal transformer network for point cloud classification and segmentation. IEEE Trans. Geosci. Remote Sens..

Lu, D., Zhou, J., Gao, K.Y., Du, J., Xu, L., Li, J., 2024b. Dynamic clustering transformer network for point cloud segmentation. Int. J. Appl. Earth. Obs. Geoinf. 128, 103791.

Ma, L., Li, Y., Li, J., Tan, W., Yu, Y., Chapman, M.A., 2019. Multi-scale point-wise convolutional neural networks for 3D object segmentation from LiDAR point clouds in large-scale environments. IEEE Trans. Intell. Transp. Syst 22, 821–836.

Ma, J., Li, F., Wang, B., 2024. U-mamba: enhancing long-range dependency for biomedical image segmentation. arXiv preprint arXiv:2401.04722.

Ma, X., Qin, C., You, H., Ran, H., Fu, Y., 2022. Rethinking network design and local geometry in point cloud: A simple residual MLP framework.

Maturana, D., Scherer, S., 2015. Voxnet: A 3D convolutional neural network for real-time object recognition. In: Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.. pp. 922–928. http://dx.doi.org/10.1109/IROS.2015.7353481.

Milioto, A., Vizzo, I., Behley, J., Stachniss, C., 2019. Rangenet++: Fast and accurate lidar semantic segmentation. In: Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.. IEEE, pp. 4213–4220.

Nikoohemat, S., Diakité, A.A., Zlatanova, S., Vosselman, G., 2020. Indoor 3D reconstruction from point clouds for optimal routing in complex buildings to support disaster management. Autom. Constr. 113, 103109.

Özçelik, R., de Ruiter, S., Criscuolo, E., Grisoni, F., 2024. Chemical language modeling with structured state space sequence models. Nat. Commun. 15 (6176).

Park, C., Jeong, Y., Cho, M., Park, J., 2022. Fast point transformer. In: Proc. IEEE Conf. Comput. Vis. Pattern Recognit.. pp. 16949–16958.

Pasternak, G., Zaczek-Peplinska, J., Pasternak, K., Jóźwiak, J., Pasik, M., Koda, E., Vaverková, M.D., 2023. Surface monitoring of an MSW landfill based on linear and angular measurements, TLS, and LiDAR UAV. Sensors 23, 1847.

Qi, C.R., Su, H., Mo, K., Guibas, L.J., 2017a. PointNet: Deep learning on point sets for 3D classification and segmentation. In: Proc. IEEE Conf. Comput. Vis. Pattern Recognit.. pp. 652–660.

Qi, C.R., Yi, L., Su, H., Guibas, L.J., 2017b. Pointnet++: deep hierarchical feature learning on point sets in a metric space. Adv. Neural Inf. Process. Syst. 30.

Qian, G., Li, Y., Peng, H., Mai, J., Hammoud, H., Elhoseiny, M., Ghanem, B., 2022. Pointnext: Revisiting pointnet++ with improved training and scaling strategies. In: Proc. Adv. Neural Inf. Process. Syst. http://papers.nips.cc/paper_files/paper/2022/hash/9318763d049edf9a1f2779b2a59911d3-Abstract-Conference.html.

Riegler, G., O. Ulusoy, A., Geiger, A., 2017. OctNet: Learning deep 3D representations at high resolutions. In: Proc. IEEE Conf. Comput. Vis. Pattern Recognit.. pp. 3577–3586. http://dx.doi.org/10.1109/CVPR.2017.701.

Robert, D., Raguet, H., Landrieu, L., 2023. Efficient 3D semantic segmentation with superpoint transformer. In: Proc. IEEE Int. Conf. Comput. Vis.. pp. 17149–17158.

Robert, D., Raguet, H., Landrieu, L., 2024. Scalable 3d panoptic segmentation with superpoint graph clustering. arXiv preprint arXiv:2401.06704.

Ronneberger, O., Fischer, P., Brox, T., 2015. U-net: convolutional networks for biomedical image segmentation. In: Med. Image. Comput. Comput. Assist. Interv.. Springer, pp. 234–241.

Shi, J., Xiang, M., 2024. Convolution SSM model for text emotion classification. In: International Symposium on Computer Applications and Information Systems. SPIE, pp. 554–560.

Stilla, U., Xu, Y., 2023. Change detection of urban objects using 3D point clouds: a review. ISPRS J. Photogramm. Remote Sens. 197, 228–255.

Sun, J., Qing, C., Tan, J., Xu, X., 2023. Superpoint transformer for 3D scene instance segmentation. In: AAAI Conf. Artif. Intell.. pp. 2393–2401.

Tan, W., Qin, N., Ma, L., Li, Y., Du, J., Cai, G., Yang, K., Li, J., 2020. Toronto-3D: a large-scale mobile LiDAR dataset for semantic segmentation of urban roadways. In: Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops. pp. 202–203.

Thomas, H., Qi, C.R., Deschaud, J.E., Marcotegui, B., Goulette, F., Guibas, L.J., 2019. Kpconv: Flexible and deformable convolution for point clouds. In: Proc. IEEE Int. Conf. Comput. Vis.. pp. 6411–6420.

Varney, N., Asari, V.K., Graehling, Q., 2020. DALES: a large-scale aerial LiDAR data set for semantic segmentation. In: Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops. pp. 186–187.

Wan, J., Xu, Y., Qiu, Q., Xie, Z., 2023. A geometry-aware attention network for semantic segmentation of MLS point clouds. Int. J. Geogr. Inf. Sci. 37, 138–161.

Wang, Z., Chen, Z., Wu, Y., Zhao, Z., Zhou, L., Xu, D., 2024b. Pointramba: a hybrid transformer-mamba framework for point cloud analysis. arXiv preprint arXiv:2405.15463.

Wang, L., Huang, Y., Hou, Y., Zhang, S., Shan, J., 2019a. Graph attention convolution for point cloud semantic segmentation. In: Proc. IEEE Conf. Comput. Vis. Pattern Recognit.. pp. 10296–10305.

Wang, H., Shi, C., Shi, S., Lei, M., Wang, S., He, D., Schiele, B., Wang, L., 2023. Dsvt: Dynamic sparse voxel transformer with rotated sets. In: Proc. IEEE Conf. Comput. Vis. Pattern Recognit.. pp. 13520–13529.

Wang, Y., Sun, Y., Liu, Z., Sarma, S.E., Bronstein, M.M., Solomon, J.M., 2019b. Dynamic graph CNN for learning on point clouds. ACM Trans. Graph. 38, 1–12. http://dx.doi.org/10.1145/3326362.

Wang, C., Tsepa, O., Ma, J., Wang, B., 2024a. Graph-mamba: towards long-range graph sequence modeling with selective state spaces. arXiv preprint arXiv:2402.00789.

Wu, X., Jiang, L., Wang, P., Liu, Z., Liu, X., Qiao, Y., Ouyang, W., He, T., Zhao, H., 2024. Point transformer V3: simpler, faster, stronger. In: Proc. IEEE Conf. Comput. Vis. Pattern Recognit.. IEEE, pp. 4840–4851.

Wu, W., Qi, Z., Fuxin, L., 2019a. PointConv: Deep convolutional networks on 3D point clouds. In: Proc. IEEE Conf. Comput. Vis. Pattern Recognit.. pp. 9621–9630. http://dx.doi.org/10.1109/CVPR.2019.00985.

Wu, B., Wan, A., Yue, X., Keutzer, K., 2018. Squeezeseg: Convolutional neural nets with recurrent crf for real-time road-object segmentation from 3d lidar point cloud. In: Proc. IEEE Int. Conf. Robot. Autom.. IEEE, pp. 1887–1893.

Wu, B., Zhou, X., Zhao, S., Yue, X., Keutzer, K., 2019b. Squeezesegv2: Improved model structure and unsupervised domain adaptation for road-object segmentation from a lidar point cloud. In: Proc. IEEE Int. Conf. Robot. Autom.. IEEE, pp. 4376–4382.

Xiang, B., Wielgosz, M., Kontogianni, T., Peters, T., Puliti, S., Astrup, R., Schindler, K., 2024. Automated forest inventory: Analysis of high-density airborne LiDAR point clouds with 3D deep learning. Remote Sens. Environ. 305, 114078.

Xiao, W., Cao, H., Tang, M., Zhang, Z., Chen, N., 2023. 3D urban object change detection from aerial and terrestrial point clouds: a review. Int. J. Appl. Earth. Obs. Geoinf. 118, 103258.

Xiao, K., Qian, J., Li, T., Peng, Y., 2022. Multispectral LiDAR point cloud segmentation for land cover leveraging semantic fusion in deep learning network. Remote. Sens. 15 (243).

Xu, C., Wu, B., Wang, Z., Zhan, W., Vajda, P., Keutzer, K., Tomizuka, M., 2020. Squeezesegv3: Spatially-adaptive convolution for efficient point-cloud segmentation. In: Proc. Eur. Conf. Comput. Vis.. Springer, pp. 1–19.

Zhang, T., Li, X., Yuan, H., Ji, S., Yan, S., 2024. Point cloud mamba: point cloud learning via state space model. arXiv preprint arXiv:2403.00762.

Zhang, Q., Peng, Y., Zhang, Z., Li, T., 2023b. Semantic segmentation of spectral LiDAR point clouds based on neural architecture search. IEEE Trans. Geosci. Remote Sens. 1–11. http://dx.doi.org/10.1109/TGRS.2023.3284995.

Zhang, C., Wan, H., Shen, X., Wu, Z., 2022. Patchformer: an efficient point transformer with patch attention. In: Proc. IEEE Conf. Comput. Vis. Pattern Recognit.. pp. 11799–11808.

Zhang, D., Zheng, Z., Niu, H., Wang, X., Liu, X., 2023a. Fully sparse transformer 3D detector for LiDAR point cloud. IEEE Trans. Geosci. Remote Sens..

Zhang, Y., Zhou, Z., David, P., Yue, X., Xi, Z., Gong, B., Foroosh, H., 2020. Polarnet: An improved grid representation for online lidar point clouds semantic segmentation. In: Proc. IEEE Conf. Comput. Vis. Pattern Recognit.. pp. 9601–9610.

Zhao, P., Guan, H., Li, D., Yu, Y., Wang, H., Gao, K., Junior, J.M., Li, J., 2021b. Airborne multispectral LiDAR point cloud classification with a feature reasoning-based graph convolution network. Int. J. Appl. Earth Obs. Geoinf. 105, 102634.

Zhao, H., Jiang, L., Jia, J., Torr, P.H., Koltun, V., 2021a. Point transformer. In: Proc. IEEE Int. Conf. Comput. Vis.. pp. 16259–16268.

Zhou, H., Feng, Y., Fang, M., Wei, M., Qin, J., Lu, T., 2021. Adaptive graph convolution for point cloud analysis. In: Proc. IEEE Int. Conf. Comput. Vis.. pp. 4965–4974.