S^{2}PT: Spatio-Sequential Point Transformer for Efficient 3D Scene Understanding

Yuqing Shen[1], Akram Akbar[1,2], Yijun Liu[1], Yanyi Li[1], Chun Liu[1]

[1]College of Surveying and Geo-informatics, Tongji University, China - (yuqingsh, shalll, 2310267, liuchun)@tongji.edu.cn
[2]College of Electronic and Information Engineering, Tongji University, China - akram@tongji.edu.cn

Keywords: Point Cloud, Geospatial AI, Semantic Segmentation, 3D Scene Understanding, Transformer

Abstract

Efficient processing of large-scale 3D point clouds acquired from Terrestrial or Airborne Laser Scanning (TLS/ALS) presents a significant computational challenge. While transformer-based architectures excel at modeling the global context crucial for interpreting these complex scenes, their quadratic computational complexity makes them infeasible for direct application on massive point sets. To address this scalability bottleneck, we propose the **Spatio-Sequential Point Transformer (S^{2}PT)**, a novel hierarchical architecture for efficient and effective large-scale point cloud processing. Our approach begins by serializing the point cloud into an ordered sequence, which enables the use of attention with linear complexity. This not only circumvents the quadratic bottleneck of standard transformers but also establishes a global receptive field at every layer. To compensate for potential information loss during serialization, we further introduce a novel Spatio-Sequential Positional Encoding (S^{2}PE) that synergistically combines 3D local geometric features with 1D sequential order information, enhancing the model's spatial awareness. Experiments on multiple benchmarks demonstrate that S^{2}PT achieves performance comparable to state-of-the-art methods while being significantly more efficient during training and inference, offering a promising path towards scalable representation learning for large-scale 3D scenes.

1. Introduction

The processing of large-scale 3D point clouds, acquired from technologies like TLS/ALS and photogrammetry, has become a cornerstone for critical applications such as high-definition (HD) mapping, ecological monitoring and city digital twins (Tan et al., 2020, Oehmcke et al., 2022, Shen et al., 2025). Early works effectively tackled the unstructured nature of point clouds, establishing two main paradigms (Guo et al., 2021b): voxel-based convolutions (Maturana and Scherer, 2015, Lang et al., 2019, Zhou and Tuzel, 2018, Chen et al., 2023) and point-based architectures (Qi et al., 2017a, Qi et al., 2017b, Thomas et al., 2019, Qian et al., 2022). While both approaches have made significant strides, they commonly struggle to efficiently model long-range dependencies across large-scale scene-level point clouds. Following the groundbreaking success in natural language processing (Devlin et al., 2019, Brown et al., 2020) and computer vision (Dosovitskiy et al., 2021, Liu et al., 2021, Carion et al., 2020) of transformer (Vaswani et al., 2017), researchers began to integrate this powerful attention-based model into 3D tasks, hoping transformers could address the global context modeling challenge (Zhao et al., 2021, Guo et al., 2021a).

The self-attention mechanism is architecturally well-suited for point cloud related tasks, as it is not only permutation-equivariant but also excels at modeling complex, long-range dependencies in unordered point sets (Lahoud et al., 2022, Thengane et al., 2025). However, the modeling power comes with a prohibitive cost - the computational and memory costs of the attention layer scale quadratically with the number of points, rendering it computationally infeasible for the real-world large-scene point clouds.

To circumvent this limitation, current leading methods have converged on a common compromise: approximating global attention by restricting it within localized neighborhoods (Lahoud et al., 2022, Zhao et al., 2021, Lai et al., 2022, Wu et al., 2022, Yang et al., 2025, Wu et al., 2024). While these methods have achieved great success in different down-stream tasks, the reliance on localization inherently constrains the model's receptive field at each layer. This tradeoff forces long-range context to be captured only implicitly through the network's depth, lacking explicit global interaction at a single scale.

To address this fundamental limitation, we propose the **Spatio-Sequential Point Transformer (S^{2}PT)**, a novel hierarchical architecture designed for efficient and effective large-scale point cloud processing. We begin by introducing **focused linear attention** (Han et al., 2023) to process the serialized point clouds, which fundamentally reduces the spatial complexity of the attention operation from quadratic to linear. To the best of our knowledge, S^{2}PT is the first framework to adapt focused linear attention to large-scale point cloud understanding. This spatial advantage liberates our model from the need to partition the point sequence into isolated patches, enabling attention computation over the entire feature representation. This holistic approach yields a more consistent feature representation in WHU-Railway dataset (Qiu et al., 2024), free of the processing artifacts common in patch-based methods as presented in Figure 1.

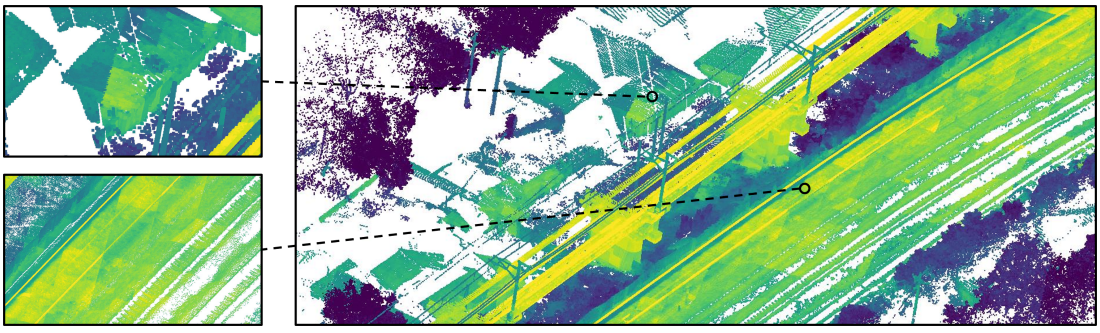
To unlock the full potential of linear attention on serialized point clouds, which can otherwise suffer from performance degradation, we introduce the **Spatio-Sequential Positional Encoding (S^{2}PE)**. This novel positional encoding mechanism enhances the model's spatial awareness from two complementary perspectives: 3D local geometry and 1D sequential order. This synergy allows S^{2}PT to push the Pareto Frontier in scene-level point cloud processing, achieving performance comparable to state-of-the-art methods while being significantly more efficient during training and inference.

2. Related Work

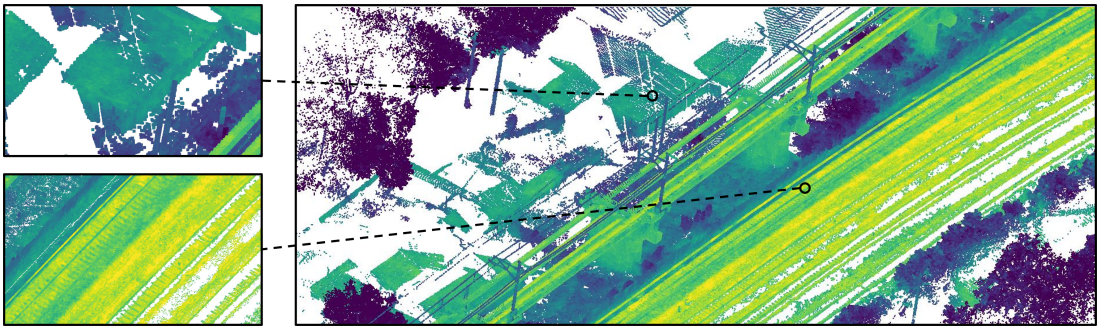
2.1 Point Cloud Transformer

The adaptation of transformers to 3D point clouds, driven by the need to model long-range dependencies, diverged into two main

This contribution has been peer-reviewed. The double-blind peer-review was conducted on the basis of the full paper.
https://doi.org/10.5194/isprs-annals-XI-4-2026-137-2026



(a) PTv3 result



(b) Our result

Figure 1. PCA visualization result of WHU-Railway dataset. (a) Block artifacts appear in PTv3; (b) Ours is more consistent.

paradigms, distinguished by the foundational data representation and strategy for managing computational complexity.

- **Voxel-Based Partitioning**: Methods falling into this category convert the unstructured point cloud into a regular grid, thereby structuring the input for attention mechanisms. Points within each grid cell (voxel) are typically abstracted into a representative token. OctFormer (Cui et al., 2023), for example, performs attention directly on the features of the octree nodes built with sparse point clouds. Swin3D (Yang et al., 2025) and its scaled version Swin3D++ (Yang et al., 2025) adapt the successful Swin Transformer (Liu et al., 2021) to 3D tasks, employing a hierarchical design with shifting window attention on voxelized features to effectively capture aggregated context.

- **Data-Driven Clustering**: This paradigm extends the philosophy of point-based networks by operating directly on raw coordinates to form adaptive, irregular neighborhoods. The pioneering Point Transformer (Zhao et al., 2021) applies vector-based self-attention within local neighborhoods found via k-Nearest Neighbors (k-NN), effectively modeled fine-grained spatial relationships. Point Transformer V2 (Wu et al., 2022) improves upon its predecessor by introducing grouped vector attention and a more effective partition-based pooling strategy that initiates from local point neighborhoods. Similarly, stratified Transformer (Lai et al., 2022) proposed a stratified sampling strategy that samples keys densely from nearby and sparsely from distant points to allow a flexible receptive field.

A significant paradigm shift was presented by Point Transformer V3 (PTv3) (Wu et al., 2024) which reframed the problem by downsampling and serializing unordered dense point clouds into ordered sequences using space-filling curves (SFCs). By applying attention within fixed-size 1D patches along this sequence, PTv3 proposed a mix of the two paradigms.

2.2 Linear Attention

The quadratic complexity of self-attention, a bottleneck for long sequences that was acknowledged and discussed in the transformer work (Vaswani et al., 2017) has catalyzed extensive research into a more efficient version of transformers.A significant branch of this research has focused on Linear Attention, which aims to reduce the complexity of self-attention from $O(N^2)$ to $O(N)$ with respect to sequence length N. Originating in the field of NLP to handle ever-growing context windows, this pursuit of efficiency has led to a diverse family of methods (Katharopoulos et al., 2020, Choromański et al., 2021, Wang et al., 2020, Hua et al., 2022, Sun et al., 2023). Subsequently, these principles were successfully adapted and further developed for 2D computer vision, enabling transformer-based model to efficiently process high-resolution images (Zhai et al., 2021, Fan et al., 2024, Han et al., 2023). While these efficient attention mechanisms have demonstrated remarkable success, their application to 3D point cloud processing remains a largely unexplored frontier. This gap is particularly striking, as point clouds arguably represent the domain where efficient, long-context attention is most critically needed. Unlike texts or images, large scene point clouds can easily contain more than hundreds of millions of points. To our knowledge, while linear attention has been explored for specific 3D object detection tasks (Liu et al., 2023), its potential for creating a general-purpose, efficient backbone for point cloud representation learning has not been systematically investigated.

2.3 3D Positional Encoding

The absence of inherent order in point clouds makes positional encoding a cornerstone of any successful point cloud transformer architecture. Unlike in NLP where token order is explicit, 3D models must encode spatial coordinates to inform the self-attention mechanism of the underlying geometric structure. The strategies for achieving this have evolved significantly. An early approach (Yu et al., 2022) directly treated point coordinates as continuous features and projected them into the embedding

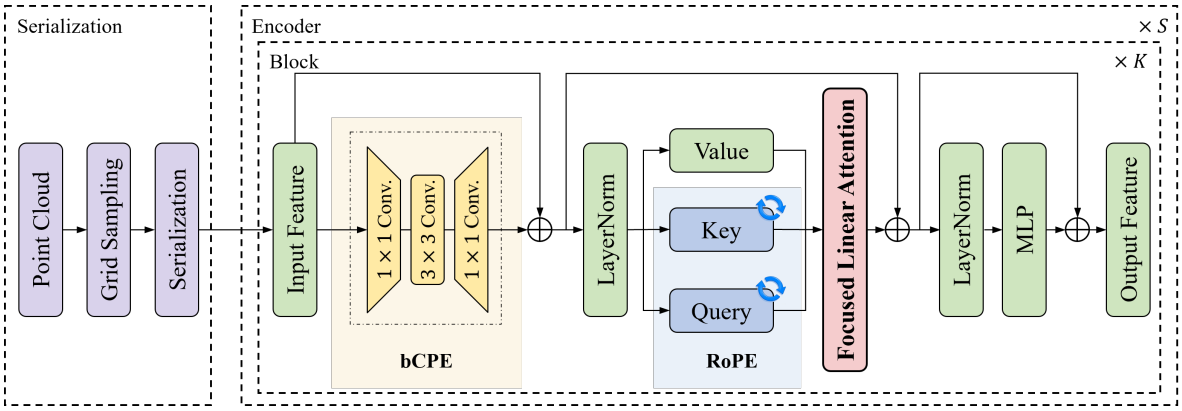


Figure 2. Architecture of our S^{2}PT encoder block. A serialized point cloud is processed using linear attention, with positional information injected by two modules: bCPE for local 3D geometry and RoPE for 1D sequential order.

space. While straightforward, this method of explicit coordinate embedding often struggles to effectively convey fine-grained relative positional information, which is crucial for local feature learning. Subsequent works (Lai et al., 2022, Yang et al., 2025, Zhao et al., 2021) shifted towards encoding relative positions through pre-grouping points into local neighborhoods. This explicit relative positional encoding (RPE) proved highly effective but faces significant challenges in computational expenses and limitation in receptive field. Later research (Cui et al., 2023, Wu et al., 2024) revealed that RPE is essentially large-kernel sparse convolution in 3D space, and thus spatial information can be encoded through a learnable, data-drive module. This paradigm is regarded as Conditional Positional Encoding (CPE).

3. Method

The application of transformers to 3D point clouds is fundamentally constrained by two factors: the quadratic complexity of self-attention and the insufficiency in space feature extraction. To tackle these issues, we propose a transformer-based architecture **Spatio-Sequential Point Transformer** for efficient and robust 3D understanding as presented in Figure 2. Our approach begins by serializing large-scale point clouds into an ordered sequence, a process that enables the application of attention with linear complexity, thereby circumventing the quadratic bottleneck of standard Transformer and allowing for a global receptive field over the entire point cloud at every layer. Crucially, we further enhance our model's spatial awareness with innovation in the mechanism of positional encoding, combining 3D local geometric features with 1D sequential order information.

3.1 Point Cloud Serialization

A fundamental challenge in applying standard transformer architectures to point clouds lies in their nature as unordered sets. This property mandates that any networks designed to process point clouds must exhibit permutation invariance (Qi et al., 2017a). To address this, we follow PTv3 (Wu et al., 2024) by randomly applying Z or Hilbert serialization strategy during training. Furthermore, we also alter the traversal sequence of x, y and z axes during serialization process to ensure our model is not overly fitted to a specific space-filling curve pattern.

3.2 Linear Attention

Attention layer is the core of any transformer architecture. Given a series X, the standard attention (Vaswani et al., 2017) value takes the form:

$$\mathbf{O}_{attn}(Q,K,V) = \text{softmax}(\frac{QK^T}{\sqrt{d_k}})V \tag{1}$$

The $O(N^2)$ space and computational complexity of this softmax attention makes it infeasible for large-scale point clouds, which can easily contain hundreds of thousands of points and would thus exhaust GPU memory. The main bottleneck lies in the computation of the square matrix $A = QK^T$, which contains the raw pairwise attention scores. Therefore, we replace softmax attention with linear attention (Katharopoulos et al., 2020) to improve data efficiency as shown in Figure 3(a). The core idea is to approximate softmax function with kernel function $\phi(\cdot)$, such that the attention score can be rewritten as a dot product of transformed queries and keys:

$$\text{softmax}(QK^T) \approx \phi(Q)\phi(K)^T \tag{2}$$

This approximation allows for a change in the computation order to avoid the calculation of $N \times N$ score matrix. The output is then computed as:

$$\mathbf{O}'_{attn}(Q,K,V) = \phi(Q)(\frac{1}{N}\phi(K)^T V) \tag{3}$$

where N is the length of the sequence. Notice that the term $\frac{1}{N}\phi(K)^T V$ can be regarded as a global context vector. It is only computed once from the entire sequence and can be reused for every query, reducing the complexity to $O(N)$.

To address the performance gap due to the absence of the softmax function, we turned to the 2D computer vision, where focused linear attention (Han et al., 2023) designed an exquisite kernel function to mimic the winner-take-it-all effect of softmax function. The kernel function is written as:

$$\phi(x) = \frac{||f(x)||}{||f(x)^{**p}||} f(x)^{**p} \tag{4}$$

where f is ReLU function. We further adapted this method for serialized point cloud series with a gated fusion mechanism between the attention output and local feature extracted by a depth-wise convolution along the point cloud sequence. The gate $\mathbf{G}$ and the final attention output is denoted as:

$$\mathbf{G} = \sigma(\mathbf{O}_{attn} + \mathbf{O}_{local}) \tag{5}$$

$$\mathbf{O}_{fused} = (1 - \mathbf{G}) \odot \mathbf{O}_{attn} + \mathbf{G} \odot \mathbf{O}_{local} \tag{6}$$

This contribution has been peer-reviewed. The double-blind peer-review was conducted on the basis of the full paper.
https://doi.org/10.5194/isprs-annals-XI-4-2026-137-2026 | © Author(s) 2026. CC BY 4.0 License.

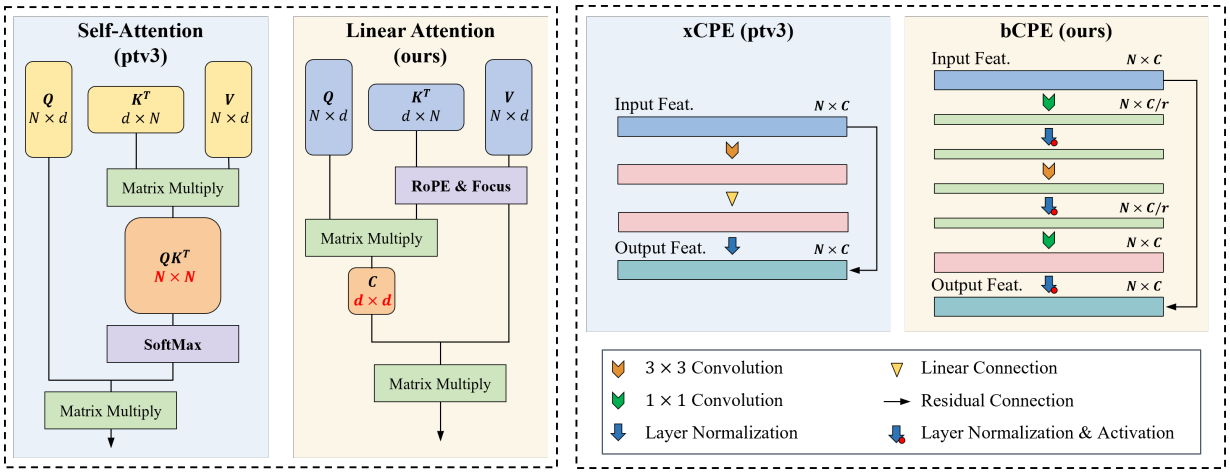


Figure 3. Comparison of core components between our S^{2}PT and PTv3. (a) Our Linear Attention avoids the quadratic complexity of standard self-attention used in PTv3. (b) Our bCPE is more parameter-efficient than the standard CPE (xCPE) used in PTv3

3.3 Spatio-Sequential Positional Encoding (S^{2}PE)

We introduce a positional encoding component by utilizing both spatial position information and the relative position between points in SFC.

In each transformer block, we follow the practice of using sparse convolution layers to extract local features. We present a conditional positional encoding (CPE) in bottleneck shape to reduce parameter amount. Our CPE module projects high-dimensional features into a low-dimensional ”bottleneck” subspace using $1\times1\times1$ sparse convolution as shown in Figure 3(b). Within this compressed, information-dense space, a $3\times3\times3$ convolution then efficiently performs spatial mixing to better capture local geometric patterns. Finally, another $1\times1\times1$ convolution projects the features back to its original dimension. With this bottleneck design, we encode the same rich spatial information with a reduced number of parameters.

However, while CPE captures local 3D geometry, it does not explicitly encode the long-range sequential order of SFC. Thus, we employ Rotary Positional Encoding (RoPE) (Su et al., 2024) to the query and key vectors before attention computation to inject relative position information. Given a point vector $\mathbf{x} \in \mathbb{R}^d$ of dimension d at position m, RoPE conceptually splits the features into $d/2$ pairs of features (x_{2i}, x_{2i+1}). And then a rotation, dependent on position m is applied to each pair:

$$\mathbf{R}(\mathbf{x}, m)_i = \begin{pmatrix} \cos(m\theta_i) & -\sin(m\theta_i) \\ \sin(m\theta_i) & \cos(m\theta_i) \end{pmatrix} \begin{pmatrix} \mathbf{x}_{2i} \\ \mathbf{x}_{2i+1} \end{pmatrix} \tag{7}$$

where $\theta_i = 10000^{-2i/d}$ is a predefined frequency for the i-th pairs. The key advantage of RoPE in standard attention is that for any two vectors q at position m and k at position n, the rotated dot product depends only on their relative position $m-n$. While this property is the cornerstone of its success in standard self-attention, a non-trivial question is whether this advantage translates to our linear attention framework, which avoids direct pairwise dot-products. We now prove that this property is not only preserved, but is central to the effectiveness of our proposed model.

Proof: Consider two vectors q at position m and k at position n. They are firstly rotated by the RoPE:

$$q'_m = \mathbf{R}(q_m, m)$$
$$k'_n = \mathbf{R}(k_n, n)$$

The output o at position m can then be written as:

$$\begin{aligned} o_m^T &= {q'_m}^T (\sum_{n=1}^{N} k'_n v_n^T) \\ &= ({q'_m}^T k'_1) v_1^T + ({q'_m}^T k'_2) v_2^T + ... + ({q'_m}^T k'_n) v_n^T \end{aligned} \tag{8}$$

Notice that

$$\forall n \in [1, N], {q'_m}^T k'_n = \mathbf{R}(q_m, m)^T \cdot \mathbf{R}(k_n, n)$$

We can now apply RoPE's dot product property on our attention output, i.e. $\mathbf{R}(q_m, m)^T \cdot \mathbf{R}(k_n, n) = \mathbf{R}(q_m^T k_n, m-n)$. Therefore, we have that

$$o_m = \sum_{n=1}^{N} \mathbf{R}(q_m^T k_n, m-n) v_n \tag{9}$$

This final form unequivocally demonstrates that the core property of RoPE is preserved within our linear attention framework. The interaction strength between any two points is explicitly modulated by their relative distance along the SFC sequence. This allows our model to effectively capture long-range dependencies and understand the global structure of the point cloud.

4. Experiments

4.1 Comparison

To provide a comprehensive evaluation, we compare our method with the family of Point Transformer (Zhao et al., 2021, Wu et al., 2022, Wu et al., 2024) and the representative of non-transformer architecture Minkowski U-Net (Choy et al., 2019). While PT-v3 employs standard softmax attention with local windowing to maintain high accuracy, it suffers from quadratic computational complexity and limited scalability in large-batch settings. In contrast, our S^{2}PT replaces softmax attention with linear attention, which inherently introduces a potential performance drop due to the loss of exact attention weights. However, through the

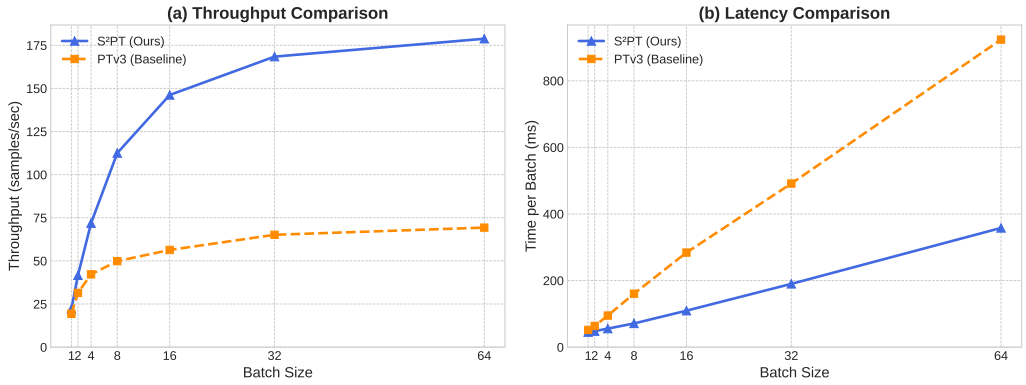


Figure 4. Performance comparison between our S^{2}PT and the baseline across various batch sizes.

Datasets	Model	mIoU	mAcc	allAcc
S3DIS	MinkUNet	66.08	71.31	80.92
	PTv1	68.29	73.51	90.47
	PTv2	70.92	75.96	90.78
	PTv3	71.15	76.51	91.13
	S^{2}PT	**71.24**	**76.53**	**91.23**
nuScenes	MinkUNet	76.45	82.07	93.70
	PTv3	**80.40**	**86.66**	**94.66**
	S^{2}PT	79.82	86.40	94.43
ModelNet40	PTv3	—	**89.79**	92.79
	S^{2}PT	—	89.58	**92.83**

Table 1. Classification on ModelNet40 & Segmentation on S3DIS and nuScenes.

design of S^{2}PE and the fusion of local convolutional features, we effectively mitigate this degradation. As shown in Table 1, S^{2}PT achieves comparable accuracy to PT-v3 across multiple benchmarks, and nearly identical performance on ModelNet40 - while operating under a fundamentally more scalable architecture.

4.2 Efficiency and Scalability Analysis

To empirically validate the core efficiency claim of our proposed architecture, we conducted a thorough comparison against our main baseline, PT-v3 (Wu et al., 2024), on 4 NVIDIA L40 GPUs.

First, we evaluated the training throughput across different workloads, as presented in Figure 4. S^{2}PT consistently outperforms PT-v3, and the performance gap widens significantly as the batch size increases. This highlights the superior scalability of our linear attention mechanism. At its peak, our method achieves a throughput of approximately 175 samples/sec, representing a 2.58x speedup over PT-v3's maximum throughput. In terms of inference latency, S^{2}PT processes a single scan (batch size 1) in 45.1 ms, outperforming PT-v3's 51.7 ms. Notably, this efficiency advantage scales dramatically with workload: at maximum load (batch size 64), S^{2}PT maintains an average processing time of just 5.6 ms per sample, effectively distinguishing itself from the quadratic complexity bottleneck of standard transformers.

Model	VRAM (MB)	Time (H:M:S)
PT-v3 (Baseline)	80,430	3:20:49
S^{2}PT (Ours)	**4,268**	**1:37:03**

Table 2. Resource Usage Comparison.

In addition to throughput, we measured the peak GPU memory footprint and the total time required to complete a standard training run (e.g., 100 epochs) under a typical setting with a batch size of 16. The results indicate that our method requires only 4.3 GB of VRAM at this setting, making it accessible on a wide range of consumer-level GPUs. In contrast, PT-v3 consumes over 80 GB of VRAM, restricting its use to high-end GPU and making larger batch sizes infeasible for most users (Table 2). This dramatic reduction in memory consumption directly translates to faster training, with S^{2}PT completing the task in approximately half the time required by PT-v3, achieving a 2.1x overall training speedup.

4.3 Ablation Study

To validate the effectiveness of proposed key components in our S^{2}PT model, we conduct a series of ablation studies on S3DIS (Armeni et al., 2016) dataset. These results are presented in Table 3. We start with a naive linear attention mechanism without focus mechanism and any positional encoding.

Method	Focus	RoPE	bCPE	mIoU (%)
Base	✗	✗	✗	59.76
+ Focus	✓	✗	✗	60.11
+ RoPE	✓	✓	✗	60.77
+ bCPE	✓	✗	✓	68.91
S^{2}PT (Ours)	✓	✓	✓	**70.20**

Table 3. Ablation study on S3DIS.

We first observe that incorporating the focus mechanism as in FLatten Transformer (Han et al., 2023) and FlatFormer (Liu et al., 2023) provides a modest but consistent improvement of +0.35% mIoU. We then treat this stronger, focused linear attention model as the new baseline for evaluating the S^{2}PE module.

Our first key finding concerns the importance of 3D geometric encoding. Our proposed bCPE module, as a lightweight version of xCPE deployed by PTv3 (Wu et al., 2024), alone yields a substantial improvement of 8.80% mIoU. This confirms that even with a more efficient architecture, the principle of re-injecting local 3D geometry by sparse space convolution remains crucial in compensating for information altered during serialization.

Furthermore, the results validate the complementary nature of our S^{2}PE. While RoPE (1D sequential encoding) (Su et al., 2024) on its own yields a +0.66% mIoU gain, combining it with bCPE lifts the total improvement to 10.09% mIoU. This super-additive improvement demonstrates a clear synergistic effect, proving

our initial hypothesis that encoding both 3D local geometry and 1D sequential order is more effective than either approach in isolation.

SFC Order	mIoU (%)
Z	69.49
Z-trans	69.56
Hilbert	69.44
Hilbert-trans	68.96

Table 4. Effect of Space-Filling Curves (SFC) on S3DIS.

We further investigate the model's sensitivity to different Space-Filling Curves during inference, as shown in Table 4. While different curves (Z-order vs. Hilbert) inherently prioritize distinct spatial localities, the performance gap remains marginal ($< 0.6\%$ mIoU). This demonstrates that our S^2PE captures robust geometric features regardless of the specific 1D unraveling path.

5. Conclusion

In this work, we presented the **Spatio-Sequential Point Transformer (S^2PT)**, a novel hierarchical architecture that successfully bridges the gap between global receptive fields and linear computational complexity for massive 3D point clouds. To the best of our knowledge, S^2PT is the first framework to adapt focused linear attention to large-scale 3D scene understanding. By synergizing this efficient attention mechanism with our proposed **S^2PE**, we fundamentally addressed the quadratic bottleneck inherent to standard transformers. Extensive experiments demonstrate that S^2PT achieves highly competitive accuracy while unlocking unprecedented efficiency, offering a $2.58\times$ speedup in training throughput and significantly lower inference latency compared to state-of-the-art methods.

Looking ahead, the ability of S^2PT to efficiently model massive point clouds can be extended beyond discriminative tasks. Specifically, we aim to leverage this linear-complexity architecture as a foundational backbone for large-scale 3D generative modeling, with a primary focus on **3D Diffusion Transformers (DiT)**. In parallel, to continually refine the robustness of this backbone, future work will also explore adaptive fusion mechanisms for S^2PE and investigate cross-dataset generalization paradigms, ensuring S^2PT serves as a versatile and scalable tool for diverse 3D vision challenges.

References

Armeni, I., Sener, O., Zamir, A. R., Jiang, H., Brilakis, I., Fischer, M., Savarese, S., 2016. 3D Semantic Parsing of Large-Scale Indoor Spaces. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 1534–1543.

Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J. D., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., Agarwal, S., Herbert-Voss, A., Krueger, G., Henighan, T., Child, R., Ramesh, A., Ziegler, D., Wu, J., Winter, C., Hesse, C., Chen, M., Sigler, E., Litwin, M., Gray, S., Chess, B., Clark, J., Berner, C., McCandlish, S., Radford, A., Sutskever, I., Amodei, D., 2020. Language Models are Few-Shot Learners. *Advances in Neural Information Processing Systems 33 (NeurIPS)*, 33, 1877–1901.

Carion, N., Massa, F., Synnaeve, G., Usunier, N., Kirillov, A., Zagoruyko, S., 2020. End-to-End Object Detection with Transformers. *Computer Vision – ECCV 2020*, 213–229.

Chen, Y., Liu, J., Zhang, X., Qi, X., Jia, J., 2023. VoxelNeXt: Fully Sparse VoxelNet for 3D Object Detection and Tracking. *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 21674–21683.

Choromański, K. M., Likhosherstov, V., Dohan, D., Song, X., Gane, A., Sarlos, T., Hawkins, P., Davis, J. Q., Mohiuddin, A., Kaiser, L., Belanger, D. B., Colwell, L. J., Weller, A., 2021. Rethinking Attention with Performers. *9th International Conference on Learning Representations (ICLR)*.

Choy, C., Gwak, J., Savarese, S., 2019. 4D Spatio-Temporal ConvNets: Minkowski Convolutional Neural Networks. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 3070–3079.

Cui, M., Long, J., Feng, M., Li, B., Kai, H., 2023. OctFormer: Efficient Octree-Based Transformer for Point Cloud Compression with Local Enhancement. *Proceedings of the AAAI Conference on Artificial Intelligence*, 37, 470–478.

Devlin, J., Chang, M.-W., Lee, K., Toutanova, K., 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (NAACL)*, 1, 4171–4186.

Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., Uszkoreit, J., Houlsby, N., 2021. An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale. *International Conference on Learning Representations*.

Fan, Q., Huang, H., Chen, M., Liu, H., He, R., 2024. RMT: Retentive Networks Meet Vision Transformers. *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 5641–5651.

Guo, M.-H., Cai, J.-X., Liu, Z.-N., Mu, T.-J., Martin, R. R., Hu, S.-M., 2021a. PCT: Point cloud transformer. *Computational Visual Media*, 7(2), 187-199.

Guo, Y., Wang, H., Hu, Q., Liu, H., Liu, L., Bennamoun, M., 2021b. Deep Learning for 3D Point Clouds: A Survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 43(12), 4338-4364.

Han, D., Pan, X., Han, Y., Song, S., Huang, G., 2023. FLatten Transformer: Vision Transformer using Focused Linear Attention. *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, 5938–5948.

Hua, W., Dai, Z., Liu, H., Le, Q., 2022. Transformer Quality in Linear Time. *Proceedings of the 39th International Conference on Machine Learning (ICML)*, 162, 9099–9117.

Katharopoulos, A., Vyas, A., Pappas, N., Fleuret, F., 2020. Transformers are RNNs: Fast Autoregressive Transformers with Linear Attention. *Proceedings of the 37th International Conference on Machine Learning (ICML)*, 119, 5156–5165.

Lahoud, J., Cao, J., Khan, F. S., Cholakkal, H., Anwer, R. M., Khan, S., Yang, M.-H., 2022. 3D Vision with Transformers: A Survey.

Lai, X., Liu, J., Jiang, L., Wang, L., Zhao, H., Liu, S., Qi, X., Jia, J., 2022. Stratified Transformer for 3D Point Cloud Segmentation. *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 8490–8499.

Lang, A. H., Vora, S., Caesar, H., Zhou, L., Yang, J., Beijbom, O., 2019. PointPillars: Fast Encoders for Object Detection From Point Clouds. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 12689–12697.

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. *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 10012–10022.

Liu, Z., Yang, X., Tang, H., Yang, S., Han, S., 2023. FlatFormer: Flattened Window Attention for Efficient Point Cloud Transformer. *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 1200–1211.

Maturana, D., Scherer, S., 2015. VoxNet: A 3D Convolutional Neural Network for real-time object recognition. *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 922–928.

Oehmcke, S., Li, L., Revenga, J. C., Nord-Larsen, T., Trepekli, K., Gieseke, F., Igel, C., 2022. Deep learning based 3D point cloud regression for estimating forest biomass. *Proceedings of the 30th International Conference on Advances in Geographic Information Systems*.

Qi, C., Su, H., Kaichun, M., Guibas, L. J., 2017a. PointNet: Deep Learning on Point Sets for 3D Classification and Segmentation. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 77–85.

Qi, C., Yi, L., Su, H., Guibas, L. J., 2017b. PointNet++: Deep Hierarchical Feature Learning on Point Sets in a Metric Space. *Advances in Neural Information Processing Systems 30 (NeurIPS)*, 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. *Advances in Neural Information Processing Systems 35 (NeurIPS)*, 35, 23192–23204.

Qiu, B., Zhou, Y., Dai, L., Wang, B., Li, J., Dong, Z., Wen, C., Ma, Z., Yang, B., 2024. WHU-Railway3D: A Diverse Dataset and Benchmark for Railway Point Cloud Semantic Segmentation. *IEEE Transactions on Intelligent Transportation Systems*, 25(12), 20900-20916.

Shen, S., Wu, Y., Zhang, H., Lu, J., Fan, H., 2025. A simplification method for large-scale urban point clouds considering diversity of terrain object features. *International Journal of Digital Earth*, 18(1), 2505993. https://doi.org/10.1080/17538947.2025.2505993.

Su, J., Ahmed, M., Lu, Y., Pan, S., Bo, W., Liu, Y., 2024. RoFormer: Enhanced transformer with Rotary Position Embedding. *Neurocomput.*, 568(C). https://doi.org/10.1016/j.neucom.2023.127063.

Sun, Y., Dong, L., Huang, S., Ma, S., Xia, Y., Xue, J., Wang, J., Wei, F., 2023. Retentive Network: A Successor to Transformer for Large Language Models.

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. *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 797–806.

Thengane, V., Zhu, X., Bouzerdoum, S., Phung, S. L., Li, Y., 2025. Foundational Models for 3D Point Clouds: A Survey and Outlook.

Thomas, H., Qi, C. R., Deschaud, J.-E., Marcotegui, B., Goulette, F., Guibas, L., 2019. KPConv: Flexible and Deformable Convolution for Point Clouds. *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, 6410–6419.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., Polosukhin, I., 2017. Attention is all you need. *Advances in Neural Information Processing Systems 30 (NeurIPS)*, 30.

Wang, S., Li, B. Z., Khabsa, M., Fang, H., Ma, H., 2020. Linformer: Self-Attention with Linear Complexity.

Wu, X., Jiang, L., Wang, P.-S., Liu, Z., Liu, X., Qiao, Y., Ouyang, W., He, T., Zhao, H., 2024. Point Transformer V3: Simpler, Faster, Stronger. *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 4840–4851.

Wu, X., Lao, Y., Jiang, L., Liu, X., Zhao, H., 2022. Point Transformer V2: Grouped Vector Attention and Partition-based Pooling. *Advances in Neural Information Processing Systems 35 (NeurIPS)*, 35, 33330–33342.

Yang, Y.-Q., Guo, Y.-X., Liu, Y., 2025. Swin3D++: Effective Multi-Source Pretraining for 3D Indoor Scene Understanding. *Computational Visual Media*, 11(3), 465-481.

Yu, X., Tang, L., Rao, Y., Huang, T., Zhou, J., Lu, J., 2022. Point-BERT: Pre-training 3D Point Cloud Transformers with Masked Point Modeling. *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 19291–19300.

Zhai, S., Talbott, W., Srivastava, N., Huang, C., Goh, H., Zhang, R., Susskind, J., 2021. An Attention Free Transformer.

Zhao, H., Jiang, L., Jia, J., Torr, P., Koltun, V., 2021. Point Transformer. *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, 16239–16248.

Zhou, Y., Tuzel, O., 2018. VoxelNet: End-to-End Learning for Point Cloud Based 3D Object Detection. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 4490–4499.