Mech. Sci., 17, 221–231, 2026
https://doi.org/10.5194/ms-17-221-2026

Mechanical Sciences
Open Access

GAPS: Group-wise Affine-Perturbed Serialization for efficient 3D semantic segmentation

Yubin Tang, Yangchen Liu, and Zichuan Fan

College of Computer and Information Science, Southwest University Beibei, Chongqing 400030, China

Correspondence: Zichuan Fan (fanzc@swu.edu.cn)

Received: 12 January 2026 – Revised: 17 February 2026 – Accepted: 24 February 2026 – Published: 11 March 2026

Abstract. Space-filling curves (SFCs) have enabled transformers to process massive 3D point clouds with linear complexity by mapping them into 1D sequences. However, standard SFCs rely on fixed, axis-aligned traversal orders, which introduce systematic axial anisotropy and frequently sever the connectivity of oblique geometric structures – a phenomenon we term "locality breaches". While existing methods attempt to mitigate this by sequentially stacking multiple patterns, they incur a prohibitive linear increase in latency. To resolve this efficiency–accuracy dilemma, we propose GAPS (Group-wise Affine-Perturbed Serialization). Mathematically, we derive a flexible mechanism to generate diverse, bijective scanning paths via affine transformations over the Galois field $\mathbb{F}_2$. This allows for the synthesis of "sheared" and "rotated" traversals that effectively establish neighborhood connectivity along non-axial geometries. Architecturally, GAPS employs a group-wise parallel design that splits feature channels to aggregate multiple-perspective contexts simultaneously, thereby circumventing the computational penalty of serial execution. Extensive experiments on the ScanNet200 benchmark demonstrate that GAPS achieves 36.8 % mean Intersection over Union (mIoU), outperforming the strong baseline, Point Transformer V3 (PTv3), by 1.6 %. Remarkably, GAPS delivers this significant gain with zero parameter overhead and minimal latency overhead, establishing a new state-of-the-art performance among standard train-from-scratch 3D backbones.

1 Introduction

Three-dimensional point cloud semantic segmentation serves as a cornerstone for machine perception, enabling autonomous systems to interpret and interact with the physical world. It underpins critical applications in mechanical sciences and industrial robotics, providing the essential perceptual foundation for autonomous driving mechanisms (Hu et al., 2023), safe human–robot collaboration in complex manufacturing environments (Mohammadi Amin et al., 2025), and the spatial navigation of embodied AI (Hughes et al., 2022). By enabling machines to interpret unstructured environments reliably, robust 3D segmentation directly facilitates advanced mechanical automation and safe operational control.

Unlike 2D images characterized by regular grid structures, 3D point clouds are inherently sparse, unstructured, and irregular (Guo et al., 2021), posing significant challenges for deploying efficient deep learning architectures in real-time applications (Liu et al., 2023). While early approaches attempted to tackle this via voxelization and sparse convolutions (Choy et al., 2019) or computationally expensive neighbor queries (Qi et al., 2017; Thomas et al., 2019), window-based transformers (Vaswani et al., 2017; Wang, 2023; Wu et al., 2024) have recently emerged as the dominant paradigm. By serializing 3D points into 1D sequences using space-filling curves (SFCs), these methods effectively reduce the quadratic complexity of self-attention to linear time $O(N)$. Notably, beyond their foundational role in data compression (Chen et al., 2022), SFCs are critically utilized in 3D robotic coverage path planning and autonomous UAV navigation (Shi et al., 2025; Mazaheri et al., 2024). This cross-disciplinary utility offers a compelling trade-off between perception accuracy and the low system latency critical for real-time mechanical operations.

Despite their success, a fundamental bottleneck persists in current window-based architectures: the reliance on fixed, axis-aligned SFCs introduces systematic axial anisotropy. Standard serialization methods impose a rigid, grid-like

traversal order on irregular 3D structures, defining locality strictly based on axis-aligned boundaries rather than intrinsic geometric proximity. Consequently, points that are geographically connected but spatially oblique – such as a diagonal wall or a rotated chair – are often severed by grid partitions and mapped to distant positions in the sequence. This phenomenon, which we term "locality breaches", leads to semantic inconsistency and severely limits the model's ability to distinguish fine-grained categories that rely on subtle, non-axial geometric cues.

To address these limitations, we propose the Group-wise Affine-Perturbed Serialization (GAPS) framework, a novel paradigm that fundamentally mitigates grid bias while maintaining high computational efficiency. Instead of relying on limited sets of predefined patterns, GAPS introduces a mathematically grounded mechanism to generate diverse, non-axial scanning paths via affine transformations over the finite field $\mathbb{F}_2$, effectively simulating multi-view scanning orders. We implement this through a group-wise parallel architecture that decouples feature channels to process multiple geometric views simultaneously. By imposing block lower-triangular constraints on the transformation matrices, we ensure that these diverse serializations are globally bijective and preserve hierarchical locality, effectively allowing the network to "heal" locality breaches by aggregating context across diverse orientations.

In summary, the main contributions of this paper are as follows:

- We identify axial anisotropy as a critical flaw in existing window-based transformers and propose a novel serialization paradigm based on affine perturbations over $\mathbb{F}_2$ to mitigate this grid bias.
- We introduce the GAPS block, a plug-and-play module utilizing a group-wise parallel architecture. This design achieves multi-view geometric modeling with zero parameter overhead and comparable latency to sequential baselines.
- Extensive experiments demonstrate the superiority of our framework. On the challenging ScanNet200 (Rozenberszki et al., 2022) dataset, GAPS achieves 36.8 % mean Intersection over Union (mIoU), outperforming the strong baseline PTv3 (Wu et al., 2024) by a significant margin of +1.6 %.

2 Related work

2.1 Deep learning on point clouds

Early approaches generally fall into two categories. Point-based architectures (Charles et al., 2017; Thomas et al., 2019) preserve geometric fidelity by processing raw coordinates but suffer from high memory costs and $O(N \log N)$ complexity. Conversely, voxel-based methods (Choy et al., 2019; Peng et al., 2024) leverage sparse convolutions for efficiency but introduce quantization artifacts. Recently, transformers have emerged to balance the precision of point-based methods with the efficiency of grid-based processing.

2.2 Serialization-based point cloud learning

The paradigm for processing 3D point clouds is shifting from expensive neighborhood queries (Qi et al., 2017) towards efficient serialization-based modeling. This approach underpins both window-based transformers (Wang, 2023; Wu et al., 2024) and emerging state space models (Li et al., 2025a, b). Fundamentally, these methods rely on mapping 3D coordinates into 1D sequences via space-filling curves (SFCs).

However, current scanning strategies remain overly rigid across architectures. Window-based transformers like OctFormer (Wang, 2023) rely on the Z-order curve, while PTv3 (Wu et al., 2024) alternates between Z-order and Hilbert curves. Similarly, recent state space model (SSM) adaptations (Li et al., 2025a, b) are still confined to combining fixed, axis-aligned patterns. Despite these efforts, both families of models suffer from significant axial anisotropy and "locality breaches" when encountering oblique structures. In contrast, GAPS introduces a mathematically grounded framework via affine perturbations over $\mathbb{F}_2$, enabling the generation of diverse, non-axial scanning paths that decouple geometric proximity from grid alignment.

2.3 Supervision strategies and backbone robustness

To further push the performance boundaries, existing state-of-the-art methods often resort to complex auxiliary mechanisms. For instance, BFANet (Zhao et al., 2025) utilizes explicit boundary-aware losses, while ODIN (Jain et al., 2024) and DITR (Zeid et al., 2026) rely on cross-modal distillation from 2D foundation models. While effective, these strategies increase engineering complexity and training overhead. GAPS, conversely, is designed as a robust, standalone 3D backbone. We demonstrate that superior performance on challenging benchmarks such as ScanNet200 can be achieved solely through principled geometric modeling, without the need for external data priors or complex auxiliary supervision.

2.4 Applications in mechanical sciences

Beyond core architectural developments, 3D point cloud perception serves as a fundamental enabler for advanced mechanical systems. In the context of industrial automation, real-time semantic segmentation is critical for robust trajectory prediction, dynamic obstacle avoidance, and safe human–robot collaboration (Hu et al., 2023; Xu et al., 2025; Mohammadi Amin et al., 2025). Furthermore, simultaneous localization and mapping (SLAM) frameworks increasingly

rely on accurate 3D semantic priors to operate safely within dynamic mechanical environments (Chen et al., 2024). Interestingly, the mathematical constructs underlying our serialization framework, such as space-filling curves (SFCs), possess deep roots in mechanical engineering and robotics. Due to their rigorous locality-preserving properties, SFCs like the Hilbert curve are recognized as standard optimization algorithms for exhaustive 3D coverage path planning in autonomous UAV networks and complex mobile mechanisms (Shi et al., 2025; Mazaheri et al., 2024). By improving the geometric isotropy and computational efficiency of 3D segmentation backbones via affine-perturbed SFCs, our GAPS framework directly addresses the stringent latency and spatial accuracy requirements of these downstream mechanical applications.

3 Methodology

In this section, we elaborate on the proposed Group-wise Affine-Perturbed Serialization (GAPS) framework. We first present the overall architecture, followed by the mathematical formulation of affine-perturbed SFCs and, finally, the design of the GAPS block for efficient multi-view geometric modeling.

3.1 Overall architecture

As illustrated in Fig. 1, our backbone adheres to the standard hierarchical encoder–decoder design established by PTv3 (Wu et al., 2024). The network processes point clouds through four downsampling stages. The critical deviation from prior art lies in the feature extraction units: instead of stacking sequential blocks with fixed patterns, we deploy our GAPS blocks. This design leverages a multi-sequence feature fusion (MSFF) strategy to enable simultaneous multi-path context aggregation within a single layer, thereby enriching geometric coverage and mitigating scanning anisotropy without the latency overhead of deep serial stacking.

3.2 Affine-perturbed SFC generation over $\mathbb{F}_2$

Standard space-filling curves (SFCs), such as Morton (Z-order) codes, map 3D coordinates to 1D indices to facilitate efficient window-based attention. However, relying on a single fixed pattern introduces axial anisotropy, leading to "locality breaches" in which spatially adjacent points become separated in the sequence. To address this, we introduce a mathematical framework for efficiently generating diverse SFCs.

3.2.1 Vector space formulation

Coordinates are normalized to $[0, 1]^3$ and quantized to 2^L levels. We interleave the bits of these coordinates from most significant to least significant (i.e., $x_L y_L z_L \ldots x_1 y_1 z_1$) to form a unified binary vector $\boldsymbol{v}_i \in \mathbb{F}_2^D$, where $D = 3L$.

We define a generalized serialization function Φ: $\mathbb{F}_2^D \to \mathbb{F}_2^D$ as an affine transformation over the finite field:

$$\Phi(\boldsymbol{v}_i; \mathbf{A}, \boldsymbol{b}) = \mathbf{A}\boldsymbol{v}_i \oplus \boldsymbol{b}, \tag{1}$$

where $\mathbf{A} \in \mathbb{F}_2^{D \times D}$ is a binary transformation matrix, $\boldsymbol{b} \in \mathbb{F}_2^D$ is a translation vector, and $\oplus$ denotes vector addition modulo 2 (bitwise exclusive OR (XOR)). The matrix multiplication is performed over $\mathbb{F}_2$ (bitwise and followed by XOR sum). Note that operations over $\mathbb{F}_2$ induce a discrete permutation of the Morton codes rather than a continuous Euclidean transformation.

The resulting binary vector $\Phi(\boldsymbol{v}_i)$ is then interpreted as a D-bit integer, providing the final 1D sorting index for point p_i.

Geometric interpretation

Different configurations of matrix $\mathbf{A}$ yield fundamentally different geometric traversal paths. For instance, setting $\mathbf{A}$ as the identity matrix $\mathbf{I}$ recovers the standard Morton (Z-order) curve. Permutation matrices correspond to swapping coordinate axes (e.g., prioritizing the Y axis over the Z axis). At the same time, off-diagonal elements introduce discrete fractal-like "shears", yielding oblique scanning paths that mimic the effect of realigning the traversal order with non-axis-aligned dependencies.

3.2.2 Theoretical guarantee of bijectivity

To prevent feature collisions, the serialization mapping must be bijective. Since the domain $\mathbb{F}_2^D$ is a finite vector space, the affine transformation $\Phi(\boldsymbol{v}) = \mathbf{A}\boldsymbol{v} \oplus \boldsymbol{b}$ is a bijection if and only if it is injective. Mathematically, for any distinct inputs $\boldsymbol{v}_1, \boldsymbol{v}_2$, the equality $\Phi(\boldsymbol{v}_1) = \Phi(\boldsymbol{v}_2)$ implies $\mathbf{A}(\boldsymbol{v}_1 \oplus \boldsymbol{v}_2) = \mathbf{0}$. Consequently, a unique mapping is guaranteed if and only if the null space of $\mathbf{A}$ is trivial, which requires $\det(\mathbf{A}) \equiv 1 \pmod 2$.

3.2.3 Data-independent iterative orthogonal search

While theoretically flexible, a dense matrix $\mathbf{A}$ incurs prohibitive computational costs and complex invertibility checks. To address this, we impose a block lower triangular (BLT) structure on $\mathbf{A}$. We group $\boldsymbol{v}$ into 3-bit octree-level sub-vectors, using diagonal blocks $\mathbf{B}_k \in \mathbb{F}_2^{3\times 3}$ for intra-level transformations and lower off-diagonal blocks for inter-level dependencies.

This design guarantees global invertibility ($\det(\mathbf{A}) = \prod \det(\mathbf{B}_k) = 1$) simply by ensuring each 3×3 diagonal block is non-singular, decomposing global constraints into trivial local checks. Crucially, the zero upper-triangular blocks maintain level consistency: coarse-level output coordinates depend exclusively on equal or coarser inputs. This preserves

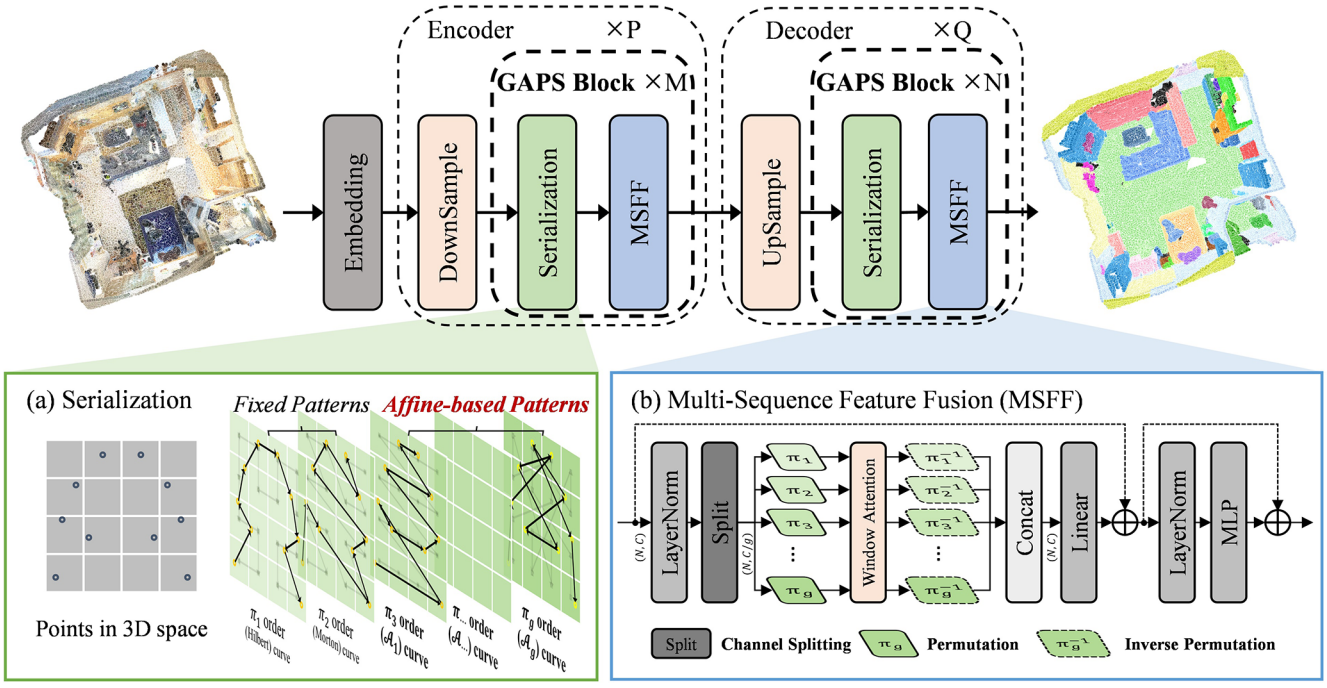

Figure 1. Overview of the proposed architecture. The framework processes raw 3D point clouds through a hierarchical U-Net structure. The core processing unit is the GAPS block, which integrates two synergistic components: **(a)** serialization, which generates diverse scanning patterns, and **(b)** the multi-sequence feature fusion (MSFF) module. The MSFF module splits features into groups, applies serialization orders (π_g) to perform window attention in parallel, and aggregates multi-view features via MLP fusion.

hierarchical locality by preventing fine-scale perturbations from disrupting global structure.

To balance geometric diversity with stability, we employ a hybrid strategy within the GAPS block. Instead of random selection, we utilize a data-independent iterative orthogonal search to generate the affine matrices. We define the diversity metric between two matrices as the matrix Hamming distance: $D_{\text{mat}}(\mathbf{A}, \mathbf{M}) = \text{popcount}(\mathbf{A} \oplus \mathbf{M})$, where popcount$(\cdot)$ denotes the number of set bits (1 s) in the binary representation. To generate a set of diverse patterns $\mathcal{S} = \{\mathbf{A}_1, \ldots, \mathbf{A}_n\}$, we maximize the minimum distance to existing patterns:

$$\mathbf{A}_k = \arg\max_{\mathbf{A}} \left[\min_{\mathbf{M} \in \{\mathbf{I}, \mathbf{A}_1, \ldots, \mathbf{A}_{k-1}\}} D_{\text{mat}}(\mathbf{A}, \mathbf{M}) \right]. \tag{2}$$

This is subject to the BLT constraint and $\mathbf{B}_k \in GL(3, \mathbb{F}_2) \setminus \{\mathbf{I}\}$. These matrices are pre-computed and frozen, ensuring zero runtime overhead. The specific 3×3 seed matrices used in our implementation are detailed in Appendix A.

3.3 Group-wise Parallel Serialization (GAPS) block

The core innovation of our architecture is the GAPS block (Fig. 1). It comprises two synergistic components: a serialization mechanism that defines diverse scanning strategies (Fig. 1a) and a multi-sequence feature fusion (MSFF) module that executes parallel feature processing (Fig. 1b).

The MSFF module decouples feature channels to facilitate the parallel application of diverse scanning patterns. Given an input feature map $\mathbf{F} \in \mathbb{R}^{N \times C}$, the processing pipeline proceeds as follows.

3.3.1 Adaptive channel splitting and parallel serialization

We first apply LayerNorm. To prevent feature collapse in shallow stages where the channel dimension C is limited, we introduce a channel protection mechanism. We define a minimum sub-group dimension $C_{\min}$ (e.g., 16), and the effective number of parallel groups G_{eff} is determined by

$$G_{\text{eff}} = \min\left(G_{\text{target}}, \lfloor C / C_{\min} \rfloor\right), \tag{3}$$

where G_{target} is the maximum desired diversity (e.g., 8). In deep stages ($C \geq G_{\text{target}} \times C_{\min}$), all G_{target} patterns activate simultaneously. In shallow stages where $G_{\text{eff}} < G_{\text{target}}$, we distribute the serialization patterns cyclically across consecutive blocks to ensure geometric coverage.

After splitting $\mathbf{F}$ into G_{eff} groups, let $\mathbf{F}_g \in \mathbb{R}^{N \times (C/G_{\text{eff}})}$ denote the feature subset for the gth group. For each group, we assign a unique affine matrix $\mathbf{A}_g$ and pre-compute the corresponding permutation indices $\mathcal{I}_g$. We then execute a parallel gather operation:

$$\hat{\mathbf{F}}_g = \text{Gather}\left(\mathbf{F}_g, \mathcal{I}_g\right). \tag{4}$$

While this reordering is a memory-bound operation, it ensures that, in group g, points defined as neighbors by the traversal $\Phi(\cdot; \mathbf{A}_g)$ become adjacent in the 1D sequence.

3.3.2 Multi-view window attention

The reordered 1D sequences are partitioned into non-overlapping windows of size W. We perform efficient windowed self-attention (WSA) on each group in parallel. To maintain parameter efficiency, the linear projections for Q, K, and V are implemented by partitioning a single global weight matrix $\mathbf{W} \in \mathbb{R}^{C\times C}$ into G sub-blocks. This ensures that each group g operates on its own feature subspace without introducing additional parameters beyond a standard attention layer. The attention output $\mathbf{Y}_g$ is computed as follows:

$$\mathbf{Y}_g = \text{Attention}\left(\mathbf{Q}_g, \mathbf{K}_g, \mathbf{V}_g\right) = \text{Softmax}\left(\frac{\mathbf{Q}_g \mathbf{K}_g^{\mathrm{T}}}{\sqrt{d_k}}\right)\mathbf{V}_g. \quad (5)$$

This allows the mechanism to aggregate context along the specific non-axial path induced by $\mathbf{A}_g$.

3.3.3 Inverse mapping and fusion

Post-attention, features are restored to the canonical order via an inverse scatter operation: $\mathbf{Z}_g = \text{Scatter}(\mathbf{Y}_g, \mathcal{I}_g^{-1})$. This ensures global consistency across all groups. Finally, the sub-features are concatenated and fused via a linear projection and an multi-layer perceptron (MLP) with residual connections. This MSFF design achieves multi-view modeling with zero parameter overhead and minimal latency overhead compared to sequential stacking.

4 Experiments

This section presents a comprehensive evaluation of the proposed GAPS framework. We begin by outlining the experimental setup, including the datasets and implementation details. Subsequently, we compare our method against state-of-the-art (SOTA) approaches based on three standard benchmarks. Finally, we conduct extensive ablation studies to empirically validate the effectiveness of the affine-perturbed serialization and the group-wise parallel architecture, followed by an analysis of computational efficiency.

4.1 Experimental setup

4.1.1 Datasets

ScanNet v2 (Dai et al., 2017) contains 1513 indoor scans annotated with 20 categories. Following standard protocols, we use 1201, 312, and 100 scenes for training, validation, and testing. ScanNet200 (Rozenberszki et al., 2022) expands the label space to 200 classes with a long-tailed distribution. Its numerous fine-grained categories with high geometric similarity provide-a rigorous test of our model's local context modeling capabilities. S3DIS (Armeni et al., 2016a) covers six large indoor areas. We strictly adhere to the standard Area-5 cross-validation protocol (training on Areas 1–4 and 6 and testing on Area 5) to evaluate generalization.

4.1.2 Implementation details

Implemented on Pointcept with PyTorch, we strictly follow the training protocols, augmentation, and optimization settings of the baseline PTv3. Specifically, to ensure a rigorously fair comparison, all models use the same setup: 800 training epochs, an AdamW optimizer with a base learning rate of 5×10^{-3}, a batch size of 12, a window size of 1024, and identical data sampling strategies. All models are trained on four NVIDIA RTX 4090 GPUs. We adopt a standard four-stage encoder–decoder. To balance diversity and feature expressiveness, we set the target group number $G_{\text{target}} = 8$ and the minimum channel dimension $C_{\min} = 16$. Consequently, for the first stage ($C = 32$), the model adaptively adjusts to $G_{\text{eff}} = 2$ parallel groups per block, cycling through different scanning patterns across blocks to maintain geometric diversity. In deeper stages ($C \geq 128$), the full $G = 8$ parallel serialization is activated within each block. To ensure robustness, we employ a complementary serialization strategy combining standard anchors (Z-order, Hilbert) with orthogonal affine-perturbed curves selected to cover oblique geometric dependencies. Voxel sizes are set to 0.02 m for ScanNet and 0.04 m for S3DIS.

4.2 Main results

This subsection benchmarks the proposed GAPS framework against representative state-of-the-art methods. Table 1 summarizes the quantitative results based on ScanNet v2, ScanNet200, and S3DIS Area 5. To ensure a structured analysis, we categorize baselines into four groups: (1) standard 3D backbones (pure geometric supervision), (2) methods with auxiliary supervision (e.g., boundary priors), (3) methods with extra data/distillation, and (4) multi-modal methods.

As shown in Table 1, our method demonstrates competitive and often superior performance compared to the strong baseline PTv3 and other established 3D backbones. Most notably, GAPS achieves 36.8 % mIoU based on the challenging ScanNet200, outperforming the direct baseline PTv3 (35.2 %) by a substantial margin of +1.6 %. This improvement is particularly significant given the fine-grained nature of the dataset, which contains numerous categories with high geometric similarity. The result strongly validates our hypothesis that introducing diverse, affine-perturbed scanning paths enables the effective capture of subtle local geometric contexts required to distinguish ambiguous classes, which are often conflated by fixed-order serialization. Notably, GAPS effectively narrows the gap towards methods

Table 1. Quantitative comparison on semantic segmentation benchmarks. We report the mean IoU (mIoU, %) based on the validation sets. To ensure a fair comparison, we distinguish between standard backbones and methods that use auxiliary boundary supervision (e.g., BFANet) or external data (e.g., ARKit LabelMaker). Bold font indicates the best result within the standard backbone category.

Method	ScanNet v2	ScanNet200	S3DIS Area 5
Standard 3D backbones (train-from-scratch)			
PointNet++ (Charles et al., 2017)	53.5	–	–
KPConv (Thomas et al., 2019)	69.2	–	67.1
MinkUNet (Choy et al., 2019)	72.2	25.0	65.4
PTv2 (Wu et al., 2022)	75.4	30.2	71.6
StratifiedFormer (Lai et al., 2022)	74.3	–	72.0
Swin3D (Yang et al., 2025)	76.4	–	72.5
OctFormer (Wang, 2023)	75.7	32.6	–
PTv3 (Wu et al., 2024) (Baseline)	77.5	35.2	**73.4**
Ours (GAPS)	**77.8**	**36.8**	73.3
Methods with auxiliary supervision (e.g., boundary loss)			
BFANet (Zhao et al., 2025)	78.0	37.3	–
Methods with extra data/distillation			
ARKit LabelMaker (Ji et al., 2025)	79.1	37.5	–
D-DITR (v2) (Zeid et al., 2026)	79.2	37.7	75.0
Multi-modal (2D+3D) methods			
ODIN (Jain et al., 2024)	77.8	40.5	68.6
DITR (Zeid et al., 2026)	80.5	41.2	74.1

that use heavy auxiliary supervision or extra data without incurring the engineering complexity.

Regarding general scene understanding based on ScanNet v2, GAPS achieves 77.8 % mIoU, surpassing classic sparse convolution networks and recent transformer architectures such as Swin3D (76.4 %) and OctFormer (75.7 %). While BFANet achieves a slightly higher score (78.0 %), it relies on an explicit boundary detection module and specific losses; in contrast, GAPS offers a more generalizable solution through intrinsic backbone improvements. Finally, based on S3DIS Area 5, our method yields 73.3 % mIoU. While marginally lower than PTv3 (73.4 %), this difference (−0.1 %) is negligible. We hypothesize that S3DIS, dominated by large planar structures (walls, floors, ceilings) and simple layouts, benefits less from the complex multi-view serialization designed for intricate geometries as the standard Z-order is already near-optimal for such environments. This suggests that future work could adaptively utilize local geometric entropy to control multi-view serialization in planar regions.

4.3 Efficiency analysis

A common limitation of multi-view methods is the substantial increase in computational cost and latency. In this subsection, we demonstrate that GAPS achieves a superior trade-off between segmentation accuracy and system efficiency. Measurements are conducted on the ScanNet validation set using a single NVIDIA RTX 4090 GPU with a batch size of 1.

As summarized in Table 2, GAPS maintains an identical parameter count (46.2 M parameters) to the baseline PTv3. This parameter invariance is guaranteed by our architectural design: the group-wise linear projections in the MSFF module are mathematically equivalent to the baseline's dense projections, merely partitioned across channels to facilitate parallel multi-view serialization. This confirms that the performance gains of GAPS stem from geometric modeling improvements rather than increased model capacity. This is because the affine transformation matrices are pre-computed constants that require no gradient updates. Regarding execution speed, we observe a marginal increase in latency (+4 ms for inference). This overhead is primarily attributed to the memory-bound gather and scatter operations required for multi-view reordering, which incur random memory access costs. Future engineering efforts could further mitigate this bottleneck by implementing hardware-level optimizations, such as custom CUDA kernels that fuse memory reordering operations with the initial attention projections. However, this impact is minimized by our group-wise parallel design. By performing computationally intensive window attention on channel split groups, GAPS theoretically maintains a total floating-point operation count (FLOP) equivalent to that of the single-view baseline, thereby avoiding the latency explosion common in deep, sequentially stacked multi-path architectures.

Compared to other state-of-the-art methods, GAPS demonstrates a substantial speed advantage. It is approximately 6.8× faster than Swin3D (67 ms vs. 460 ms) and outperforms both OctFormer and PTv2 in terms of speed and memory efficiency. By accepting a minimal 4 ms latency cost for a +1.6 % mIoU gain on ScanNet200, GAPS achieves an

Table 2. Efficiency comparison. We report model size (Params), training and inference latency (per scene), and peak GPU memory usage. Compared to heavy transformer architectures (e.g., Swin3D) and previous sparse baselines, our method maintains real-time capability (∼ 15 FPS) while delivering superior performance. Bold values indicate the best (lowest) results in each column.

Method	Params (M)	Training		Inference	
		Latency (ms)	Memory (GB)	Latency (ms)	Memory (GB)
MinkUNet (Choy et al., 2019)	37.9	270	5.0	91	4.8
OctFormer (Wang, 2023)	44.0	268	13.1	88	12.6
Swin3D (Yang et al., 2025)	71.1	608	13.8	460	8.9
PTv2 (Wu et al., 2022)	**12.8**	315	13.6	195	18.3
PTv3 (Wu et al., 2024) (baseline)	46.2	**155**	**7.0**	**63**	**5.3**
Ours (GAPS)	46.2	162	7.4	67	5.4

inference speed of ∼ 15 FPS, validating its suitability as a practical backbone for real-time applications.

4.4 Ablation studies

We conduct ablation studies on the ScanNet200 validation set to verify our architectural design. ScanNet200 is selected as the primary test bed due to its high sensitivity to fine-grained geometric modeling.

4.4.1 Effectiveness of serialization strategies

A core premise of our method is that diverse, non-axial scanning paths mitigate the locality breaches inherent in standard space-filling curves (SFCs). Table 3 disentangles the performance gains stemming from the parallel architecture versus the curve generation strategy. Note the structural implementation differences: the standard PTv3 backbone uses a fixed number of sequential blocks per stage (specifically [2, 2, 6, 2]). In sequential experiments (rows 1–4), distinct curves must be assigned to different blocks. Consequently, stages with few blocks (e.g., stages 1, 2, and 4) cannot utilize all four curve types simultaneously; the model is forced to select a subset or alternate them. In contrast, our parallel architecture (rows 5–7) overcomes this by splitting channels within a single block, allowing all specified curves to be active simultaneously at every stage.

Impact of adaptive grouping

Our channel protection mechanism ($C_{\min} = 16$) creates distinct behavior across stages. In stage 1 ($C = 32$), GAPS operates with $G_{\text{eff}} = 2$, processing only two diverse views per block – functionally resembling the sequential baseline. However, the advantage becomes decisive at deeper stages (e.g., stage 3, $C = 128$), where GAPS fully activates eight parallel views within each block. Conversely, the sequential baseline remains structurally limited to processing a single view per block, regardless of channel width, and fails to exploit the high-dimensional feature space fully.

Rows 1–4 demonstrate that increasing fixed patterns (Z-order and Hilbert) within the sequential PTv3 baseline improves performance from 32.6 % to 35.2 %. This confirms that single-view serialization is a bottleneck; introducing diverse scanning orders recovers lost spatial contexts even when constrained by sequential blocks. When utilizing the identical set of standard curves ($\mathcal{Z} + \mathcal{H}$), our parallel GAPS architecture (row 5) outperforms the sequential PTv3 (row 4) by 0.6 % without increasing network depth or latency. While the number of blocks limits the sequential baseline, our GAPS block-splitting feature enables parallel processing of multiple views. This allows the network to integrate multi-perspective geometric information simultaneously within every block, maximizing curve utility regardless of stage depth, validating the efficiency of our multi-sequence feature fusion (MSFF) module.

The integration of our proposed affine-perturbed curves yields the most significant boost, improving mIoU to 36.8 % (row 7). This supports our claim in Sect. 3.2: while standard curves are axis-aligned, our affine transformations introduce "sheared" scanning paths. These non-axial traversals optimize the memory layout for oblique geometric structures that standard grid-aligned traversals would otherwise fragment.

4.4.2 Impact of group number (G)

We investigate the trade-off between serialization diversity and feature capacity by varying the group number G. Increasing G from 1 to 8 yields steady gains (32.6 % to 36.8 % mIoU), confirming that decoupling features allows the model to leverage diverse scanning patterns to reduce geometric blind spots. However, performance degrades when $G > 8$ (e.g., 35.9 % at $G = 16$). We attribute this to *attention context fragmentation*. Despite sufficient channels per group, excessive splitting isolates attention computation into too many specialized sequences, hindering effective integration by the final fusion layer. Consequently, we adopt $G = 8$ as the optimal configuration.

Table 3. Ablation study on serialization strategies and architecture. We investigate the impact of increasing curve diversity and compare the sequential stacking strategy (PTv3 style) vs. our parallel group-wise fusion (GAPS). Legend: $\mathcal{Z}$ – standard Z-order; $\mathcal{H}$ – Hilbert curve; $\mathcal{A}$ – proposed Affine-perturbed curves. Rotated variants are included in the sets. Bold values indicate the best performance (highest mIoU and largest improvement Δ).

ID	Architecture	Serialization set	mIoU (%)	Δ
Baseline with standard curves				
1	PTv3 (seq.)	$\mathcal{Z}$ (Single view)	32.6	–
2	PTv3 (seq.)	$\mathcal{Z} + \mathcal{Z}_{\text{rot}}$	34.3	+1.7
3	PTv3 (seq.)	$\mathcal{H} + \mathcal{H}_{\text{rot}}$	34.5	+1.9
4	PTv3 (seq.)	$\mathcal{Z} + \mathcal{H} + \mathcal{Z}_{\text{rot}} + \mathcal{H}_{\text{rot}}$	35.2	+2.6
Effect of GAPS architecture				
5	Ours (parallel)	$\mathcal{Z} + \mathcal{H} + \mathcal{Z}_{\text{rot}} + \mathcal{H}_{\text{rot}}$	35.8	+3.2
Effect of affine perturbations				
6	Ours (parallel)	$\ldots + \mathcal{A}_1 + \mathcal{A}_{1,\text{rot}}$	36.2	+3.6
7	Ours (parallel)	$\mathbf{\ldots + \mathcal{A}_1 + \ldots + \mathcal{A}_2 + \ldots}$	**36.8**	**+4.2**

4.5 Qualitative analysis

Figures 2 and 3 present qualitative comparisons of segmentation results and effective receptive fields (ERFs), respectively.

4.5.1 Segmentation consistency

Figure 2 highlights how GAPS addresses critical baseline failures. In Scene0081_02 (top), PTv3 suffers from semantic inconsistency, misclassifying a chair's base as a stool due to grid partitioning; GAPS resolves this by integrating cross-boundary context. In Scene0494_00 (middle), our multi-view approach successfully distinguishes fine-grained classes (standard vs. office chairs). Furthermore, in the dense clutter of Scene0164_00 (bottom), GAPS resists the semantic bleeding observed in PTv3, preserving instance boundaries where the baseline over-smooths.

4.5.2 Mechanism visualization

Figure 3 verifies the mechanism behind this consistency using a chair instance where the seat and base are spatially disconnected due to scanning sparsity. As shown in (b), the standard Z-order is severely restricted by the partition boundary, focusing exclusively on right-side fragments. While sequential stacking (c) marginally captures signals on the left wheels and backrest, the activation remains weak. In contrast, GAPS (d) utilizes affine perturbations to bridge both grid partitions and spatial gaps; it strongly activates the symmetric left wheels (orange box) and establishes robust long-range dependencies with the backrest, ensuring part-whole consistency despite structural discontinuities.

5 Conclusion

In this work, we address the axial anisotropy inherent in window-based transformers by proposing GAPS, a framework utilizing affine-perturbed serialization via our group-wise parallel design to recover non-axial geometric locality. We demonstrate that diversifying scanning paths significantly enhances fine-grained segmentation (e.g., +1.6 % mIoU on ScanNet200) with zero parameter overhead and marginal latency cost, validating the critical role of serialization isotropy in 3D understanding. While GAPS significantly reduces partition-induced errors, the benefits naturally saturate in simple, planar-dominated environments (e.g., S3DIS), suggesting the need for future adaptive mechanisms based on local geometric entropy. By reconciling serialization efficiency with geometric isotropy, GAPS paves the way for scalable, real-time 3D perception.

Appendix A: Reproducibility of affine matrices

For full reproducibility of the GAPS framework, we provide the specific 3×3 diagonal seeds $\mathbf{B}_{\text{seed}}$ used to construct the primary affine-perturbed curves $\mathcal{A}_1$ and $\mathcal{A}_2$. By applying these seeds consistently across all levels $k \in \{1, \ldots, L\}$ and following the iterative algorithm defined in Sect. 3.2.3, the exact scanning paths can be reconstructed.

– $\mathcal{A}_1$ seed matrix ($\mathbf{B}_{\text{seed},1}$): selected to prioritize coordinate permutation, facilitating the capture of oblique 45° geometric dependencies.

$$\mathbf{B}_{\text{seed},1} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{bmatrix} \tag{A1}$$

– $\mathcal{A}_2$ seed matrix ($\mathbf{B}_{\text{seed},2}$): a binary shearing matrix designed to mitigate axial anisotropy by diversifying bit-

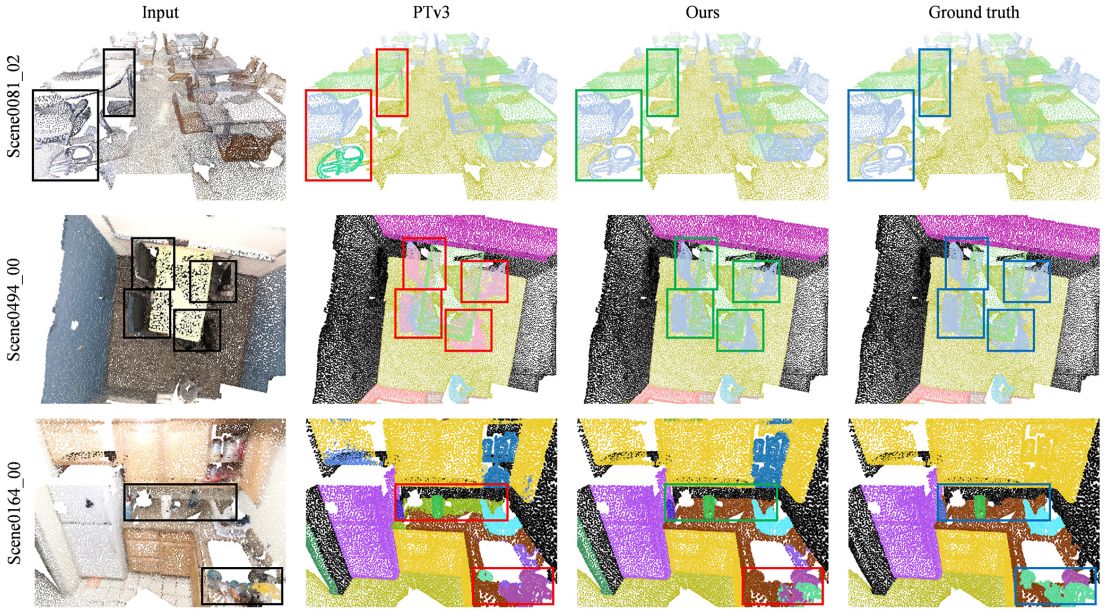

Figure 2. Qualitative comparison on ScanNet200. We highlight baseline failures (red) and our improvements (green). Top: GAPS resolves semantic inconsistency (chair vs. stool). Middle: GAPS distinguishes fine-grained classes (standard vs. office chair). Bottom: GAPS resists semantic bleeding in dense clutter.

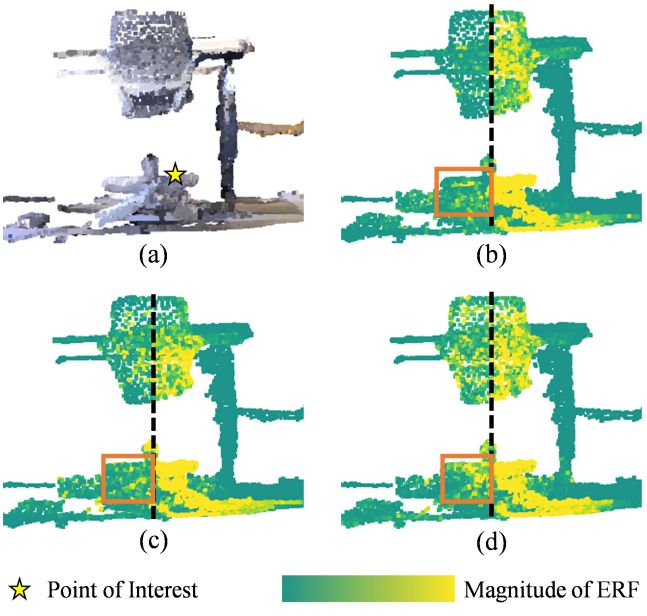

Figure 3. Visualization of the effective receptive field (ERF). **(a)** We visualize activation (teal to yellow) for a query point on the chair base (yellow star). The instance exhibits structural sparsity, separating the seat from the base. **(b)** PTv3 is strictly confined by the partition boundary (dashed line). **(c)** Sequential stacking shows marginal improvement, capturing faint signals on the left. **(d)** GAPS bridges these gaps via affine perturbations, triggering strong activations on the left wheels (orange box) and a more comprehensive focus on the backrest.

level dependencies.

$$\mathbf{B}_{\text{seed},2} = \begin{bmatrix} 1 & 1 & 1 \\ 0 & 1 & 1 \\ 1 & 0 & 1 \end{bmatrix} \tag{A2}$$

The off-diagonal blocks $\mathbf{M}_{i,j}$ are filled using the iterative repulsion algorithm described in Sect. 3.2.3, ensuring each parallel group in the MSFF module provides a unique geometric perspective.

Code availability. The code for this project is available upon request from the corresponding author, Fan Zichuan (fanzc@swu.edu.cn).

Data availability. The ScanNet dataset used in this study is publicly available at http://www.scan-net.org/ (last access: 4 March 2026). The S3DIS dataset used in this study is publicly available at https://doi.org/10.57761/gk3g-wc33 (Armeni et al., 2016b).

Author contributions. Tang Yubin designed the study, developed the methodology, conducted the experiments, and wrote the original draft. Liu Yangchen contributed to the validation and data curation. Fan Zichuan provided the supervision, project administration, and funding acquisition.

Competing interests. The contact author has declared that none of the authors has any competing interests.

Acknowledgements. We thank the Pointcept team for their open-source code, the providers of the ScanNet and S3DISs datasets, and the anonymous reviewers for their valuable feedback.

Review statement. This paper was edited by Pengyuan Zhao and reviewed by two anonymous referees.

References

Armeni, I., Sener, O., Zamir, A. R., Jiang, H., Brilakis, I., Fischer, M., and Savarese, S.: 3D Semantic Parsing of Large-Scale Indoor Spaces, in: 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016, IEEE, 1534–1543, https://doi.org/10.1109/CVPR.2016.170, 2016a.

Armeni, I., Sener, O., Zamir, A. R., Jiang, H., Brilakis, I., Fischer, M., and Savarese, S.: Stanford Large-Scale 3D Indoor Spaces Dataset (S3DIS), Redivis [data set], https://doi.org/10.57761/gk3g-wc33, 2016b.

Charles, R. Q., Su, H., Kaichun, M., and Guibas, L. J.: PointNet: Deep Learning on Point Sets for 3D Classification and Segmentation, in: 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017, IEEE, 77–85, https://doi.org/10.1109/CVPR.2017.16, 2017.

Chen, J., Yu, L., and Wang, W.: Hilbert Space Filling Curve Based Scan-Order for Point Cloud Attribute Compression, IEEE T. Image Process., 31, 4609–4621, https://doi.org/10.1109/TIP.2022.3186532, 2022.

Chen, M., Guo, H., Qian, R., Gong, G., and Cheng, H.: Visual simultaneous localization and mapping (vSLAM) algorithm based on improved Vision Transformer semantic segmentation in dynamic scenes, Mech. Sci., 15, 1–16, https://doi.org/10.5194/ms-15-1-2024, 2024.

Choy, C., Gwak, J., and Savarese, S.: 4D Spatio-Temporal ConvNets: Minkowski Convolutional Neural Networks, in: 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, 15–20 June 2019, IEEE, 3070–3079, https://doi.org/10.1109/CVPR.2019.00319, 2019.

Dai, A., Chang, A. X., Savva, M., Halber, M., Funkhouser, T., and Niessner, M.: ScanNet: Richly-Annotated 3D Reconstructions of Indoor Scenes, in: 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017, IEEE, 2432–2443, https://doi.org/10.1109/CVPR.2017.261, 2017.

Guo, Y., Wang, H., Hu, Q., Liu, H., Liu, L., and Bennamoun, M.: Deep Learning for 3D Point Clouds: A Survey, IEEE T. Pattern Anal., 43, 4338–4364, https://doi.org/10.1109/TPAMI.2020.3005434, 2021.

Hu, Y., Yang, J., Chen, L., Li, K., Sima, C., Zhu, X., Chai, S., Du, S., Lin, T., Wang, W., Lu, L., Jia, X., Liu, Q., Dai, J., Qiao, Y., and Li, H.: Planning-Oriented Autonomous Driving, in: 2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Vancouver, BC, Canada, 17–24 June 2023, IEEE, 17853–17862, https://doi.org/10.1109/CVPR52729.2023.01712, 2023.

Hughes, N., Chang, Y., and Carlone, L.: Hydra: A Real-Time Spatial Perception System for 3D Scene Graph Construction and Optimization, in: Robotics: Science and Systems XVIII, Vol. 18, https://www.roboticsproceedings.org/rss18/p050.html (last access: 31 December 2025), 2022.

Jain, A., Katara, P., Gkanatsios, N., Harley, A. W., Sarch, G., Aggarwal, K., Chaudhary, V., and Fragkiadaki, K.: ODIN: A Single Model for 2D and 3D Segmentation, in: 2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA, 16–22 June 2024, IEEE, 3564–3574, https://doi.org/10.1109/CVPR52733.2024.00342, 2024.

Ji, G., Weder, S., Engelmann, F., Pollefeys, M., and Blum, H.: ARKit LabelMaker: A New Scale for Indoor 3D Scene Understanding, in: 2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Nashville, TN, USA, 10–17 June 2025, IEEE, 4398–4407, https://doi.org/10.1109/CVPR52734.2025.00415, 2025.

Lai, X., Liu, J., Jiang, L., Wang, L., Zhao, H., Liu, S., Qi, X., and Jia, J.: Stratified Transformer for 3D Point Cloud Segmentation, in: 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), New Orleans, LA, USA, 18–24 June 2022, IEEE, 8490–8499, https://doi.org/10.1109/CVPR52688.2022.00831, 2022.

Li, D., Guan, J., Chen, Z., Liao, J., and Du, J.: PointSSM: State Space Model for Large-Scale LiDAR Point Cloud Semantic Segmentation, Int. J. Appl. Earth Obs., 144, 104830, https://doi.org/10.1016/j.jag.2025.104830, 2025a.

Li, Z., Ai, Y., Lu, J., Wang, C., Deng, J., Chang, H., Liang, Y., Yang, W., Zhang, S., and Zhang, T.: Pamba: Enhancing Global Interaction in Point Clouds via State Space Model, in: 39th AAAI Conference on Artificial Intelligence (AAAI), Philadelphia, PA, USA, 25 February–4 March 2025, AAAI Press, Washington, DC, USA, 39, 5092–5100, https://doi.org/10.1609/aaai.v39i5.32540, 2025b.

Liu, Z., Yang, X., Tang, H., Yang, S., and Han, S.: FlatFormer: Flattened Window Attention for Efficient Point Cloud Transformer, in: 2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Vancouver, BC, Canada, 17–24 June 2023, IEEE, 1200–1211, https://doi.org/10.1109/CVPR52729.2023.00122, 2023.

Mazaheri, H., Goli, S., and Nourollah, A.: A Survey of 3D Space Path-Planning Methods and Algorithms, ACM Comput. Surv., 57, 1–32, https://doi.org/10.1145/3673896, 2024.

Mohammadi Amin, F., Caldwell, D. G., and van de Venn, H. W.: Enhancing Human-Robot Collaboration: A Sim2Real Domain Adaptation Algorithm for Point Cloud Segmentation in Industrial Environments, J. Intell. Robot. Syst., 111, 94, https://doi.org/10.1007/s10846-025-02290-9, 2025.

Peng, B., Wu, X., Jiang, L., Chen, Y., Zhao, H., Tian, Z., and Jia, J.: OA-CNNs: Omni-Adaptive Sparse CNNs for 3D Semantic Segmentation, in: 2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR),

Seattle, WA, USA, 16–22 June 2024, IEEE, 21305–21315, https://doi.org/10.1109/CVPR52733.2024.02013, 2024.

Qi, C. R., Yi, L., Su, H., and Guibas, L. J.: PointNet++: Deep Hierarchical Feature Learning on Point Sets in a Metric Space, in: Proceedings of the 31st International Conference on Neural Information Processing Systems (NeurIPS), Red Hook, NY, USA, 4–9 December 2017, 5105–5114, https://proceedings.neurips.cc/paper_files/paper/2017/file/d8bf84be3800d12f74d8b05e9b89836f-Paper.pdf (last access: 4 March 2026), 2017.

Rozenberszki, D., Litany, O., and Dai, A.: Language-Grounded Indoor 3D Semantic Segmentation in the Wild, in: Proceedings of the 17th European Conference on Computer Vision (ECCV), Tel Aviv, Israel, 23–27 October 2022, edited by: Avidan, S., Brostow, G., Cissé, M., Farinella, G. M., and Hassner, T., Lecture Notes in Computer Science, Springer, Cham, 13693, 125–141, https://doi.org/10.1007/978-3-031-19827-4_8, 2022.

Shi, K., Wang, R., Liu, J., Wang, H., and Zhang, D.: Design and analysis of mobile mechanism based on three-dimensional Hilbert curve, Mech. Sci., 16, 851–876, https://doi.org/10.5194/ms-16-851-2025, 2025.

Thomas, H., Qi, C. R., Deschaud, J.-E., Marcotegui, B., Goulette, F., and Guibas, L.: KPConv: Flexible and Deformable Convolution for Point Clouds, in: 2019 IEEE/CVF International Conference on Computer Vision (ICCV), Seoul, Korea (South), 27 October–2 November 2019, IEEE, 6410–6419, https://doi.org/10.1109/ICCV.2019.00651, 2019.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., and Polosukhin, I.: Attention Is All You Need, in: Proceedings of the 31st International Conference on Neural Information Processing Systems (NeurIPS), Red Hook, NY, USA, 4–9 December 2017, 6000–6010, https://proceedings.neurips.cc/paper_files/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf (last access: 4 March 2026), 2017.

Wang, P.-S.: OctFormer: Octree-based Transformers for 3D Point Clouds, ACM T. Graphic., 42, 155, https://doi.org/10.1145/3592131, 2023.

Wu, X., Lao, Y., Jiang, L., Liu, X., and Zhao, H.: Point Transformer V2: Grouped Vector Attention and Partition-Based Pooling, in: Proceedings of the 36th International Conference on Neural Information Processing Systems (NeurIPS), edited by: Koyejo, S., Mohamed, S., Agarwal, A., Belgrave, D., Cho, K., and Oh, A., New Orleans, LA, USA, 28 November–9 December 2022, 35, 33330–33342, https://proceedings.neurips.cc/paper_files/paper/2022/file/d78ece6613953f46501b958b7bb4582f-Paper-Conference.pdf (last access: 4 March 2026), 2022.

Wu, X., Jiang, L., Wang, P.-S., Liu, Z., Liu, X., Qiao, Y., Ouyang, W., He, T., and Zhao, H.: Point Transformer V3: Simpler, Faster, Stronger, in: 2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA, 16–22 June 2024, IEEE, 4840–4851, https://doi.org/10.1109/CVPR52733.2024.00463, 2024.

Xu, R., Li, J., Zhang, S., Li, L., Li, H., Ren, G., and Tang, X.: Interactive trajectory prediction for autonomous driving based on Transformer, Mech. Sci., 16, 87–97, https://doi.org/10.5194/ms-16-87-2025, 2025.

Yang, Y.-Q., Guo, Y.-X., Xiong, J.-Y., Liu, Y., Pan, H., Wang, P.-S., Tong, X., and Guo, B.: Swin3D: A Pre-trained Transformer Backbone for 3D Indoor Scene Understanding, Computational Visual Media, 11, 83–101, https://doi.org/10.26599/CVM.2025.9450383, 2025.

Zeid, K. A., Yilmaz, K., de Geus, D., Hermans, A., Adrian, D., Linder, T., and Leibe, B.: DINO in the Room: Leveraging 2D Foundation Models for 3D Segmentation, arXiv [preprint], https://doi.org/10.48550/arXiv.2503.18944, 2026.

Zhao, W., Zhang, R., Wang, Q., Cheng, G., and Huang, K.: BFANet: Revisiting 3D Semantic Segmentation with Boundary Feature Analysis, in: 2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Nashville, TN, USA, 10–17 June 2025, IEEE, 29395–29405, https://doi.org/10.1109/CVPR52734.2025.02737, 2025.