Journal of Computer and Communications, 2026, 14(1), 46-65
https://www.scirp.org/journal/jcc
ISSN Online: 2327-5227
ISSN Print: 2327-5219

Geo-Refined Point Transformer: Coordinate-Aware Excitation and Positional Upsampling for 3D Scene Segmentation

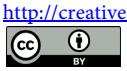
Jingwei Lu*, Yi Zhang

College of Informatics, Huazhong Agricultural University, Wuhan, China
Email: *mayli@webmail.hzau.edu.cn

How to cite this paper: Lu, J.W. and Zhang, Y. (2026) Geo-Refined Point Transformer: Coordinate-Aware Excitation and Positional Upsampling for 3D Scene Segmentation. *Journal of Computer and Communications*, **14**, 46-65.
https://doi.org/10.4236/jcc.2026.141004

Received: December 17, 2025
Accepted: January 16, 2026
Published: January 19, 2026

Copyright © 2026 by author(s) and Scientific Research Publishing Inc.
This work is licensed under the Creative Commons Attribution International License (CC BY 4.0).
http://creativecommons.org/licenses/by/4.0/



Open Access

Abstract

As a work exploring the existing trade-off between accuracy and efficiency in the context of point cloud processing, Point Transformer V3 (PTV3) has made significant advancements in computational efficiency through its innovative point cloud serialization strategy. However, this optimization for computational efficiency comes at the cost of sacrificing high-fidelity perception of fine-grained local geometric structures, thereby introducing a limitation termed "geometric information vacuum" in the model. To address this issue, our work proposes a coordinate-aware feature activation module, which enhances the model's sensitivity to spatial locations by dynamically calibrating feature channel responses using the 3D absolute coordinates of points during the encoder stage. Furthermore, our work designs a position-aware upsampling mechanism that accurately restores the geometric details smoothed out during downsampling by learning a feature compensation term associated with the relative positions of points within voxels during the decoder stage. Experiments on 3D point cloud segmentation on S3DIS and ScanNe t v2 show that the Geo-PT model proposed in this study achieves better performance than PTv3 with negligible additional computational cost.

Keywords

Geometric Information Vacuum, Coordinate-Aware Feature Excitation, Position-Aware Upsampling, Efficiency-Accuracy Trade-Off

1. Introduction

The paradigm of direct point cloud processing pioneered by PointNet [1] has been revolutionized by Transformer-based architectures in recent years. To break

through the bottleneck of quadratic computational complexity imposed by traditional neighborhood queries (e.g., K-Nearest Neighbors and Ball query), Point cloud transformer [2] directly applies a global transformer to all points. This limits its applicability to large-scale point clouds. OctFormer [3] maintains linear complexity through octree sorting and fixed-point count window partitioning. Building on OctFormer, PTV3 [4] adopts an efficient point serialization strategy to significantly reduce computational and memory overheads, enabling training and inference on ultra-large-scale point clouds.

A notable trade-off is that optimizing for computational efficiency requires sacrificing the perception of explicit and high-fidelity local geometry. PTV3 forcibly maps 3D spatial structures to 1D sequences via point cloud serialization, a design that relies on the attention mechanism and positional encoding module to learn spatial neighborhood relationships. While PTV3 increases the probability of long-range point interactions through Shuffle Order strategies, its sequence proximity-based relationship construction is stochastic—it fails to perform deterministic and high-fidelity modeling of each point's true Euclidean spatial neighborhoods in the way that traditional neighborhood querying does. The adverse effects of point cloud serialization are particularly pronounced when processing regions with complex topology or fine-grained structures.

PTV3 incorporates the concept of locality via xCPE [4]. However, its spatially invariant convolution kernels cannot adaptively respond to specific geometric structures at different locations, resulting in insufficient sensitivity to the absolute spatial positions of points and the unique geometric morphologies of local point clusters during the encoder stage. To address this issue, our work proposes a coordinate-aware feature activation module. This module generates attention weights from the 3D absolute grid coordinates of points, which are then element-wise multiplied with features after sparse convolution, thereby achieving spatially adaptive feature calibration. Furthermore, when upsampling coarse feature maps to recover details, unpooling causes all points within the same coarse-grained voxel to receive identical feature updates regardless of their exact relative positions. This makes it difficult for the network to reconstruct the high-frequency geometric information lost during downsampling, leading to geometric ambiguity in the decoder stage. To mitigate this, our work designs a position-aware upsampling mechanism, which introduces a feature compensation term encoded by relative positions to recover high-frequency geometric details.

Overall, our work aims to systematically reintroduce high-fidelity geometric priors into the model without sacrificing the core efficiency of PTV3, while avoiding reverting to the paradigm of performing computationally expensive neighborhood searches at each layer, as seen in Point Transformer v1/v2 (PTV1/PTV2) [5] [6]. The key contributions of our work are as follows:

- We systematically identified and analyzed the "geometric information vacuum" problem in highly efficient serialized point clouds, and proposed a solution consisting of two targeted modules to enhance their geometric capa-

bilities.

- The proposed Coordinate-Aware Feature Excitation (CAFE) module and Position-Aware Upsampling (Pos-Up) module both adhere to the design principles of lightweight and high efficiency, enabling seamless integration into existing serialized point cloud architectures with minimal computational overhead.
- Our Geo-PT achieved performance surpassing PTV3 on multiple large-scale 3D segmentation benchmarks, demonstrating the significance and great potential of precisely incorporating geometric information into serialized point cloud architectures.

2. Related Work

A. 3D Point Cloud Processing Methods

Learning-based approaches to processing 3D point clouds can be classified into the following types: projection-based, voxel-based, and point-based networks.

Projection-based Networks. One approach to converting irregular point clouds into regular representations is to map them onto 2D image spaces, leveraging the strong representational capabilities of mature 2D CNNs [7]. Such methods circumvent the complexity of 3D geometric processing via multi-view or viewpoint-specific projections. For instance, MVCNN [8] renders 3D shapes into multi-view images; after feature extraction via CNNs with shared weights, it enables recognition through view pooling. RotationNet [9] renders images from unsupervised viewpoints and jointly learns classification and pose to enhance viewpoint robustness; PointPillars [10] projects point clouds into bird's-eye-view pseudo-images, enabling the direct and efficient application of 2D detection frameworks. However, the projection process loses 3D geometric details (e.g., spatial topology and fine-grained structures), and 3D occlusions cause information loss in 2D images—fundamentally limiting perceptual accuracy.

Voxel-based Networks. Another approach to converting irregular point clouds into regular representations is to represent 3D data using uniformly sampled voxels before applying 3D convolution. However, constrained by voxel resolution, computational and memory costs grow cubically with resolution. The solution leverages sparsity, as most voxels are unoccupied. For example, OctNet [11] employs an unbalanced octree with hierarchical partitioning; MinkowskiNet [12], based on sparse convolution, operates only on non-empty voxels, further reducing computational and memory requirements. These methods have demonstrated good accuracy; however, quantizing to voxel grids still results in the loss of geometric details.

Point-based Networks. Such networks do not require rasterizing 3D shapes into regular voxels, directly taking raw point clouds as input. Due to the disorder and non-structurality of point clouds, they employ permutation- invariant operations, continuous convolution kernels, or adaptive weights to aggregate and update point features. For example, PointCNN [13] addresses the disorder of points

by weighting and rearranging local point clouds via learned X-transforms, enabling the direct application of standard convolution operations to irregular point sets; PointNet [1] aggregates point set features using permutation-invariant operators (point-wise MLPs and max pooling); DGCNN [14] explicitly encodes edge features of point clouds in dynamically updated graph structures; KPConv [15] enhances geometric perception using deformable convolution kernels. These methods emphasize direct processing of raw point clouds without voxel quantization, capturing local geometry more accurately than fixed-kernel convolutions while balancing accuracy and efficiency. Transformer-based networks introduced in the next section belong to the category of point-based networks for point cloud understanding [5].

Transformer-based Networks. CNNs' fixed receptive fields and local kernels struggle to adapt to the dynamic distribution of point clouds, whereas the self-attention of Transformers can dynamically adjust weights, naturally aligning with the set properties of point clouds. For instance, Point Cloud Transformer [2] applies offset attention to all point features; Point-MAE [16] leverages standard Transformers for unsupervised pre-training on point clouds; Swin Transformer [17] introduces a grid-based local attention mechanism, operating Transformer blocks within a series of shifted windows. These methods operate independently on the local neighborhood of each point, but the lack of computational sharing between overlapping neighborhoods results in a significant waste of computational resources. Motivated by ViT [18], which partitions images into regular patches and processes them sequentially, OctFormer [3] uses the z-order sorting of octrees to divide the input point cloud into groups with equal numbers of points, facilitating parallel computing and scalable expansion; FlatFormer [19] employs window-based pillar sorting to partition point clouds into local sequences, suitable for large-scale outdoor detection tasks while balancing efficiency and locality; Building on OctFormer, PTV3 [4] adopts an efficient point serialization strategy and flash attention to reduce computational and memory overheads, enabling training and inference on ultra-large-scale point clouds.

B. Modeling Geometric Information in Point Cloud Learning

Modeling geometric information has been proven to be a key factor in improving the performance of point cloud understanding tasks. Existing studies have primarily explored this from two perspectives:

Geometric Awareness from the Perspective of Network Module Design. During the feature encoding stage, researchers have proposed various geometrically adaptive operations. For instance, PointWeb [20] explicitly models fine-grained geometric relationships through dense feature interactions between point pairs within local neighborhoods and adaptive propagation mechanisms; Dynamic Kernel [21] dynamically generates convolution kernel weights based on input features, enabling adaptive modeling of local geometric structures; A-CNN [22] designs a circular convolution structure, which captures fine-grained local geometric features via multi-scale dilated rings.

Geometric Detail Reconstruction in the Decoder Stage. During the upsampling and feature reconstruction stage, existing works primarily compensate for information loss via geometric priors. RepSurf [23] corrects feature offsets using surface parameterization; PU-GAN [24] employs neighborhood-based interpolation to achieve final feature assignment and point generation; PU-Net [25] achieves detail enhancement through multiscale feature concatenation.

Most of these methods are tightly coupled with neighborhood search-based architectures. How to efficiently and lightweightly inject these proven effective geometric priors into the latest serialized point cloud architectures remains an open and critical issue. Our work directly addresses this challenge.

3. Methodology

This chapter elaborates on the Geo-Refined Point Transformer (Geo-PT) proposed in this study. First, Section III. A reviews the baseline model PTV3 and conducts an in-depth analysis of the "geometric information vacuum" problem resulting from its pursuit of ultimate efficiency. Subsequently, Sections III.B and III.C introduce the two core modules designed to address this vacuum, respectively: the Coordinate-Aware Feature Excitation (CAFE) module and the Position-Aware Upsampling (Pos-Up) mechanism. These two modules together form a lightweight geometric refinement solution, aimed at improving segmentation accuracy without sacrificing the core efficiency of serialized point cloud architectures.

A. Revisiting PTV3: Beneath Efficiency the Geometric Information Vacuum

PTV3's overall architecture follows a hierarchical encoder-decoder design (see Figure 1). Its core lies in mapping 3D point clouds into 1D sequences via space-filling curves, thereby leveraging the efficient Transformer architecture. However, while pursuing ultimate efficiency, this design introduces two key limitations:

- **Geometric Insensitivity in the Encoding Stage.** In the encoding stage, PTV3 incorporates an efficient 3D sparse submanifold convolution (xCPE) that aggregates local features before the attention module to compensate for the loss of local geometric information caused by serialization. However, a core property of traditional convolution is spatial invariance. This means a point cluster is processed with the same set of shared weights, whether it lies at the center of a flat wall surface or a sharp object corner. While efficient, this design sacrifices the ability to adaptively model different spatial locations and specific fine geometric structures, resulting in the model lacking perception of the absolute position of a point or region in its scene.
- **Geometric Ambiguity in the Decoding Stage.** In the decoding stage, PTV3 adopts unpooling for its upsampling operation. This operation directly copies features from coarse-level voxels to all high-resolution points that fall within the voxel. The process ignores the precise relative position of each high-resolution point within its parent voxel, resulting in identical feature increments for points located at the center or edge of the voxel. Ultimately, this fails to

recover the high-frequency geometric information lost during downsampling, leading to smooth, blurry segmentation results at object boundaries and fine structures.

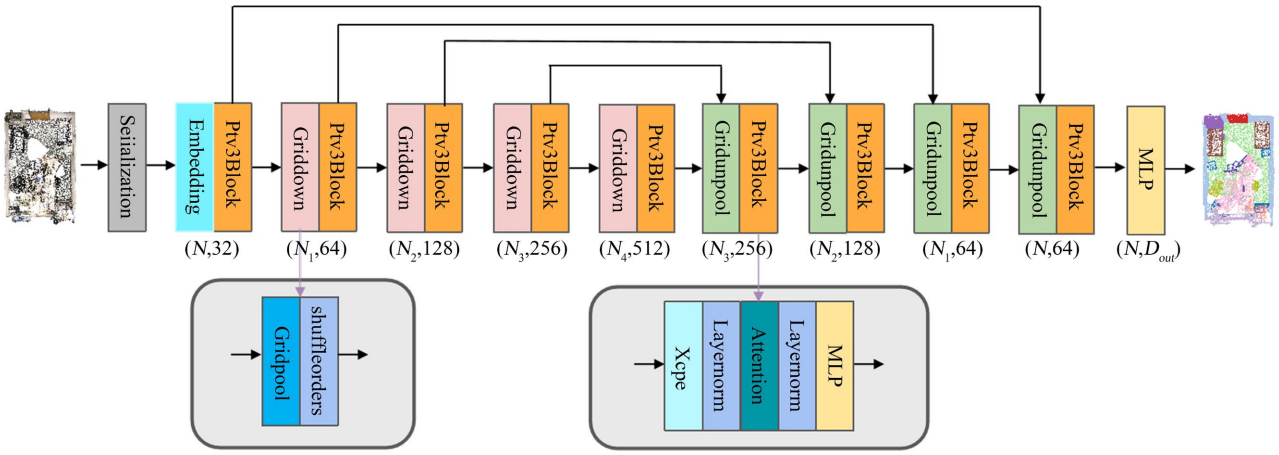


Figure 1. PTV3 architecture.

B. Encoder Enhancement: Coordinate-Aware Feature Excitation

To address the spatial invariance issue of the xCPE module in the encoding stage, we introduce the CAFE module, drawing inspiration from CoordConv [26] (which uses coordinates for spatial adaptive calibration) and GACNet [27] (which adopts an attention mechanism for adaptive weighting of feature responses). Unlike the classic Squeeze-and-Excitation Networks [28], which generate channel attention from global feature pooling, the innovation of CAFE lies in that attention weights are directly generated from the 3D absolute grid coordinates of points. This design enables the network to learn a spatially adaptive feature recalibration strategy. CAFE is subtly integrated inside the xCPE block, functioning after the sparse convolution and before the residual connection (see **Figure 2**). Specifically, the CAFE module takes the input feature tensor $f_{in} \in \mathbb{R}^{N\times C}$ and the corresponding discrete integer grid coordinates $p_{grid} \in \mathbb{Z}^{N\times 3}$ as input, its complete computation process is expressed by the following formulas:

$$f_{out} = f_{in} + \sigma\left(\alpha\left(p_{grid}\right)\right) \odot \beta\left(f_{in}\right) \tag{1}$$

where β denotes the standard sparse 3D convolution operation in the xCPE module. α is a mapping function (such as MLP) used to map input 3D coordinates to a vector matching the dimension of feature channels. This vector is first activated by σ (Sigmoid function) to generate channel attention weights within the range of $(0,1)$. Subsequently, the weight vector dynamically adjusts the feature map extracted by β via element-wise multiplication. Finally, it is added to the original input f_{in} through a residual connection, forming the final output feature f_{out}.

Notably, this design stands in stark contrast to many previous works [1] [5] [6] [29] [30]. The latter primarily utilizes relative coordinates to encode the invari-

ance of local geometry, while the former deliberately employs absolute coordinates to break the invariance of global space. Through this design, the network can learn context-aware feature representations. For instance, it enhances the response of a specific feature channel for points near the ground, while suppressing its activation in ceiling areas. Furthermore, since this mechanism only introduces a minuscule MLP, the resulting computational and parameter overheads are nearly negligible. Thus, it enhances the model's geometric awareness and scene understanding capability without sacrificing the computational efficiency of the baseline model.

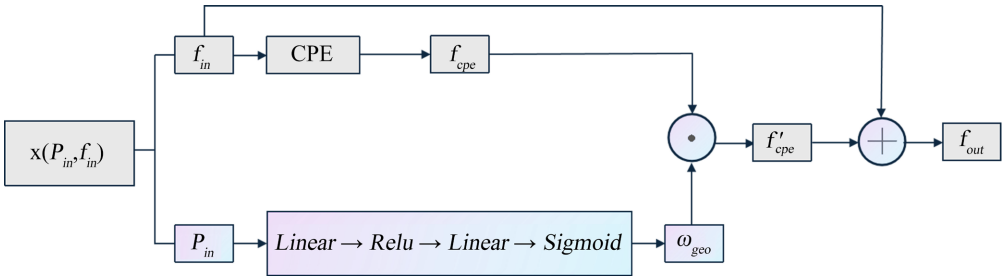


Figure 2. Coordinate-aware feature excitation module.

C. Decoder Enhancement: Positional Upsampling

To address the issue of blurred geometric details caused by unpooling in the decoding stage, we introduce the Pos-Up mechanism, drawing on the successful practices of RepSurf [23] and PU-GAN [24]—which use coordinate offsets to learn feature compensation terms for modeling continuous surfaces and recovering details. Our goal is to introduce a feature compensation term encoded by precise relative positions based on the unpooling operation (see **Figure 3**).

In unpooling, for a point $p_i \in P_{s-1}$ upsampled from the coarse level s to the fine level $s-1$, its feature is directly copied from the feature of the parent node $p_{parent(i)} \in P_s$. Pos-Up corrects this. For each fine point p_i, we first calculate the normalized coordinates relative to the center of the parent voxel:

$$\Delta p_i = \frac{p_i - p_{parent(i)}}{voxelsize_s} \tag{2}$$

where $voxelsize_s$ denotes the voxel size of the coarse level s, serving as the normalization factor. Δp_i precisely encodes the micro-position of point p_i precisely encodes the micro-position of point p_i within its parent voxel. Normalizing the relative coordinates not only enhances the model's equivariance to translation and stabilizes the training process, but more importantly, enables the MLP to learn a standardized "local position-feature compensation" function, endowing it with generalization ability across voxels of different sizes [30].

The normalized position encoding Δp_i is fed into a small MLP to learn a feature compensation vector δ_i with the same feature dimension:

$$\delta_i = MLP_{pos}\left(\Delta p_i\right) \tag{3}$$

The compensation vector Δp_i is subsequently directly concatenated to the standard upsampled feature f_i^{up}, forming the final position-aware upsampled feature f_i^{pos-up}:

$$f_i^{pos-up} = f_i^{up} + \delta_i \tag{4}$$

Finally, the position-compensated upsampled features are concatenated and fused with the skip connection features from the corresponding levels of the encoder. This thereby accurately recovers the high-frequency geometric details lost during downsampling in the decoding stage, especially at object boundaries and complex structures.

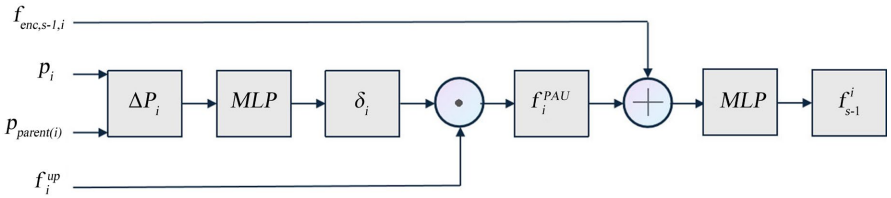


Figure 3. Position-aware upsampling module.

4. Experiments

To validate the effectiveness of the two proposed modules, this study conducted semantic segmentation experiments on the mainstream datasets of ScanNet v2 [31] and S3DIS [32].

A. Data and metric

This study evaluates our model on two widely used large-scale indoor scene segmentation benchmarks: ScanNet v2 and S3DIS. The ScanNet v2 dataset contains 1513 indoor scenes reconstructed from RGB-D video frames, which are officially split into 1201 training scenes and 312 validation scenes. Point clouds are sampled from the vertices of the reconstructed meshes and annotated with 20 semantic categories. The S3DIS dataset consists of six areas across three different buildings, totaling 271 rooms. Following the common evaluation protocol [6] [29] [33], we use Area-5 as the test set for evaluation, with the remaining five areas used for training. Point clouds in S3DIS are densely sampled on mesh surfaces and annotated into 13 categories.

Following the standard protocol [6], we use mean Intersection over Union (mIoU) as the evaluation metric for the validation and test sets of ScanNet v2. For evaluating the performance of Area-5 of S3DIS, we employ mIoU, mean class Accuracy (mAcc), and Overall Accuracy (OA).

B. Implementation Details

This study's model is implemented based on the Pointcept codebase, a dedicated codebase focused on point cloud perception and representation learning. All models are trained on four NVIDIA RTX 4090 GPUs. The optimizer used is AdamW, and the learning rate scheduler is OneCycleLR. This section provides a detailed description of the details of our model implementation.

B.1. Training Settings

Indoor Semantic Segmentation. Table 1 outlines the settings for indoor semantic segmentation.

Table 1. Indoor semantic segmentation settings.

Config	ScanNet v2	ScanNet 200	S3DIS
optimizer	AdamW	AdamW	AdamW
scheduler	Cosine	Cosine	Cosine
criteria	CrossEntropy	CrossEntropy	CrossEntropy
	Lovasz	Lovasz	Lovasz
learning rate	2e−3	2e−3	2e−3
block lr scaler	0.1	0.1	0.1
weight decay	2e−2	2e−2	2e−2
batch size	12	12	12
warmup epochs	40	40	40
epochs	800	800	800

B.2. Model Settings

Our model configurations are comprehensively listed in Table 2, and the data augmentation used is presented in Table 3.

Table 2. Model settings.

Config	Value
serialization pattern	$Z+TZ+H+TH$
patch interaction	Shift Order + Shuffle Order
embedding depth	2
embedding channels	48
encoder depth	[3, 3, 12, 3]
encoder channels	[96, 192, 384, 512]
encoder num heads	[6, 12, 24, 32]
encoder patch size	[1024, 1024, 1024, 1024]
decoder depth	[2, 2, 2, 2]
decoder channels	[64, 96, 192, 384]
decoder num heads	[4, 6, 12, 24]
decoder patch size	[1024, 1024, 1024, 1024]
down stride	[×2, ×2, ×2, ×2]
mlp ratio	4
qkv bias	True
enable flash	True
pre norm	True

Table 3. Data augmentation setting.

Augmentations	Parameters
random rotate target angle	angle: [0.5, 1, 1.5], p = 0.75
random rotate	axis: z, angle: [−1, 1], p: 0.5
	axis: x, angle: [−1/64, 1/64], p: 0.5
	axis: y, angle: [−1/64, 1/64], p: 0.5
random scale	scale: [0.9, 1.1]
random shift	shift: [[−0.2, 0.2], [−0.2, 0.2], [−0.2, 0.2]]
random flip	p: 0.5
random jitter	sigma: 0.005, clip: 0.02
auto contrast	p: 0.2
chromatic translation	p: 0.95, ratio: 0.05
color jitter	std: 0.05; p: 0.95
random color drop	p: 0.2, color_augment: 0.0
grid sampling	grid size: 0.02 (indoor), 0.05 (outdoor)
sphere crop	ratio: 0.8, max points: 128000
normalize color	p: 1

C. Performance comparison

We benchmarked the performance of Geo-PT against previous state-of-the-art models and report the top results obtained for each benchmark. In **Table 4**, we present the validation and test performance of Geo-PT on the ScanNet v2 [31] and ScanNet200 [34] benchmarks, as well as its performance on Area-5 of S3DIS and 6-fold cross-validation [1] (see **Table 5** for details). Results demonstrate that Geo-PT achieves superior performance across all benchmarks. Notably, compared to the strong baseline model PTV3, Geo-PT achieves a 0.8% mIoU improvement on the ScanNet v2 validation set and a 1.31% mIoU improvement on S3DIS Area-5 with almost no additional computational overhead, fully verifying the effectiveness of our proposed geometric refinement modules.

We evaluate model efficiency based on average inference latency and peak memory consumption on the ScanNet v2 validation set. Efficiency metrics are measured on a single RTX 4090, excluding the first iteration to ensure steady-state measurement. As shown in **Table 6**, after integrating the CAFE and Pos-Up modules, our Geo-PT exhibits only negligible increases in inference latency and memory consumption compared with the baseline PTV3. Experiments demonstrate that while achieving performance improvements, the geometric refinement modules successfully maintain the core efficiency advantages of the serialized point cloud architecture.

Table 4. Indoor semantic segmentation results.

Methods	ScanNet200		ScanNet200		S3DIS	
	Val	Test	Val	Test	Area5	6-fold
ST [35]	74.3	73.7	–	–	72.0	–
PointNeXt	71.5	71.2	–	–	70.5	74.9
OctFormer	75.7	76.6	32.6	32.6	–	–
Swin3D [36]	76.4	–	–	–	72.5	76.9
PTv1	70.6	–	27.8	–	70.4	65.4
PTv2	75.4	74.2	30.2	–	71.6	73.5
PTv3	76.5	77.8	33.2	34.6	73.38	77.64
Geo-PT	**77.1**	**78.6**	**33.8**	**35.7**	**74.69**	**80.88**

Table 5. S3DIS 6-fold cross-validation results.

Method	Metric	Area1	Area2	Area3	Area4	Area5	Area6	6-fold
PTv2	allAcc	92.30	86.00	92.98	89.23	91.24	94.26	90.76
	mACC	88.44	72.81	88.41	82.50	77.85	92.44	83.13
	mIoU	81.14	61.25	81.65	69.06	72.02	85.95	75.17
PTv3	allAcc	93.22	86.26	94.56	90.72	92.00	94.98	91.53
	mACC	89.92	74.44	94.45	81.11	79.70	93.55	85.31
	mIoU	83.01	63.42	86.66	71.34	73.38	87.31	77.70
Geo-PT	allAcc	93.27	90.17	94.54	92.12	91.99	95.06	92.82
	mACC	90.69	82.46	93.79	83.81	79.90	93.69	88.83
	mIoU	**83.74**	**70.41**	**87.84**	**74.96**	**74.09**	**88.43**	**80.81**

Table 6. Efficiency Comparison on ScanNet v2 Validation Set.

Indoor Efficiency (ScanNet)					
Methods	Params	Training		Inference	
		Latency	Memory	Latency	Memory
MinkUNet [12]	37.9 M	267 ms	4.9 G	90 ms	4.7 G
OctFormer	44.0 M	264 ms	12.9 G	86 ms	12.5 G
Swin3D	71.1 M	602 ms	13.6 G	456 ms	8.8 G
PTv2	12.8 M	312 ms	13.4 G	191 ms	18.2 G
PTv3	46.2 M	182 ms	7.2 G	121 ms	3.3 G
Geo-PT	**46.5 M**	**185 ms**	**7.35 G**	**122 ms**	**3.3 G**

D. Visualization

The qualitative results of point cloud semantic segmentation are shown in Figure 4 and Figure 5. Our model is capable of generating semantic segmentation

results that closely match real-world scenes. Notably, the model performs exceptionally well in capturing fine-grained structural information and can make accurate semantic predictions in complex scenes. For instance, in S3DIS scenes containing door frames, the Geo-PT model can clearly predict the structure of door frames, demonstrating excellent recovery performance.

E. Ablation Study

We conducted a series of ablation studies on the S3DIS Area-5 dataset to validate the effectiveness of the various components and design choices we proposed.

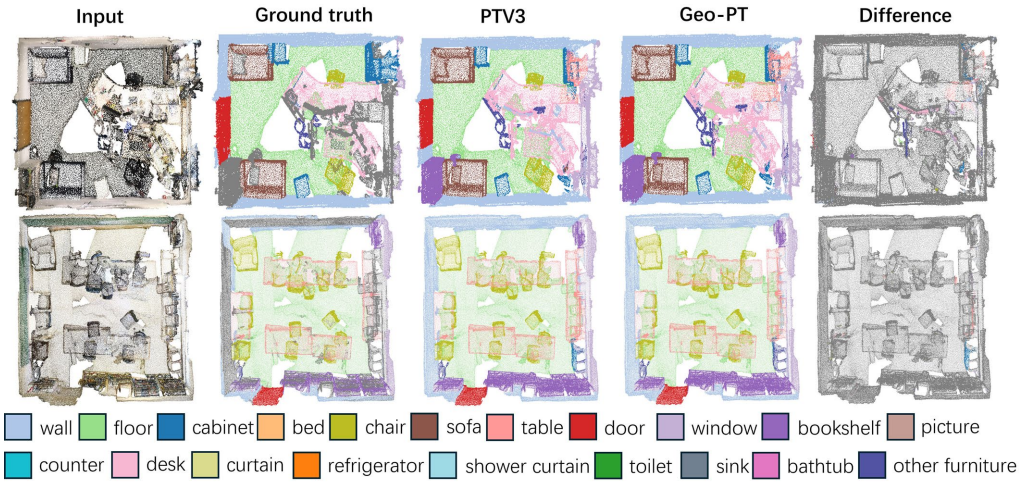


Figure 4. Visualization results of the ScanNet segmentation dataset.

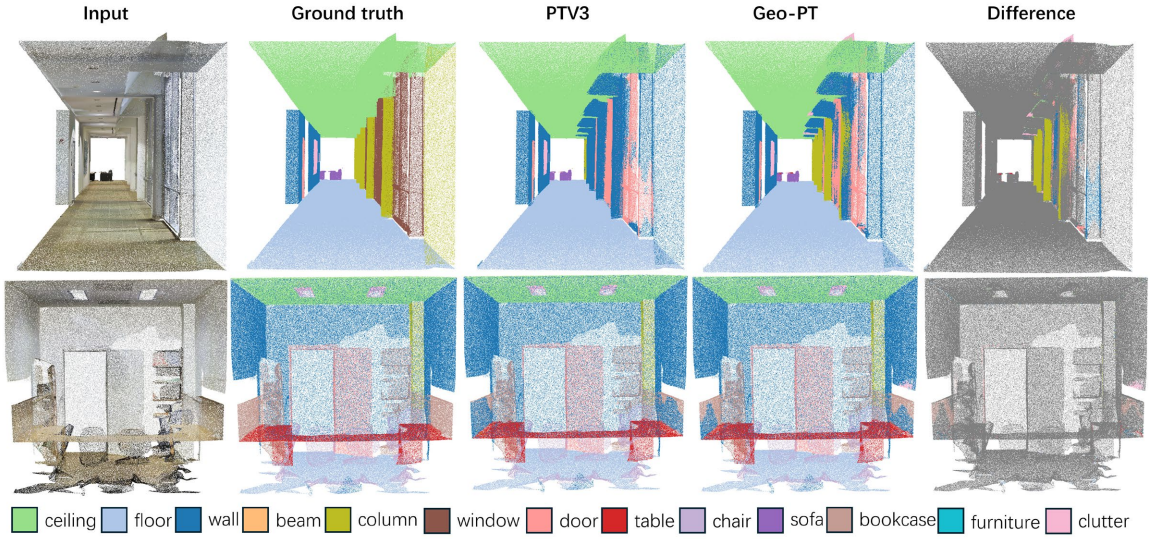


Figure 5. Visualization results of the S3DIS segmentation dataset.

E.1. Analysis of the Effectiveness of Core Modules.

We analyzed the independent contributions of the CAFE and Pos-Up modules. As shown in Table 7, the mIoU of the baseline model PTV3 is 73.38%. By only incorporating the CAFE module into the encoder, the performance increases to

74.07%, demonstrating the benefits of spatially adaptive feature calibration. Similarly, by only integrating the Pos-Up module into the decoder, the mIoU also rises to 73.68%, confirming the importance of accurately recovering details during the upsampling stage. Our complete model Geo-PT, which combines these two modules, achieves the best performance of 74.69%—this indicates that there is a synergistic effect between enhancing geometric perception in the encoder and refining geometric details in the decoder.

Table 7. Ablation study of core modules on S3DIS Area-5.

Methods	CAFE	Pos-Up	mIoU (%)	mAcc (%)	OA (%)
Baseline (PTV3)			73.38	79.70	92.00
PTV3 (+CAFE)	✓		74.07	80.66	92.05
PTV3 (+Pos-Up)		✓	73.68	80.92	91.54
Geo-PT	✓	✓	**74.69**	**81.98**	**92.75**

E.2. Analysis of Geometric Information Injection Strategies.

A key design question is when and how geometric priors should be injected into the serialized point cloud architecture. To fully verify the superiority of our proposed in-network geometric refinement strategy (CAFE and Pos-Up), we designed and evaluated two alternative schemes. Below, we provide a detailed introduction to the aforementioned GPC module and ASF module.

E.2.1. Implementation Details of the GPC Module

Accurate geometric priors are crucial for 3D point cloud segmentation tasks, especially in the process of recovering fine-grained geometric details during the decoder stage. To address the limitation of the "geometric information vacuum" posed by PTV3, we attempted to compensate using a "homogeneous hybrid" architecture called GPC. Unlike CoAtNet [37], a "heterogeneous hybrid" model that combines the advantages of convolutional networks and Transformers, our motivation is not to fuse modules that process different data modalities, but to focus on fusing two paradigms for processing point clouds: local geometric aggregation based on real neighborhoods and global sequence modeling based on pseudo-neighborhoods.

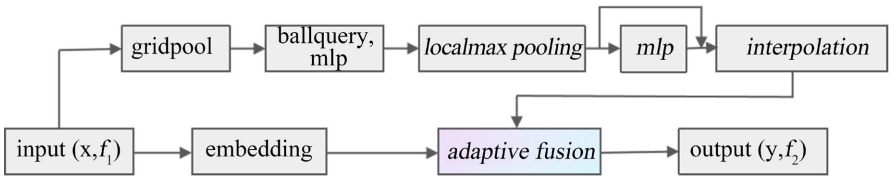


Figure 6. GPC module.

We designed an efficient Geometric Pre-Corrector (GPC), whose structure is illustrated in **Figure 6**. Serving as an independent "geometric engine," this module

aims to refine and encode rich local spatial relationships from raw coordinates, and "pre-correct" such information into input features—enabling them to carry stronger geometric priors before entering the serialized encoding pipeline.

The design of GPC draws on the inverted bottleneck design of efficient operators in PointNeXt, and incorporates our unique fusion mechanism. Its computational flow is as follows:

To improve computational efficiency while preserving the global structure, we first downsample the input point cloud P via voxel grid averaging to obtain a sparse set of key points P_{key}. Subsequently, for any key point P_i in P_{key}, with a ball query radius set to r, we obtain its real neighborhood $N_i = \left\{ p_j \mid dist\left(p_i, p_j\right) < r \right\}$. Inspired by PointNet++, we capture local structures by encoding relative positions within the neighborhood. Specifically, we concatenate the normalized relative coordinates of neighborhood points with their raw absolute coordinates to form a purely geometric input vector h_j:

$$h_j = Concat\left(\frac{p_j - p_i}{r}, p_j \right), \forall p_j \in N\left(i\right) \tag{5}$$

Research on PointNext notes that normalizing relative coordinates can stabilize the training process, avoiding gradient instability in the network caused by processing extremely small values. Subsequently, we adopt an inverted residual MLP structure to aggregate neighborhood information. This structure decouples spatial aggregation from channel transformation, achieving a balance between lightweightness and efficiency: it processes the geometric vectors h_j of all neighborhood points through a shared MLP, and aggregates information via a max pooling operation to obtain the preliminary geometric representation f_i' of p_i:

$$f_i' = \underset{j \in N(i)}{MaxPool} \left\{ MLP_{local}\left(h_j\right) \right\} \tag{6}$$

Subsequently, f_i' is fed into a point-wise inverted residual MLP. This MLP first expands the feature channels, then compresses them back to the original dimension, and adds the result to the input via a residual connection.

This "expand-first-then-compress" design allows the network to perform nonlinear transformations in a higher-dimensional space, significantly enhancing the representational capacity of the module while maintaining a low parameter count. Finally, we obtain the geometric features F_i^{geo} on the key points:

$$F_i^{geo} = f_i' + MLP_{pw}\left(f_i'\right) \tag{7}$$

Finally, the geometric features F_i^{geo} computed on the key point set P_{key} are efficiently propagated back to each point in the original dense point cloud P via trilinear interpolation, yielding the dense geometric features F_{geo}.

To fuse the purely geometric features F_{geo} extracted by GPC with the semantic features F_{sem} output by the Embedding layer of the backbone network, we designed two schemes: one is gated fusion, and the other is direct concatenation. Taking gated fusion as an example, this module dynamically learns a fusion weight $\omega \in \left[0,1\right]$ based on the concatenation of the two types of features, and generates

the final enhanced feature $F_{enhanced}$ in the following manner:

$$\omega = Sigmoid\left(Linear\left(Concat\left(F_{sem}, F_{geo}\right)\right)\right) \tag{8}$$

$$F_{enhanced} = (1-\omega) \odot F_{sem} + \omega \odot F_{geo} \tag{9}$$

Although this approach enables the network to adaptively decide whether to rely more on the original semantic information or the geometric priors provided by GPC at each point, it could theoretically achieve effective "pre-correction" of the input. However, experimental results show that regardless of whether simple additive fusion or gated fusion is adopted, the GPC module leads to a performance drop (see Table 8). We hypothesize that the geometric features explicitly and independently extracted by the GPC module may be incompatible with the high-level abstract spatial representations implicitly learned by the Transformer backbone network at deep levels.

E.2.2. Implementation Details of the ASF Module

In the standard U-Net architecture, the decoder typically only fuses encoder features of the same scale. Considering that the semantic information contained in high-level features and the geometric information in low-level features are complementary, we attempted to draw on the idea of dense skip connections from architectures such as UNet++ [38] and UNet3+ [39] in the field of 2D image segmentation, and designed and evaluated an alternative decoder enhancement scheme.

Its core hypothesis is that: when the decoder performs feature reconstruction at a certain scale s, in addition to relying on contextual information from the encoder E_s of the same scale, it may directly benefit from high-frequency geometric details retained in the adjacent, finer encoder layer E_{s-1}. However, considering that full-scale fusion strategies in point cloud processing would involve multiple expensive interpolations or aggregations, introducing unacceptable computational overhead, we explored this pragmatic compromise scheme that only fuses adjacent scales. This module is intended to serve as a comparative baseline for our main contributions, to investigate the impact of different geometric information injection strategies on model performance.

The ASF module modifies the standard upsampling pipeline, whose data flow is illustrated in Figure 7. The execution flow of the ASF module can be summarized as follows:

First is Multi-source Feature Preparation, where the module aggregates three feature streams:

- Upsampled Features f_{up}: Base features upsampled from the deeper and coarser decoder layer D_{s+1}.
- Same-scale Skip Features f_{skip}: Standard skip connection features from the corresponding encoder layer E_s.
- Adjacent Fine-scale Features f_{finear}: Features from the encoder layer E_{s-1}, which retains more geometric information.

Next is Efficient Scale Alignment. Aligning the high-resolution f_{finear} feature

map with the current low-resolution decoding features is a key step. To avoid introducing high-cost operations such as K-nearest neighbor interpolation, we adopted an efficient, non-parametric aggregation method. This method cleverly leverages the parent-child node mapping relationship of point clouds generated during encoder downsampling (from E_{s-1} to E_s).

By reusing this mapping, we can directly perform mean aggregation on the features of all child nodes corresponding to the same parent node in f_{finear}. This operation does not require learning any new parameters, yet can quickly aggregate fine-scale feature information to the current decoding scale, generating the aligned feature f_{finear_agg}.

Finally, Fusion and Feature Alignment. After scale alignment of the three feature streams ($f_{up}, f_{skip}, f_{finear\_agg}$), we concatenate them along the channel dimension. To enable the network to adaptively learn how to integrate this information with diverse sources and scales, the concatenated features are fed into a fusion MLP with a bottleneck structure. Through non-linear transformations of first compressing then expanding, this MLP aims to learn optimal feature combination weights, and finally outputs fused features for use by subsequent decoder blocks.

Although this design can theoretically provide additional high-frequency information, as shown in our ablation study (Table 8), it did not lead to performance improvement. We speculate that this relatively straightforward feature-level fusion is less effective and direct than Pos-Up, which directly performs fine-grained compensation at the coordinate level.

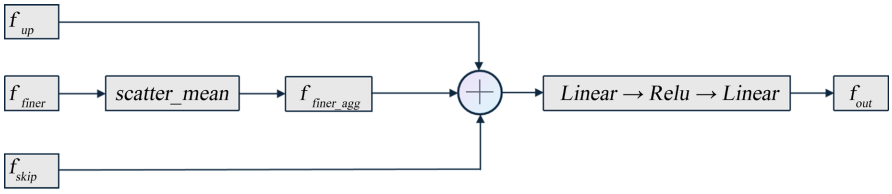


Figure 7. ASF module.

Table 8. Performance comparison of different geometric injection strategies.

Methods	Description	mIoU (%)
Baseline (PTV3)	—	73.38
Baseline + GPC (Add)	Additive Fusion	72.24
Baseline + GPC (Gate)	Gated Fusion	70.33
Baseline + ASF	Adjacent Scale Fusion	73.12
Geo-PT	**In-Network Geometric Refinement**	**74.69**

This comparative experiment strongly demonstrates the impact of geometric information injection methods on model performance. Compared with simple pre-pretreatment or crude multi-scale fusion, our in-network refinement methods (CAFE and Pos-Up) are superior due to contextualization and adaptability.

CAFE dynamically adjusts features based on context during the encoding process, while Pos-Up performs precise detail compensation as needed during decoding. This correction strategy, which is deeply coupled with the model processing pipeline, fills the "geometric information vacuum" more effectively than "one-time" external interventions.

5. Conclusion

This study systematically analyzes and identifies the "geometric information vacuum" problem existing in current efficient serialized point cloud architectures—specifically, these architectures sacrifice the ability to perceive fine-grained geometric structures in pursuit of extreme computational efficiency. To address this issue, this study proposes a set of lightweight geometric refinement modules consisting of CAFE and Pos-Up. Results demonstrate that Geo-PT achieves a 1.3% improvement on the S3DIS segmentation dataset and a 0.8% improvement on the ScanNet v2 segmentation dataset without sacrificing core efficiency. This work demonstrates the feasibility and great potential of performing fine-grained, targeted geometric information compensation in efficient serialized point cloud architectures.

Conflicts of Interest

The authors declare no conflicts of interest regarding the publication of this paper.

References

[1] Qi, C.R., Su, H., Mo, K. and Guibas, L.J. (2017) PointNet: Deep Learning on Point Sets for 3D Classification and Segmentation. arXiv: 1612.00593.

[2] Guo, M., Cai, J., Liu, Z., Mu, T., Martin, R.R. and Hu, S. (2021) PCT: Point Cloud Transformer. *Computational Visual Media*, **7**, 187-199. https://doi.org/10.1007/s41095-021-0229-5

[3] Wang, P. (2023) OctFormer: Octree-Based Transformers for 3D Point Clouds. *ACM Transactions on Graphics*, **42**, 1-11. https://doi.org/10.1145/3592131

[4] Wu, X., Jiang, L., Wang, P., Liu, Z., Liu, X., Qiao, Y., *et al.* (2024) Point Transformer V3: Simpler, Faster, Stronger. 2024 *IEEE/CVF Conference on Computer Vision and Pattern Recognition* (*CVPR*), Seattle, 16-22 June 2024, 4840-4851. https://doi.org/10.1109/cvpr52733.2024.00463

[5] Wu, X., Lao, Y., Jiang, L., Liu, X. and Zhao, H. (2022) Point Transformer V2: Grouped Vector Attention and Partition-Based Pooling. arXiv: 2210.05666.

[6] Zhao, H., Jiang, L., Jia, J., Torr, P. and Koltun, V. (2021) Point Transformer. 2021 *IEEE/CVF International Conference on Computer Vision* (*ICCV*), Montreal, 10-17 October 2021, 16239-16248. https://doi.org/10.1109/iccv48922.2021.01595

[7] Vafeiadis, T., Kolokas, N., Dimitriou, N., Zacharaki, A., Yildirim, M., Selvi, H.G., *et al.* (2022) A Comparison of 2DCNN Network Architectures and Boosting Techniques for Regression-Based Textile Whiteness Estimation. *Simulation Modelling Practice and Theory*, **114**, Article ID: 102400. https://doi.org/10.1016/j.simpat.2021.102400

[8] Su, H., Maji, S., Kalogerakis, E. and Learned-Miller, E. (2015) Multi-View Convolutional Neural Networks for 3D Shape Recognition. 2015 *IEEE International Confer-*

ence on Computer Vision (*ICCV*), Santiago, 7-13 December 2015, 945-953. https://doi.org/10.1109/iccv.2015.114

[9] Kanezaki, A., Matsushita, Y. and Nishida, Y. (2018) RotationNet: Joint Object Categorization and Pose Estimation Using MultiViews from Unsupervised Viewpoints. 2018 *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, Salt Lake City, 18-23 June 2018, 5010-5019. https://doi.org/10.1109/cvpr.2018.00526

[10] Lang, A.H., Vora, S., Caesar, H., Zhou, L., Yang, J. and Beijbom, O. (2019) PointPillars: Fast Encoders for Object Detection from Point Clouds. 2019 *IEEE/CVF Conference on Computer Vision and Pattern Recognition* (*CVPR*), Long Beach, 15-20 June 2019, 12689-12697. https://doi.org/10.1109/cvpr.2019.01298

[11] Riegler, G., Ulusoy, A.O. and Geiger, A. (2017) OctNet: Learning Deep 3D Representations at High Resolutions. 2017 *IEEE Conference on Computer Vision and Pattern Recognition* (*CVPR*), Honolulu, 21-26 July 2017, 6620-6629 https://doi.org/10.1109/cvpr.2017.701

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

[13] Li, Y., Bu, R., Sun, M., Wu, W., Di, X. and Chen, B. (2018) PointCNN: Convolution On X-Transformed Points. arXiv: 1801.07791.

[14] Wang, Y., Sun, Y., Liu, Z., Sarma, S.E., Bronstein, M.M. and Solomon, J.M. (2019) Dynamic Graph CNN for Learning on Point Clouds. *ACM Transactions on Graphics*, **38**, 1-12. https://doi.org/10.1145/3326362

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

[16] Pang, Y., Wang, W., Tay, F.E.H., Liu, W., Tian, Y. and Yuan, L. (2022) Masked Autoencoders for Point Cloud Self-Supervised Learning. In: Avidan, S., Brostow, G., Cissé, M., Farinella, G.M. and Hassner, T., Eds., *Computer Vision—ECCV* 2022, Springer, 604-621. https://doi.org/10.1007/978-3-031-20086-1_35

[17] Liu, Z., Lin, Y., Cao, Y., Hu, H., Wei, Y., Zhang, Z., *et al.* (2021) Swin Transformer: Hierarchical Vision Transformer Using Shifted Windows. 2021 *IEEE/CVF International Conference on Computer Vision* (*ICCV*), Montreal, 10-17 October 2021, 9992-10002. https://doi.org/10.1109/iccv48922.2021.00986

[18] Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., *et al.* (2021) An Image Is Worth 16 × 16 Words: Transformers for Image Recognition at Scale. arXiv: 2010.11929.

[19] Liu, Z., Yang, X., Tang, H., Yang, S. and Han, S. (2023) FlatFormer: Flattened Window Attention for Efficient Point Cloud Transformer. 2023 *IEEE/CVF Conference on Computer Vision and Pattern Recognition* (*CVPR*), Vancouver, 17-24 June 2023, 1200-1211. https://doi.org/10.1109/cvpr52729.2023.00122

[20] Zhao, H., Jiang, L., Fu, C. and Jia, J. (2019) PointWeb: Enhancing Local Neighborhood Features for Point Cloud Processing. 2019 *IEEE/CVF Conference on Computer Vision and Pattern Recognition* (*CVPR*), Long Beach, 15-20 June 2019, 5560-5568. https://doi.org/10.1109/cvpr.2019.00571

[21] Huang, H., Zhang, Y. and Ren, P. (2025) KernelDNA: Dynamic Kernel Sharing via Decoupled Naive Adapters. arXiv: 2503.23379.

[22] Komarichev, A., Zhong, Z. and Hua, J. (2019) A-CNN: Annularly Convolutional Neural Networks on Point Clouds. 2019 *IEEE/CVF Conference on Computer Vision and Pattern Recognition* (*CVPR*), Long Beach, 15-20 June 2019, 7413-7422. https://doi.org/10.1109/cvpr.2019.00760

[23] Ran, H., Liu, J. and Wang, C. (2022) Surface Representation for Point Clouds. arXiv: 2205.05740.

[24] Li, R., Li, X., Fu, C., Cohen-Or, D. and Heng, P. (2019) PU-GAN: A Point Cloud Upsampling Adversarial Network. 2019 *IEEE/CVF International Conference on Computer Vision* (*ICCV*), Seoul, 27 October-2 November 2019, 7202-7211. https://doi.org/10.1109/iccv.2019.00730

[25] Yu, L., Li, X., Fu, C., Cohen-Or, D. and Heng, P. (2018) PU-Net: Point Cloud Upsampling Network. 2018 *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, Salt Lake City, 18-23 June 2018, 2790-2799. https://doi.org/10.1109/cvpr.2018.00295

[26] Liu, R., Lehman, J., Molino, P., Such, F.P., Frank, E., Sergeev, A. and Yosinski, J. (2018) An Intriguing Failing of Convolutional Neural Networks and the CoordConv Solution. arXiv: 1807.03247.

[27] Wang, L., Huang, Y., Hou, Y., Zhang, S. and Shan, J. (2019) Graph Attention Convolution for Point Cloud Semantic Segmentation. 2019 *IEEE/CVF Conference on Computer Vision and Pattern Recognition* (*CVPR*), Long Beach, 15-20 June 2019, 10288-10297. https://doi.org/10.1109/cvpr.2019.01054

[28] Hu, J., Shen, L., Albanie, S., Sun, G. and Wu, E. (2019) Squeeze-and-Excitation Networks. arXiv: 1709.01507.

[29] Qi, C.R., Yi, L., Su, H. and Guibas, L.J. (2017) PointNet++: Deep Hierarchical Feature Learning on Point Sets in a Metric Space. arXiv: 1706.02413.

[30] Qian, G., Li, Y., Peng, H., Mai, J., Hammoud, H.A.A.K., Elhoseiny, M. and Ghanem, B. (2022) PointNeXt: Revisiting PointNet++ with Improved Training and Scaling Strategies. arXiv: 2206.04670.

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

[32] Armeni, I., Sener, O., Zamir, A.R., Jiang, H., Brilakis, I., Fischer, M., *et al.* (2016) 3D Semantic Parsing of Large-Scale Indoor Spaces. 2016 *IEEE Conference on Computer Vision and Pattern Recognition* (*CVPR*), Las Vegas, 27-30 June 2016, 1534-1543. https://doi.org/10.1109/cvpr.2016.170

[33] Tchapmi, L., Choy, C., Armeni, I., Gwak, J. and Savarese, S. (2017) SEGCloud: Semantic Segmentation of 3D Point Clouds. 2017 *International Conference on 3D Vision* (*3DV*), Qingdao, 10-12 October 2017, 537-547. https://doi.org/10.1109/3dv.2017.00067

[34] Rozenberszki, D., Litany, O. and Dai, A. (2022) Language-Grounded Indoor 3D Semantic Segmentation in the Wild. In: Avidan, S., Brostow, G., Cissé, M., Farinella, G.M. and Hassner, T., Eds., *Computer Vision—ECCV* 2022, Springer, 125-141. https://doi.org/10.1007/978-3-031-19827-4_8

[35] Lai, X., Liu, J., Jiang, L., Wang, L., Zhao, H., Liu, S., *et al.* (2022) Stratified Transformer for 3D Point Cloud Segmentation. 2022 *IEEE/CVF Conference on Computer Vision and Pattern Recognition* (*CVPR*), New Orleans, 18-24 June 2022, 8490-8499. https://doi.org/10.1109/cvpr52688.2022.00831

[36] Yang, Y.Q., Guo, Y.X., Xiong, J.Y., Liu, Y., Pan, H., Wang, P.S., Tong, X. and Guo, B. (2023) Swin3D: A Pretrained Transformer Backbone for 3D Indoor Scene Understanding. arXiv: 2304.06906.

[37] Dai, Z., Liu, H., Le, Q.V. and Tan, M. (2021) CoAtNet: Marrying Convolution and Attention for All Data Sizes. arXiv: 2106.04803.

[38] Zhou, Z., Rahman Siddiquee, M.M., Tajbakhsh, N. and Liang, J. (2018) UNet++: A Nested U-Net Architecture for Medical Image Segmentation. In: Stoyanov, D., *et al.*, Eds., *Deep Learning in Medical Image Analysis and Multimodal Learning for Clinical Decision Support*, Springer, 3-11. https://doi.org/10.1007/978-3-030-00889-5_1

[39] Huang, H., Lin, L., Tong, R., Hu, H., Zhang, Q., Iwamoto, Y., *et al.* (2020) UNet 3+: A Full-Scale Connected UNet for Medical Image Segmentation. *ICASSP* 2020—2020 *IEEE International Conference on Acoustics*, *Speech and Signal Processing* (*ICASSP*), Barcelona, 4-8 May 2020, 1055-1059. https://doi.org/10.1109/icassp40776.2020.9053405