Hybrid Position Encoding and Boundary-Aware Deep Supervision for 3D Point Cloud Semantic Segmentation

1st Yongyue Ma
Institute of Computing Technology, Chinese Academy of Sciences
Beijing, China
mayongyue23s@ict.ac.cn

2nd Shuqin Gao
Institute of Computing Technology, Chinese Academy of Sciences
Beijing, China
gaoshuqin@ict.ac.cn

3rd Yucheng Zhang
Institute of Computing Technology, Chinese Academy of Sciences
Beijing, China
zhangyucheng@ict.ac.cn

***Abstract*—3D point cloud semantic segmentation serves as a fundamental task for 3D scene understanding, whose performance directly influences decision-making in critical applications such as autonomous driving. Although advanced methods such as Point Transformer V3 have been proposed, they still face challenges in complex scenes, including low accuracy in fine-grained and small-object segmentation, inadequate position encoding, and limited capability in local-global context modeling. To tackle these issues, this paper presents a novel framework for 3D point cloud semantic segmentation that integrates supervision enhancement and feature enhancement. Specifically, we design a Boundary-Aware Deep Supervision (BADS) module to strengthen fine-grained feature learning and refine the quality of the supervision in representative regions. Meanwhile, we propose a Hybrid Position Encoding (HPE) strategy that effectively mitigates the deficiency of position encoding by fusing multi-scale HashGrid absolute encoding and multi-axis Rotary Position Embedding (RoPE). Experiments on ScanNet V2 and S3DIS demonstrate that our method achieves 78.1% mIoU on ScanNet V2 and 74.4% mIoU on S3DIS Area-5, outperforming Point Transformer V3 and other state-of-the-art approaches, especially on fine-grained and small-object categories. Ablation studies validate the effectiveness of both BADS and HPE individually and synergistically, offering a new technical solution for 3D point cloud semantic segmentation in complex scenes.**

***Index Terms*—Deep Supervision, Hybrid Position Encoding, Boundary-Aware, 3D Point Cloud Semantic Segmentation**

I. INTRODUCTION

The core goal of 3D point cloud semantic segmentation is to assign precise semantic labels to unordered, sparse, and noisy 3D point cloud data. As a fundamental task in autonomous driving, robotic navigation, smart agriculture, and industrial quality inspection, its performance directly determines the decision-making reliability of downstream 3D vision systems. Unlike 2D images with a regular grid structure, point cloud data inherently exhibits unorderedness, sparsity, and density heterogeneity, posing two fundamental challenges: (1) collaborative modeling of local fine-grained features and global context remains difficult; (2) insufficient feature discriminability between fine-grained categories and small objects often leads to missed detections or blurred boundaries.

Point cloud segmentation methods fall into two broad categories: traditional approaches and deep learning-based methods. Traditional methods rely on handcrafted geometric rules and statistical features (e.g., thresholding, K-means/DBSCAN clustering, region growing). While highly interpretable, they lack robustness to noise and complex topological structures. Deep learning methods automatically learn discriminative features via neural networks and have become the dominant paradigm. Existing deep learning approaches can be further grouped into two primary paradigms, both grappling with the fundamental accuracy-efficiency trade-off in complex large-scale scenes.

- **Convolution-based models** (e.g., PointNet++ [1], KPConv [2]) are computationally efficient with linear $O(N)$ complexity. However, their inherent locality limits long-range dependency modeling, often causing omission of small objects and blurred boundary segmentation.
- **Attention-based models** (e.g., the Point Transformer series [3]) enable long-range global context modeling via self-attention. Among them, Point Transformer V3 [4] achieves major efficiency and performance gains through space-filling curves and sequential encoding (delivering $3.3\times$ faster inference and $10.2\times$ lower memory footprint than V2). Nevertheless, it suffers from three critical limitations: (a) imbalanced local–global feature representation, degrading segmentation accuracy for fine-grained categories and small objects; (b) limited positional encoding, reducing robustness to rotation, scaling and translation transformations; (c) no effective deep supervision mechanism with adaptive label reliability strategies for low-resolution intermediate layers.

Existing state-of-the-art methods only address these challenges in isolation: HybridTM [5] balances local–global modeling via a Transformer-Mamba hybrid yet retains the original single CPE, leading to poor geometric robustness; PointMamba [6] eliminates Transformer overhead via a pure SSM design but suffers from degraded accuracy on fine-grained categories and small objects.

To address this gap, this paper proposes a 3D point cloud semantic segmentation framework integrating Boundary-Aware Deep Supervision (BADS) and Hybrid Position Encoding (HPE). Our main contributions are threefold:

1) We design a BADS framework that constructs multi-scale auxiliary supervision branches in intermediate decoder layers. Through clustering consistency evaluation and adaptive weight adjustment, it dynamically modulates supervision intensity based on intra-cluster label purity, focusing on enhancing feature learning for fine-grained categories and small objects, and improving supervision quality for label-reliable regions in low-resolution intermediate layers.
2) We propose an HPE strategy that fuses multi-scale HashGrid absolute positional encoding [7] with multi-axis Rotary Position Embedding (RoPE) relative positional encoding [8]. By jointly modeling complementary absolute and relative spatial relationships, HPE addresses the limitation of insufficient positional encoding, enhances robustness to geometric transformations, and enhances local-global context integration.
3) We present a unified end-to-end segmentation framework that seamlessly integrates the BADS module and HPE strategy into the core architecture of Point Transformer V3. It retains the high efficiency of Point Transformer V3 while enabling targeted optimization of fine-grained categories and small object segmentation. Experimental results on ScanNet V2 [9] and S3DIS [10] show that our method outperforms Point Transformer V3 and other state-of-the-art baselines, validating its effectiveness and practicality for 3D point cloud semantic segmentation in complex scenes.

II. RELATED WORK

A. Model Architectures

Deep learning has transformed point cloud processing from manual feature engineering to end-to-end learning paradigms, with mainstream architectures evolving along four key directions to address the inherent challenges of unstructured data. Early direct point processing methods laid the foundational groundwork for this field: Qi et al. proposed PointNet [11], the first framework enabling unified feature extraction for unordered point sets, though it lacked effective local context aggregation. This limitation was addressed by Qi et al. in PointNet++ [1], which introduced multi-scale grouping and hierarchical sampling strategies, establishing itself as a de facto baseline for subsequent point cloud research. Convolution-based models enhance local feature capture through point-adaptive kernel designs. Thomas et al. proposed KPConv [2], introducing deformable convolution kernels to adapt to the geometric distribution of point clouds. Li et al. presented PointCNN [12], leveraging permutation-invariant convolutions to model intricate inter-point relationships. For large-scale point cloud processing, Hu et al. proposed RandLA-Net [13], optimizing efficiency via random sampling and local feature aggregation to balance computational cost and segmentation accuracy. Attention-based models break the locality constraint of convolutional operations to strengthen global context modeling. The Point Transformer series originated from the work of Zhao et al. [3], which first introduced offset attention mechanisms. Subsequent variants proposed by Wu et al. further incorporated grouped vector attention and space-filling curve-based serialization for efficient large-scale processing. However, these models suffer from imbalanced local-global feature representation, which degrades segmentation performance for fine-grained categories and small objects. Hybrid architectures integrating Transformers and state space models (Mamba) have emerged to balance accuracy and efficiency. PointMamba [6] adopts a pure SSM-based single architecture (no Transformer components) with octree-based serialization, achieving linear complexity global-local modeling, while HybridTM [5] (Wang et al.) uses an inner-layer hybrid strategy (attention for local features, Mamba for global semantics). However, both still struggle with subtle boundary detail capture in complex scenes, leaving room for optimization.

B. Positional Encoding

Accurate modeling of positional information is critical for 3D point cloud segmentation, as it directly governs the model's ability to perceive spatial topology and the robustness of learned feature representations. Existing approaches can be broadly classified into three categories, each with notable limitations. Relative Position Encoding (RPE) quantifies geometric relationships by constructing local neighborhoods (e.g., K-Nearest Neighbors, Ball Query) and computing pairwise point distances. Although it supplies fine-grained spatial cues, its O(NlogN) computational complexity hinders real-time processing of large-scale point clouds. Conditional Position Encoding (CPE) and its improved variant xCPE (enhanced Conditional Position Encoding) rely on octree division to achieve lightweight absolute positional encoding with linear O(N) complexity. However, drawbacks such as fixed grid resolution and decoupled absolute-relative positional information prevent them from simultaneously capturing local details and global structures. More importantly, existing research still focuses on optimizing single-type encoding schemes, leading to insufficient collaborative modeling of absolute and relative positions, a core bottleneck restricting further performance gains.

Notably, the strong advantages of cross-domain adaptive methods in their original fields provide new opportunities to overcome this bottleneck, yet none have been adapted for positional encoding in 3D point clouds. HashGrid encoding, proposed by Müller et al. at SIGGRAPH 2022 [7], originates from computer graphics for neural rendering. By discretizing 3D space with multi-scale hash grids, it reduces memory overhead by an order of magnitude, but has not been applied to positional modeling for point clouds. Similarly, Rotary Position Embedding (RoPE), introduced by Su et al. [8] for NLP sequence modeling, encodes relative relationships implicitly through rotation transformations without extra learn-

able parameters, but has not been tailored for spatial feature representation in point cloud data.

C. Deep Supervision Mechanisms

Deep supervision, which injects supervision signals at multiple network layers to alleviate gradient vanishing and enhance feature learning, has been widely adopted in 3D point cloud segmentation. Nevertheless, mainstream paradigms based on "hard labels + fixed weights" are ill-suited to the intrinsic characteristics of point clouds (irregular topology, sparse distribution, blurred semantic boundaries), leading to three critical limitations of traditional deep supervision paradigms. First, auxiliary supervision branches lack ambiguity-aware design. Traditional point cloud segmentation frameworks utilize unified hard label constraints in auxiliary branches, disregarding the gradual label transition characteristics in category-interleaved boundary regions, which induces semantic misclassification. Second, supervision weights exhibit insufficient dynamic adaptability. Approaches like AMContrast3D [14], which relies on a single ambiguity metric, and DyCo3D [15], which requires manual parameter tuning, fail to adapt to scene-specific variations in boundary topology and regional semantic complexity, resulting in inadequate supervision of key structural details. Third, the synergy between noise robustness and gradient transmission is inadequate. Classic models such as RandLA-Net [13] suffer from gradient dilution in sparse boundary regions, while methods including AMContrast3D [14] lack mechanism-level noise-resistant architectures, rendering them vulnerable to annotation noise and scanning interference.

III. METHOD

A. Overall Framework

We propose a symmetric encoder-decoder architecture with Point Transformer V3 (PTV3) [4] as the backbone, which seamlessly integrates the Boundary-Aware Deep Supervision (BADS) framework and Hybrid Position Encoding (HPE) strategy to form a synergistic optimization loop defined as "feature enhancement and supervision reinforcement". The backbone inherits the core modules of PTV3, including the sequential encoding mechanism and global attention modeling. Specifically, by employing space-filling curves (e.g., Z-order, Hilbert curves) and the sequential encoding mechanism, the network converts unordered point clouds into ordered sequences to support structured processing. It exhibits inherent advantages in large-scale point cloud processing and achieves significant performance improvements over existing methods in both inference speed and memory efficiency.

The encoder performs progressive downsampling on the input through five hierarchical stages (Stage0–Stage4). Each stage consists of multiple Point Transformer blocks [4], where the HPE strategy is embedded to inject multi-dimensional positional information, namely HashGrid-based absolute encoding [7] and RoPE-based relative encoding [8]. The decoder adopts a mirrored symmetric structure with four upsampling stages (Stage3–Stage0), which are connected to the corresponding encoder stages via skip connections to retain fine-grained spatial details. The BADS framework is integrated into the intermediate decoder stages (Stage2 and Stage1), where auxiliary supervision heads conduct clustering consistency evaluation and adaptive weight tuning to strengthen the supervision for fine-grained categories and small objects. The final output of Stage0 is sent to the main segmentation head, while the intermediate outputs of Stage2 and Stage1 are passed to the auxiliary supervision heads, enabling adaptive multi-scale supervision that focuses on high-confidence regions.

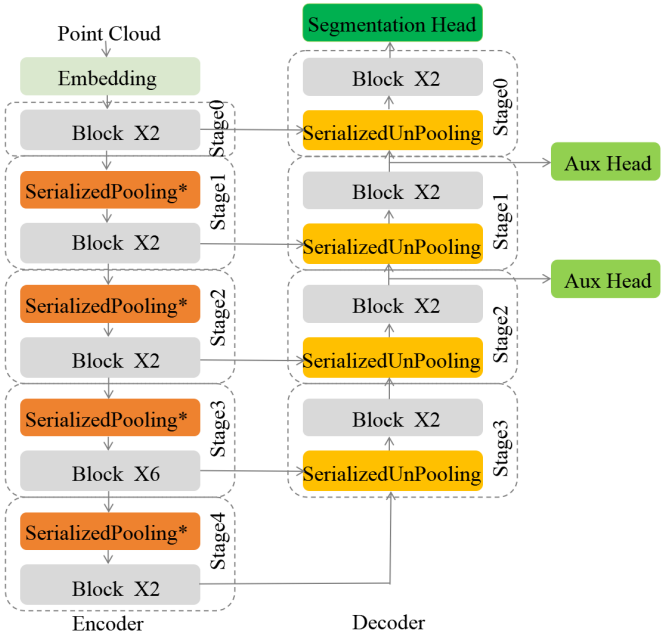

Fig. 1: Overall architecture of the proposed network.

B. Boundary-Aware Deep Supervision (BADS) Framework

To address the limitations of traditional deep supervision mechanisms, including unreliable intermediate labels in low-resolution layers, fixed supervision weights, and gradient vanishing, the Boundary-Aware Deep Supervision (BADS) framework achieves differentiated and precise supervision for label-reliable regions via a hierarchical supervision structure, consistency evaluation, and adaptive weight tuning. It focuses on enhancing feature learning for fine-grained categories and small-scale object instances.

Auxiliary supervision branches are added at intermediate decoder stage 2 (resolution 1/4, rich semantic information) and stage 1 (resolution 1/2, clear spatial details), with lightweight linear layers introducing negligible additional computational overhead.

1) Label Matching Strategy: During the encoder's SerializedPooling operation, point clouds are aggregated into superpoints (clusters) via downsampling. To match label and feature resolutions for auxiliary supervision, a majority voting mechanism maps point-level labels to superpoint-level labels.

For cluster $\mathcal{C}_j$ (corresponding to superpoint j), its aggregated label is defined as:

$$\hat{y}_j = \arg\max_k \sum_{p_i \in \mathcal{P}_j} \mathbb{1}[y_i = k] \cdot \mathbb{1}[y_i \geq 0] \tag{1}$$

where $\mathcal{P}_j$ denotes the original point set of cluster $\mathcal{C}_j$, $y_i \in \{0, 1, \ldots, K-1, -1\}$ is the semantic label of point p_i (label value -1 indicates occluded invalid labels), and $\mathbb{1}[\cdot]$ is the indicator function. This strategy effectively reduces interference from original label noise while preserving the statistical properties of the label distribution.

2) Consistency Evaluation and Adaptive Weight Tuning: To quantify the label reliability within each cluster, consistency scores are computed synchronously during the encoder's SerializedPooling operation, introducing no additional computational overhead. For cluster $\mathcal{C}_j$, its consistency score is defined as:

$$c_j = \frac{\max_{k \in \{1,2,\ldots,K\}} v_{j,k}}{n_j}, \tag{2}$$

where $n_j = |\mathcal{P}_j|$ is the number of original points in cluster $\mathcal{C}_j$, and $v_{j,k} = \sum_{p_i \in \mathcal{P}_j} \mathbb{1}[y_i = k]$ is the number of points in cluster $\mathcal{C}_j$ belonging to category k.

The consistency score $c_j \in [0, 1]$ reflects the label reliability within the cluster. When c_j approaches 1.0, all points in the cluster belong to the same category (reliable labels, e.g., complete small objects or fine-grained categories). When c_j is low, the cluster contains points from multiple categories (unreliable labels, e.g., boundary-mixed regions).

Based on consistency scores, a stage-wise adaptive weight adjustment mechanism is designed, with the mathematical expression:

$$w_s^{\text{adaptive}} = w_s \cdot \bar{c}_s, \tag{3}$$

where w_s is the base weight for stage s ($w_{\text{dec2}} = 0.1$, $w_{\text{dec1}} = 0.2$), and $\bar{c}_s = \frac{1}{M_s} \sum_{j=1}^{M_s} c_j$ is the overall consistency score of the current sample at stage s (M_s is the total number of clusters at stage s).

The core idea of this mechanism is: When the proportion of label-reliable regions is higher ($\bar{c}_s$ is larger), supervision weights are increased to strengthen learning for fine-grained categories and small-scale object instances; when the proportion of label-unreliable regions is higher ($\bar{c}_s$ is smaller), supervision weights are reduced to avoid excessive interference from noisy labels during model training.

3) Total Loss Function: The multi-branch joint constraint loss function is defined as:

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{main}} + \sum_{s \in \{\text{dec2}, \text{dec1}\}} w_s^{\text{adaptive}} \cdot \mathcal{L}_s^{\text{aux}}, \tag{4}$$

where $\mathcal{L}_{\text{main}}$ is the main branch loss, defined as a weighted sum of cross-entropy loss and Lovász loss:

$$\mathcal{L}_{\text{main}} = \mathcal{L}_{\text{CE}} + \lambda_{\text{Lov}} \cdot \mathcal{L}_{\text{Lov}}, \quad \lambda_{\text{Lov}} = 1.0, \tag{5}$$

where $\lambda_{\text{Lov}} = 1.0$ is the weight coefficient for the Lovász loss. The cross-entropy loss $\mathcal{L}_{\text{CE}}$ ensures the overall precision of segmentation, while the Lovász loss $\mathcal{L}_{\text{Lov}}$ mitigates class imbalance through submodular optimization. $\mathcal{L}_s^{\text{aux}}$ denotes the cross-entropy loss of auxiliary branch s, which is used to optimize the feature learning effect at the corresponding stage.

Through adaptive weight w_s^{adaptive} adjustment, this loss function ensures that the model focuses on feature learning in label-reliable regions, effectively improving segmentation accuracy for fine-grained categories and small-scale object instances.

C. Hybrid Position Encoding (HPE)

The Hybrid Position Encoding (HPE) strategy fuses HashGrid encoding [7], Rotary Position Embedding (RoPE) [8], and enhanced Conditional Position Encoding (xCPE) for comprehensive spatial information modeling. These three mechanisms are complementary, operate at distinct network levels, and are fully consistent with PTV3's [4] design logic. Specifically, HashGrid, originally proposed for neural rendering tasks in computer graphics to efficiently represent continuous 3D space for view synthesis, is not simply reused here. Instead, we recalibrate its hash grid resolution and embedding dimension to adapt to the sparse and discrete characteristics of 3D point clouds, enabling it to prioritize encoding semantically sensitive regions for segmentation tasks. We integrate it as the core of absolute positional encoding in HPE, injecting absolute positional bias at the feature level.

For RoPE, initially designed for NLP text sequence modeling to implicitly encode relative positional relationships between words in 1D sequences via rotation transformations, we migrate it from the 1D sequence domain to 3D point cloud processing by extending its rotation logic to three-dimensional space, matching the geometric properties of point clouds. Deployed at the attention level of HPE, it modulates relative positional relationships between points, breaking the single paradigm of only cooperating with word embeddings in NLP. Finally, xCPE further reinforces local spatial structure modeling via sparse convolution at the convolution level. Together, the three modules form a hierarchical complementary system, realizing comprehensive representation of 3D spatial information, and we achieve seamless integration with the PTV3 Transformer backbone through a residual dimension alignment layer.

1) HashGrid Multi-Scale Position Encoding: HashGrid [7] employs a multi-resolution hash table encoding scheme to represent absolute positional information in 3D space. By replacing explicit grid structures with hash-based indexing, HashGrid simultaneously captures fine-grained local details and global spatial structures while maintaining low memory overhead, making it particularly suitable for large-scale point cloud processing.

HashGrid injects multi-scale absolute positional information into feature representations through learnable hash table embeddings and residual connections, complementing relative encoding:

- For a 3D point $\mathbf{p} = (x, y, z)$, query L hash tables at different resolutions, extract feature vectors, and fuse them.

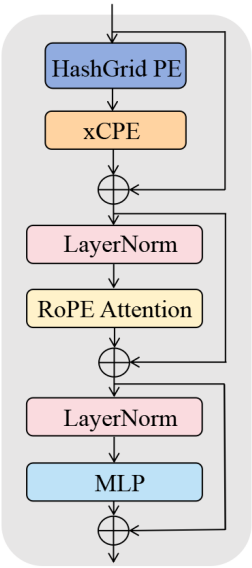

Fig. 2: Detail of Block with HPE Strategy.

- Resolution configuration: $N_l = N_{\text{min}} \cdot b^l$ ($L = 8$, $N_{\text{min}} = 24$, $b = 1.4$ for ScanNet V2 [9]; $L = 12$, $N_{\text{min}} = 32$, $b = 1.5$ for S3DIS [10]).
- Spatial hashing maps grid indices $\mathbf{g}_l = \lfloor \mathbf{p} \cdot N_l \rfloor$ to hash table entries, with learnable embeddings initialized as $\mathcal{U}(-0.0005, 0.0005)$.
- Feature fusion: Concatenate L-layer features and project to match feature dimension, then add to original features via residual connection:

$$\mathbf{f}_i^{\text{hash}} = \mathbf{f}_i + \mathbf{W}_{\text{proj}}[\mathbf{h}_0, \mathbf{h}_1, \ldots, \mathbf{h}_{L-1}] \tag{6}$$

- Applied before the xCPE layer in each Point Transformer block.

2) RoPE Rotational Multi-Axis Position Encoding: RoPE [8] injects 3D relative positional information into self-attention without additional parameters, complementing absolute encoding:

- Only modulates query ($\mathbf{q}$) and key ($\mathbf{k}$) vectors during attention computation:

$$\text{Attn}(\mathbf{q}, \mathbf{k}, \mathbf{v}) = \text{softmax}\left(\frac{\mathbf{q}'\mathbf{k}'^{\top}}{\sqrt{D}}\right)\mathbf{v} \tag{7}$$

  where $\mathbf{q}'$ and $\mathbf{k}'$ are rotated Q/K vectors, D is the attention head dimension.
- Frequency vector: $\theta_i = \text{rope\_base}^{-i/(D/2)}$ ($\text{rope\_base} = 10000$), multi-axis phase:

$$\boldsymbol{\phi} = \frac{1}{3}(x\boldsymbol{\theta} + y\boldsymbol{\theta} + z\boldsymbol{\theta}) \tag{8}$$

- Rotation operation:
  The rotated query and key vectors are computed as:

$$\mathbf{q}' = \mathbf{q} \odot \text{cos\_full} + \text{rotate\_half}(\mathbf{q}) \odot \text{sin\_full}, \tag{9}$$

$$\mathbf{k}' = \mathbf{k} \odot \text{cos\_full} + \text{rotate\_half}(\mathbf{k}) \odot \text{sin\_full} \tag{10}$$

  where the cosine/sine vectors are defined as:

$$\text{cos\_full} = [\cos(\boldsymbol{\phi}), \cos(\boldsymbol{\phi})] \in \mathbb{R}^D, \tag{11}$$

$$\text{sin\_full} = [\sin(\boldsymbol{\phi}), \sin(\boldsymbol{\phi})] \in \mathbb{R}^D \tag{12}$$

  and the half-rotation operator is defined as:

$$\text{rotate\_half}(\mathbf{x}) = [-\mathbf{x}_b, \mathbf{x}_a] \tag{13}$$

  for any vector $\mathbf{x} = [\mathbf{x}_a, \mathbf{x}_b] \in \mathbb{R}^D$ with $\mathbf{x}_a, \mathbf{x}_b \in \mathbb{R}^{D/2}$.

Our adapted RoPE preserves relative geometric relationships under rotation and translation transformations, with negligible computational overhead (only two element-wise trigonometric operations and one half-rotation operation per attention head).

IV. EXPERIMENTS

A. Experimental Setup

To comprehensively validate the effectiveness and generalization of the proposed method for point cloud semantic segmentation, extensive experiments are conducted on two authoritative benchmark datasets. The detailed experimental configurations are described as follows:

- **Datasets:**
  - *ScanNet V2* [9]: A widely-used indoor scene point cloud dataset containing 1,513 scans in total, including 1,201 training scenes, 312 validation scenes, and 100 test scenes. It covers 20 semantic categories (e.g., walls, floors, furniture) with diverse scene layouts and point density variations, serving as a common benchmark for evaluating indoor point cloud segmentation models.
  - *S3DIS* [10]: A large-scale architectural scene dataset collected from 6 regions (Area-1 to Area-6) of three buildings, covering 13 semantic categories (e.g., ceilings, columns, boards). Following the standard protocol, Area-5 is adopted as the test set, and the remaining areas are used for training to ensure fair comparisons.
- **Implementation Details:** Our model is implemented in Python 3.10 with PyTorch 2.1.2 (torchvision 0.16.2) and accelerated by CUDA 11.8 and cuDNN 8.9.2. All experiments are conducted on a workstation with two NVIDIA H20-NVLink GPUs (96 GB per GPU), an Intel Xeon Platinum 8457C CPU (40 vCPUs, 2.90 GHz), and 256 GB DDR5 RAM. We adopt the AdamW optimizer with a weight decay of 0.05. The initial learning rate is set to 0.006, optimized by the OneCycleLR scheduler with cosine annealing (decaying from 0.006 to 0.0006). The model is trained for 800 epochs with a batch size of 12 on both ScanNet V2 and S3DIS. Data augmentation strategies include random scaling (0.9–1.1×), random rotation around the z-axis (± 1 radian, $\pm 57°$), minor random rotations around the x and y axes ($\pm 1/64$ radian), and random point dropout with a rate of 0.2.

- **Evaluation Metrics:**
  - *Mean Intersection over Union (mIoU)*: Average IoU across all semantic categories.
  - *Mean Accuracy (mAcc)*: Average per-category classification accuracy.
  - *Overall Accuracy (allAcc)*: Percentage of correctly classified points across all categories.

B. Quantitative Result Comparison

Extensive quantitative results on standard benchmarks demonstrate the superiority of our method (Tables I–II).

Figs. 3 and 4 present qualitative comparisons on the S3DIS validation set. The proposed method achieves more accurate segmentation for boundary regions and small objects, reducing ambiguous segmentations attributed to the HPE strategy and BADS framework.

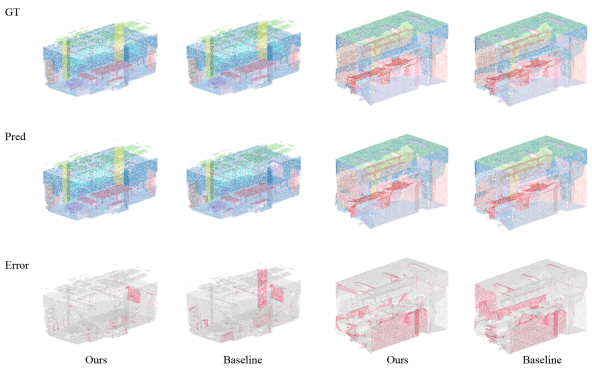

Fig. 3: Qualitative comparison on S3DIS validation set (a): From top to bottom are Ground Truth (GT), prediction results (Pred), and error regions (red denotes misclassified points).

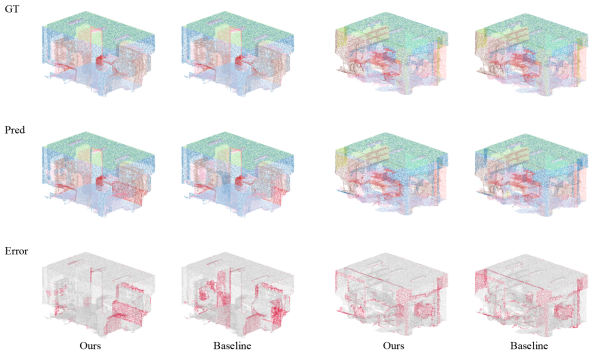

Fig. 4: Qualitative comparison on S3DIS validation set (b): From top to bottom are Ground Truth (GT), prediction results (Pred), and error regions (red denotes misclassified points).

TABLE I: Semantic Segmentation Results (mIoU %) on S3DIS Area-5

Model Name	Year	mIoU (%)
PointNet [11]	2017	41.1
SegCloud [16]	2017	48.9
TanConv [17]	2018	52.6
PointCNN [12]	2018	57.3
ParamConv [18]	2018	58.3
PointWeb [19]	2019	60.3
HPEIN [20]	2019	61.9
KPConv [2]	2019	67.1
GACNet [21]	2019	62.9
PAT [22]	2019	60.1
SPGraph [23]	2018	58.0
SegGCN [24]	2020	63.6
PAConv [25]	2021	66.6
PointNeXt [26]	2022	70.5
PointMetaBase [27]	2023	72.0
Swin3D [28]	2023	72.5
MinkUNet [29]	2019	65.4
MinkUNet+PC [30]	2020	70.3
MinkUNet+CSC [31]	2021	72.2
MinkUNet+MSC [32]	2023	70.1
MinkUNet+GC [33]	2024	72.0
MinkUNet+PPT [34]	2024	72.7
PTv1 [3]	2021	70.4
PTv2 [35]	2022	71.6
PTv3 [4]	2024	73.4
HybridTM [5]	2025	72.1
Ours	**2026**	**74.4**

Table I presents the mIoU comparison on S3DIS Area-5. Our method achieves an mIoU of 74.4%, outperforming PTv3 (73.4%) and HybridTM (72.1%).

TABLE II: Semantic Segmentation Results (mIoU %) on ScanNet V2

Model Name	Year	mIoU (%)
PointNet++ [1]	2017	53.5
SparseConvNet [36]	2018	69.3
PointConv [37]	2019	61.0
JointPointBased [38]	2019	69.2
KPConv [2]	2019	69.2
PointASNL [39]	2020	63.5
PointNeXt [26]	2022	71.5
LargeKernel3D [40]	2023	73.5
PointMetaBase [27]	2023	72.8
Swin3D [28]	2023	77.5
MinkUNet [29]	2019	72.2
MinkUNet+PC [30]	2020	74.1
MinkUNet+CSC [31]	2021	73.8
MinkUNet+MSC [32]	2023	75.5
MinkUNet+GC [33]	2024	75.7
MinkUNet+PPT [34]	2024	76.4
OA-CNNs [41]	2024	76.1
PTv1 [3]	2021	70.6
PTv2 [35]	2022	75.4
PTv3 [4]	2024	77.5
HybridTM [5]	2025	77.8
Ours	**2026**	**78.1**

Table II presents the semantic segmentation results on the ScanNet V2 dataset. Our method achieves an mIoU of 78.1%, outperforming PTv3 (77.5%) and HybridTM (77.8%) by 0.6 and 0.3 percentage points, respectively, which further validates the superior performance of our method.

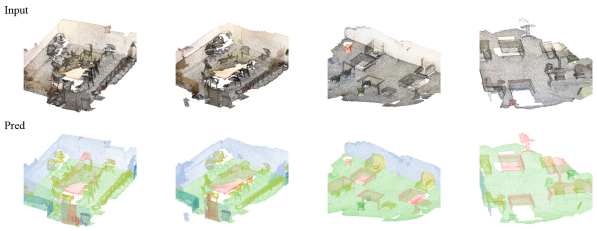

Fig. 5: Semantic segmentation results on ScanNet V2 (a).

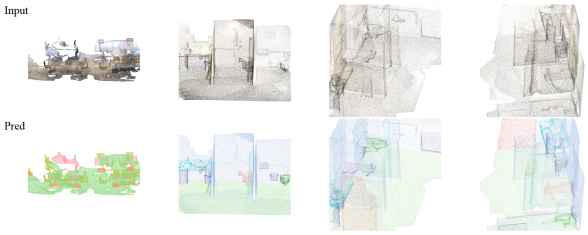

Fig. 6: Semantic segmentation results on ScanNet V2 (b).

C. Ablation Experiments

Ablation experiments (Tables III–IV) validate component effectiveness (baseline: PTv3).

TABLE III: Ablation experiment results on S3DIS dataset

Added Components	mIoU (%)	mAcc (%)	allAcc (%)
Baseline	73.40	78.50	91.80
+ HPE	73.99	78.10	92.00
+ BADS	74.32	78.90	92.06
+ BADS + HPE	74.38	79.04	92.19

TABLE IV: Ablation experiment results on ScanNet V2 dataset

Added Components	mIoU (%)	mAcc (%)	allAcc (%)
Baseline	77.50	84.50	92.20
+ BADS	77.86	85.04	92.10
+ HPE	78.00	85.09	92.15
+ BADS + HPE	78.10	85.20	92.30

Ablation insights: Ablation experiments validate the efficacy of the proposed Hybrid Position Encoding (HPE) and Boundary-Aware Deep Supervision (BADS). HPE alone improves mIoU by 0.5–0.6% via enhanced spatial modeling, while BADS contributes an additional 0.4–0.9% mIoU by refining supervision on boundaries and small objects. Visual results on the column class of S3DIS (Fig. 7) corroborate

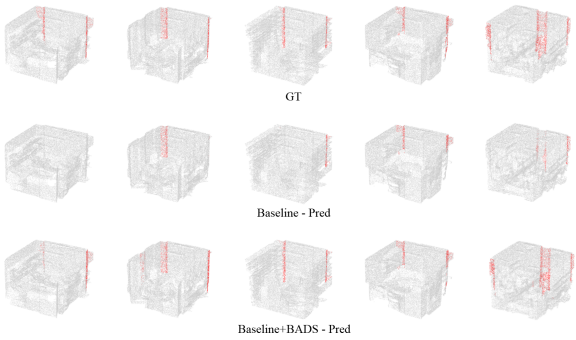

Fig. 7: Segmentation visualization of the *column* category (red points) on S3DIS: Top row = Ground Truth (GT), middle row = baseline predictions (without BADS), bottom row = our predictions (with BADS).

these gains. Combining both modules yields a total 1.0% mIoU improvement, revealing their complementary nature in a "feature enhancement–supervision reinforcement" paradigm that jointly improves segmentation performance.

V. CONCLUSION

This paper tackles the critical challenges of insufficient boundary segmentation accuracy and weak spatial representation robustness in modern 3D point cloud semantic segmentation methods. Built upon the Point Transformer V3 (PTV3) backbone, we propose a unified segmentation framework that integrates Boundary-Aware Deep Supervision (BADS) and Hybrid Position Encoding (HPE). The BADS module realizes adaptive and fine-grained supervision for boundary regions via clustering consistency evaluation and dynamic weight tuning, which effectively mitigates label ambiguity and alleviates gradient vanishing during training. As a complementary component, the HPE strategy fuses multi-scale absolute positional encoding from HashGrid and multi-axis relative positional encoding from Rotary Position Embedding (RoPE), enabling comprehensive spatial modeling and improving the robustness of feature representations against geometric variations. Extensive experimental results show that our approach surpasses several representative state-of-the-art methods, including the original PTV3. Ablation studies verify the effectiveness of each proposed module and their mutual reinforcement, which sheds light on the design of high-performance and boundary-aware 3D semantic segmentation models.

VI. ACKNOWLEDGEMENTS

Supported by the science and technology projects of the Ministry of Agriculture and Rural Affairs of China.

REFERENCES

[1] C. R. Qi, L. Yi, H. Su, and L. J. Guibas, "Pointnet++: Deep hierarchical feature learning on point sets in a metric space," *Advances in neural information processing systems*, vol. 30, 2017.

[2] H. Thomas, C. R. Qi, J.-E. Deschaud, B. Marcotegui, F. Goulette, and L. J. Guibas, "Kpconv: Flexible and deformable convolution for point clouds," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2019, pp. 6411–6420.

[3] H. Zhao, L. Jiang, J. Jia, P. H. Torr, and V. Koltun, "Point transformer," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2021, pp. 16 259–16 268.

[4] X. Wu, L. Jiang, P.-S. Wang, Z. Liu, X. Liu, Y. Qiao, W. Ouyang, T. He, and H. Zhao, "Point transformer v3: Simpler faster stronger," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2024, pp. 4840–4851.

[5] X. Wang, J. Hou, Z. Liu, and Y. Zhu, "Hybridtm: Combining transformer and mamba for 3d semantic segmentation," *arXiv preprint arXiv:2507.18575*, 2025.

[6] J. Liu, R. Yu, Y. Wang, Y. Zheng, T. Deng, W. Ye, and H. Wang, "Point mamba: A novel point cloud backbone based on state space model with octree-based ordering strategy," *arXiv preprint arXiv:2403.06467*, 2024.

[7] T. Müller, A. Evans, C. Schied, and A. Keller, "Instant neural graphics primitives with a multiresolution hash encoding," *ACM transactions on graphics (TOG)*, vol. 41, no. 4, pp. 1–15, 2022.

[8] J. Su, M. Ahmed, Y. Lu, S. Pan, W. Bo, and Y. Liu, "Roformer: Enhanced transformer with rotary position embedding," *Neurocomputing*, vol. 568, p. 127063, 2024.

[9] A. Dai, A. X. Chang, M. Savva, M. Halber, T. Funkhouser, and M. Nießner, "Scannet: Richly-annotated 3d reconstructions of indoor scenes," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 5828–5839.

[10] I. Armeni, O. Sener, A. R. Zamir, H. Jiang, I. Brilakis, M. Fischer, and S. Savarese, "3d semantic parsing of large-scale indoor spaces," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 1534–1543.

[11] C. R. Qi, H. Su, K. Mo, and L. J. Guibas, "Pointnet: Deep learning on point sets for 3d classification and segmentation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 652–660.

[12] Y. Li, R. Bu, M. Sun, W. Wu, X. Di, and B. Chen, "Pointcnn: Convolution on x-transformed points," *Advances in neural information processing systems*, vol. 31, 2018.

[13] Q. Hu, B. Yang, L. Xie, S. Rosa, Y. Guo, Z. Wang, N. Trigoni, and A. Markham, "Randla-net: Efficient semantic segmentation of large-scale point clouds," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 11 108–11 117.

[14] Y. Chen, Y. Duan, R. Zhang, and Y.-P. Tan, "Adaptive margin contrastive learning for ambiguity-aware 3d semantic segmentation," in *2024 IEEE International Conference on Multimedia and Expo (ICME)*. IEEE, 2024, pp. 1–6.

[15] T. He, C. Shen, and A. Van Den Hengel, "Dyco3d: Robust instance segmentation of 3d point clouds through dynamic convolution," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2021, pp. 354–363.

[16] L. Tchapmi, C. Choy, I. Armeni, J. Gwak, and S. Savarese, "Segcloud: Semantic segmentation of 3d point clouds," in *2017 international conference on 3D vision (3DV)*. IEEE, 2017, pp. 537–547.

[17] M. Tatarchenko, J. Park, V. Koltun, and Q.-Y. Zhou, "Tangent convolutions for dense prediction in 3d," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 3887–3896.

[18] S. Wang, S. Suo, W.-C. Ma, A. Pokrovsky, and R. Urtasun, "Deep parametric continuous convolutional neural networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 2589–2597.

[19] H. Zhao, L. Jiang, C.-W. Fu, and J. Jia, "Pointweb: Enhancing local neighborhood features for point cloud processing," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019, pp. 5565–5573.

[20] L. Jiang, H. Zhao, S. Liu, X. Shen, C.-W. Fu, and J. Jia, "Hierarchical point-edge interaction network for point cloud semantic segmentation," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2019, pp. 10 433–10 441.

[21] L. Wang, Y. Huang, Y. Hou, S. Zhang, and J. Shan, "Graph attention convolution for point cloud semantic segmentation," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019, pp. 10 296–10 305.

[22] J. Yang, Q. Zhang, B. Ni, L. Li, J. Liu, M. Zhou, and Q. Tian, "Modeling point clouds with self-attention and gumbel subset sampling," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019, pp. 3323–3332.

[23] L. Landrieu and M. Simonovsky, "Large-scale point cloud semantic segmentation with superpoint graphs," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 4558–4567.

[24] H. Lei, N. Akhtar, and A. Mian, "Seggcn: Efficient 3d point cloud segmentation with fuzzy spherical kernel," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 11 611–11 620.

[25] M. Xu, R. Ding, H. Zhao, and X. Qi, "Paconv: Position adaptive convolution with dynamic kernel assembling on point clouds," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2021, pp. 3173–3182.

[26] G. Qian, Y. Li, H. Peng, J. Mai, H. Hammoud, M. Elhoseiny, and B. Ghanem, "Pointnext: Revisiting pointnet++ with improved training and scaling strategies," *Advances in neural information processing systems*, vol. 35, pp. 23 192–23 204, 2022.

[27] H. Lin, X. Zheng, L. Li, F. Chao, S. Wang, Y. Wang, Y. Tian, and R. Ji, "Meta architecture for point cloud analysis," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2023, pp. 17 682–17 691.

[28] Y.-Q. Yang, Y.-X. Guo, J.-Y. Xiong, Y. Liu, H. Pan, P.-S. Wang, X. Tong, and B. Guo, "Swin3d: A pretrained transformer backbone for 3d indoor scene understanding," *Computational Visual Media*, vol. 11, no. 1, pp. 83–101, 2025.

[29] C. Choy, J. Gwak, and S. Savarese, "4d spatio-temporal convnets: Minkowski convolutional neural networks," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019, pp. 3075–3084.

[30] S. Xie, J. Gu, D. Guo, C. R. Qi, L. Guibas, and O. Litany, "Pointcontrast: Unsupervised pre-training for 3d point cloud understanding," in *European conference on computer vision*. Springer, 2020, pp. 574–591.

[31] J. Hou, B. Graham, M. Nießner, and S. Xie, "Exploring data-efficient 3d scene understanding with contrastive scene contexts," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2021, pp. 15 587–15 597.

[32] X. Wu, X. Wen, X. Liu, and H. Zhao, "Masked scene contrast: A scalable framework for unsupervised 3d representation learning," in *Proceedings of the IEEE/CVF Conference on computer vision and pattern recognition*, 2023, pp. 9415–9424.

[33] C. Wang, L. Jiang, X. Wu, Z. Tian, B. Peng, H. Zhao, and J. Jia, "Groupcontrast: Semantic-aware self-supervised representation learning for 3d understanding," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 4917–4928.

[34] X. Wu, Z. Tian, X. Wen, B. Peng, X. Liu, K. Yu, and H. Zhao, "Towards large-scale 3d representation learning with multi-dataset point prompt training," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2024, pp. 19 551–19 562.

[35] X. Wu, Y. Lao, L. Jiang, X. Liu, and H. Zhao, "Point transformer v2: Grouped vector attention and partition-based pooling," *Advances in Neural Information Processing Systems*, vol. 35, pp. 33 330–33 342, 2022.

[36] B. Graham, M. Engelcke, and L. Van Der Maaten, "3d semantic segmentation with submanifold sparse convolutional networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 9224–9232.

[37] W. Wu, Z. Qi, and L. Fuxin, "Pointconv: Deep convolutional networks on 3d point clouds," in *Proceedings of the IEEE/CVF Conference on computer vision and pattern recognition*, 2019, pp. 9621–9630.

[38] H.-Y. Chiang, Y.-L. Lin, Y.-C. Liu, and W. H. Hsu, "A unified point-based framework for 3d segmentation," in *2019 International Conference on 3D Vision (3DV)*. IEEE, 2019, pp. 155–163.

[39] X. Yan, C. Zheng, Z. Li, S. Wang, and S. Cui, "Pointasnl: Robust point clouds processing using nonlocal neural networks with adaptive sampling," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 5589–5598.

[40] Y. Chen, J. Liu, X. Zhang, X. Qi, and J. Jia, "Largekernel3d: Scaling up kernels in 3d sparse cnns," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2023, pp. 13 488–13 498.

[41] B. Peng, X. Wu, L. Jiang, Y. Chen, H. Zhao, Z. Tian, and J. Jia, "Oa-cnns: Omni-adaptive sparse cnns for 3d semantic segmentation," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2024, pp. 21 305–21 315.