Boundary Cues for Improved 3D Semantic Segmentation

Waseem Iqbal[1], Jens-André Paffenholz[1]

[1] Institute of Geotechnology and Mineral Resources – Geomatics, Clausthal University of Technology, Germany
(waseem.iqbal, jens-andre.paffenholz)@tu-clausthal.de

Keywords: 3D Point Clouds, Boundary Cues, Semantic Segmentation, PointNeXt, S3DIS.

Abstract

Accurate semantic segmentation of 3D point clouds is essential for applications in photogrammetry, robotics, and large-scale scene understanding. While recent point-based architectures such as PointNeXt achieve strong performance through hierarchical feature learning, they still struggle near semantic boundaries, where points from different classes share local neighborhoods and feature aggregation leads to oversmoothing and ambiguous predictions. To address this limitation, we propose a lightweight boundary-aware learning framework that explicitly models boundary regions during training. The method introduces an auxiliary boundary prediction head that learns boundary cues from local semantic disagreement and integrates them into the segmentation process through a simple late-stage feature fusion mechanism. This design enhances feature discrimination near class transitions without modifying the backbone architecture or increasing inference complexity. Experiments on the S3DIS benchmark with the standard 6-fold cross-validation protocol show consistent improvements over the PointNeXt baseline, achieving gains of 3.22% in mean Intersection over Union (mIoU) and 2.85% in mean class accuracy (mACC) (relative), with notably improved predictions along object boundaries. These results show that incorporating boundary-aware supervision provides an effective and efficient approach to improving segmentation quality in challenging regions.

1. Introduction

Three-dimensional (3D) point clouds have become a fundamental representation for describing real-world environments. Advances in LiDAR sensing, multi-view photogrammetry, and mobile mapping systems enable the acquisition of dense and large-scale 3D data for applications such as urban modeling, digital twins, autonomous navigation, and robotic perception. Extracting meaningful semantic information from these datasets is therefore a key challenge in modern photogrammetry and computer vision.

Semantic segmentation of 3D point clouds aims to assign a semantic label to each point in a scene. Accurate segmentation enables higher-level tasks such as object recognition, scene understanding, and structural analysis. Early approaches relied primarily on handcrafted geometric descriptors combined with traditional machine learning algorithms. Although these methods captured certain geometric properties, they often struggled to generalize to complex scenes and large-scale datasets.

The introduction of deep learning methods significantly advanced the field of 3D point cloud processing. PointNet (Qi et al., 2017a) demonstrated that neural networks can operate directly on unordered point sets by applying shared multilayer perceptrons followed by symmetric aggregation functions. However, this architecture does not explicitly model local spatial relationships between points.

PointNet++ (Qi et al., 2017b) addressed this limitation by introducing hierarchical feature learning with neighborhood grouping and multi-scale aggregation, allowing the capture of both local geometric structures and global contextual information. Building upon this design, subsequent work has explored various improvements in sampling strategies, feature propagation, and architectural refinements.

More recently, PointNeXt (Qian et al., 2022) revisited the PointNet++ framework and demonstrated that careful architectural refinement, combined with modern training strategies, can significantly improve segmentation performance while maintaining computational efficiency. As a result, it achieves state-of-the-art performance on several benchmark datasets.

Despite these advances, accurately segmenting points near semantic boundaries remains a persistent challenge. Points located at the interface between different objects often share local neighborhoods with multiple semantic classes. As a result, neighborhood-based feature aggregation can blend features across class boundaries, leading to oversmoothing and ambiguous predictions. This issue becomes particularly pronounced in indoor environments, where objects are densely arranged and semantic transitions are frequent, such as between boards and walls, doors and walls, or furniture and floors.

To address this limitation, we propose a lightweight boundary-aware learning framework that explicitly incorporates boundary information into the segmentation process. An important aspect of our approach is that boundary supervision is automatically derived from existing semantic labels based on local neighborhood disagreement, eliminating the need for additional manual annotation while enabling the model to identify semantically ambiguous regions.

Building on this supervision signal, we introduce a lightweight auxiliary branch that learns to predict boundary likelihoods from intermediate features. The resulting boundary-aware representations are then integrated into the segmentation pipeline through a late-stage feature fusion mechanism, allowing boundary cues to refine feature representations. In addition, a boundary-aware optimization scheme further emphasizes these regions during training, encouraging the network to better preserve semantic discontinuities.

The proposed approach improves feature discrimination in

challenging regions while preserving the efficiency and simplicity of existing point-based architectures, as it does not require any modification to the backbone network.

The main contributions of this work are summarized as follows:

- We propose a lightweight boundary prediction head that captures boundary-sensitive features from 3D point clouds and is supervised using automatically generated boundary targets.
- We introduce a boundary-guided feature fusion mechanism that integrates boundary cues into the segmentation process at the prediction stage to enhance feature discrimination near semantic transitions.
- We present a boundary-aware training strategy that emphasizes boundary regions during learning.
- Experiments on the S3DIS dataset demonstrate that our framework consistently outperforms the baseline PointNeXt model while maintaining computational efficiency and leaving the backbone architecture unchanged.

2. Related Work

2.1 Deep Learning for Point Cloud Segmentation

Deep learning has substantially advanced 3D point cloud semantic segmentation by enabling direct processing of irregular and unordered data. PointNet (Qi et al., 2017a) established a foundational framework for point-based learning, which was later extended by hierarchical methods such as PointNet++ (Qi et al., 2017b) to better capture local geometric structures.

Beyond these foundational models, subsequent research has focused on improving neighborhood representation and feature interaction mechanisms. For example, DGCNN (Wang et al., 2019) introduces dynamic graph construction to capture edge-based relationships, while PointCNN (Li et al., 2018) learns transformations that enable convolution-like operations on point sets. Kernel-based approaches such as KPConv (Thomas et al., 2019) and efficient architectures like RandLA-Net (Hu et al., 2020) further explore the balance between accuracy and scalability.

More recent work has shifted attention toward optimization and scalability rather than entirely new architectures. PointNeXt (Qian et al., 2022) exemplifies this trend by revisiting existing designs and demonstrating that improved training strategies and architectural refinements can yield strong performance. In parallel, voxel-based methods such as VoxelNeXt (Chen et al., 2023) leverage structured representations for efficiency in large-scale scenarios.

Overall, current methods continue to face challenges in balancing fine-grained geometric detail, long-range context, and computational efficiency (Guo et al., 2023, Niemeyer et al., 2024).

2.2 Attention-based Point Cloud Networks

To better capture contextual relationships, attention mechanisms, particularly transformers, have been increasingly adopted in point cloud processing. Point Transformer (Zhao et al., 2021) introduces a self-attention operator that adaptively models interactions between neighboring points using both feature and spatial information, overcoming limitations of fixed local aggregation.

However, the flexibility of attention comes with increased computational cost. To mitigate this, Point Transformer V2 (Wu et al., 2022) improves efficiency through grouped vector attention and optimized pooling strategies. More recently, Point Transformer V3 (Wu et al., 2024) further simplifies the architecture and enhances scalability by focusing on efficient neighborhood construction and larger receptive fields.

Other variants, such as GeoTransformer (He et al., 2023), incorporate geometric priors and sparse attention to improve robustness in large-scale scenarios like outdoor LiDAR data (Roynard et al., 2023). Nevertheless, transformer-based approaches generally remain computationally demanding, which motivates continued exploration of lightweight yet effective architectures such as PointNeXt.

2.3 Boundary-aware Segmentation

Accurate boundary delineation remains a fundamental challenge in point cloud segmentation. Points near class transitions often exhibit ambiguous features due to mixed neighborhoods, leading to blurred predictions and misclassification.

Early approaches address this issue indirectly through loss design. Boundary Loss (Kervadec et al., 2019) emphasizes errors near class borders using distance-based supervision, while Superpoint Graphs (Landrieu and Simonovsky, 2018) enforce geometric consistency via region partitioning. However, these methods do not explicitly learn boundary representations.

More recent methods explicitly incorporate boundary modeling into the network. For example, Gated-SCNN (Takikawa et al., 2019) adopts a multi-task framework to jointly predict semantic labels and boundary maps, allowing boundary cues to refine feature learning. Similarly, dedicated boundary-aware learning strategies have been proposed for 3D point clouds to improve feature discrimination in transition regions (Gong et al., 2021). While effective, such approaches often introduce additional modules and increase model complexity.

An alternative direction integrates boundary prediction into feature extraction pipelines. BADENet (Zhao et al., 2024), for instance, leverages predicted boundary points to guide feature aggregation in encoder and decoder through additional graph-based components. Although this improves boundary localization, it further increases computational overhead and architectural complexity.

In contrast, our approach incorporates boundary reasoning in a lightweight manner by introducing an auxiliary boundary prediction head during training. Rather than modifying the backbone, the boundary features are fused into the segmentation features before the final segmentation logits. This enables effective boundary refinement with minimal computational overhead. This design preserves backbone efficiency while improving segmentation performance in challenging boundary regions, without incurring additional inference cost.

3. Methodology

The proposed framework introduces a lightweight boundary-aware training strategy to improve semantic segmentation of 3D

point clouds. Standard point-based architectures often struggle near object boundaries, as points in these regions tend to share local neighborhoods with points from different semantic classes, resulting in ambiguous feature representations. To address this limitation, we augment a strong backbone, such as PointNeXt (Qian et al., 2022), with an auxiliary boundary prediction head and a boundary-guided feature fusion mechanism, as illustrated in Figure 1.

Our framework consists of five main components, corresponding to flowchart modules (C0–C4):

- Hierarchical feature extraction (C0) using a point-based backbone.
- An auxiliary boundary prediction head (C1) as shown in Figure 1.
- Boundary-target generation (C2) as shown in Figure 2.
- Boundary-guided feature fusion (C3) as shown in Figure 1.
- Boundary-aware optimization (C4) as shown in Figure 2.

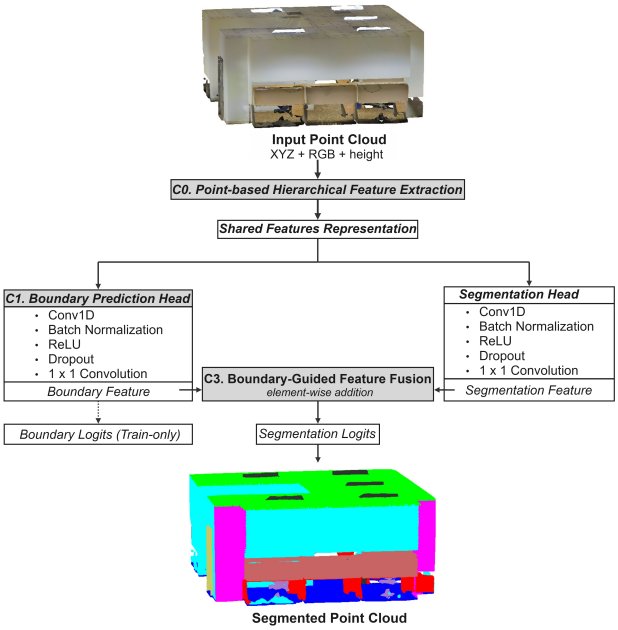

Figure 1. **Overview of the boundary-aware segmentation framework.** Input point cloud (XYZ + RGB + height) is encoded by a hierarchical point-based backbone into shared features. Two heads operate on these features: a boundary prediction head and a segmentation head. Boundary features are mapped via a 1×1 layer and fused into the segmentation features. The fused features produce segmentation logits whose argmax yields the final segmented point cloud. Segmentation logits are used at inference; while boundary supervision is applied only during training.

3.1 Backbone Feature Extraction

This subsection corresponds to the hierarchical feature extraction module (C0) shown in Figure 1. The backbone network follows the PointNeXt (Qian et al., 2022) architecture, which builds upon the hierarchical design of PointNet++ (Qi et al., 2017b). The model processes an input point cloud $\{p_i\}_{i=1}^{N}$,

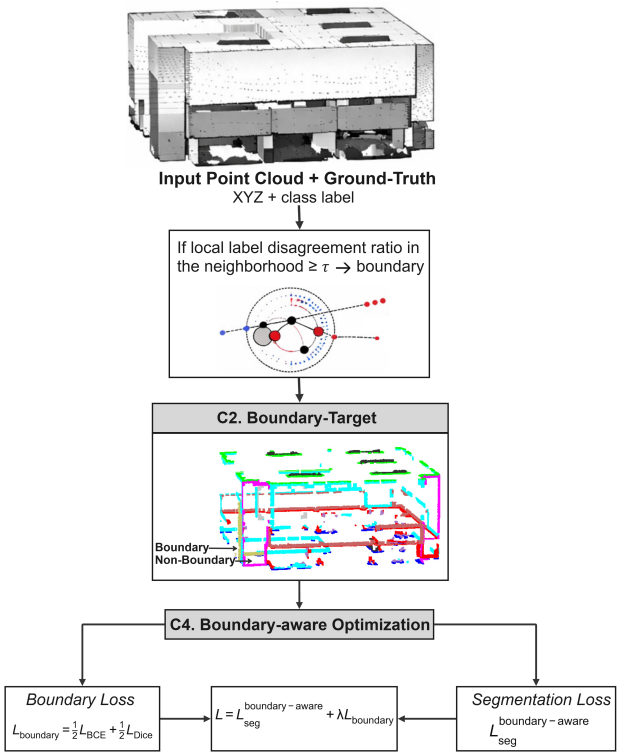

Figure 2. **Boundary-target generation.** From the input point cloud (XYZ) and ground-truth labels, local neighborhoods are built (radius-based ball query). Points whose neighborhoods contain mixed semantic labels are marked as boundary, forming a binary mask. This target supervises the boundary prediction head (C1) as shown in Figure 1 and reweights the segmentation loss. Segmentation logits are used at inference only.

where each point p_i is represented by its spatial coordinates and optional features such as color or intensity.

The backbone network extracts hierarchical features through repeated sampling and grouping operations. In each layer, a subset of points is selected using farthest point sampling, and local neighborhoods are constructed using a radius-based search. Features from neighboring points are aggregated through multilayer perceptrons (MLPs) and residual connections.

This hierarchical structure enables the network to capture both local geometric patterns and global contextual information, producing intermediate feature representations that are shared by both the segmentation (cf Figure 1) and boundary prediction (C1 in Figure 1) heads.

3.2 Boundary-Target Generation

This subsection describes the boundary-target generation process (C2) illustrated in Figure 2. Boundary targets are derived from the semantic labels available in the training data without requiring additional annotations. For a point i with the semantic label l_i, we consider its local neighborhood $\mathcal{N}(i)$ consisting of k nearest neighbors. A local disagreement ratio is computed as:

$$p_{\text{disagree}}(i) = \frac{|\{j \in \mathcal{N}(i) : l_j \neq l_i\}|}{k} \tag{1}$$

This quantity measures the proportion of neighboring points whose labels differ from the central point. Points located near

semantic class boundaries naturally exhibit higher disagreement ratios.

A binary boundary-target is then defined as:

$$y_b(i) = \begin{cases} 1, & \text{if } p_{\text{disagree}}(i) \geq \tau, \\ 0, & \text{otherwise} \end{cases} \tag{2}$$

These boundary targets provide supervision for the boundary prediction head and reweight the segmentation loss as shown in Figure 2.

3.3 Boundary Prediction Head

This subsection introduces the auxiliary boundary prediction head (C1) shown in Figure 1. During training, a lightweight boundary prediction head is introduced. This head receives the intermediate features from the backbone to learn boundary-sensitive features and predicts boundary logits per point. In addition to the final boundary logits, the intermediate feature produced by the boundary branch is retained as the boundary feature F_b, which is later used in the boundary-guided feature fusion (Section 3.4).

Let $F \in \mathbb{R}^{N \times C}$ denote the intermediate point features of N points with C channels. A small multilayer perceptron (MLP) $f_b(\cdot)$ predicts a boundary probability for each point as follows:

$$\hat{y}_b(i) = \sigma(f_b(F_i)) \tag{3}$$

where $\sigma(\cdot)$ is the sigmoid activation function, producing $\hat{y}_b(i) \in [0, 1]$.

Boundary targets $y_b(i)$ are automatically computed from the disagreement of the local neighborhood (see Section 3.2). As boundary points are sparse and labels may be noisy, the prediction head is supervised with a combination of Binary Cross-Entropy (BCE) and Dice loss:

$$L_{\text{boundary}} = \frac{1}{2} L_{\text{BCE}} + \frac{1}{2} L_{\text{Dice}}, \tag{4}$$

where the Dice loss is defined as:

$$L_{\text{Dice}} = 1 - \frac{2 \sum_i y_b(i) \cdot \hat{y}_b(i) + \epsilon}{\sum_i y_b(i) + \sum_i \hat{y}_b(i) + \epsilon} \tag{5}$$

where $y_b(i)$ and $\hat{y}_b(i)$ denote the ground-truth and the predicted value for point i, respectively. The numerator represents twice the sum of the element-wise product of predictions and ground truth (the soft intersection), while the denominator is the total number of positive values in both prediction and ground truth. The Dice loss decreases as the overlap between prediction and ground truth increases, reaching zero when there is perfect agreement. In practice, a small epsilon $\epsilon = 1 \cdot 10^{-5}$ is added for numerical stability.

The auxiliary boundary head is used only during training; at inference, the backbone operates unchanged, ensuring identical runtime and memory requirements.

3.4 Boundary-Guided Feature Fusion

This subsection details the boundary-guided feature fusion mechanism (C3) depicted in Figure 1, which effectively incorporates boundary information into the segmentation process through a lightweight fusion mechanism. The boundary prediction branch produces intermediate boundary features F_b (see Section 3.3), which encode boundary-sensitive cues learned from local label inconsistencies.

These boundary features are first projected through a learnable 1×1 MLP $g(\cdot)$ to match the dimensionality of the segmentation feature F_{seg}. The transformed boundary features are then fused with the segmentation features via element-wise addition before the final segmentation logits:

$$F'_{\text{seg}} = F_{\text{seg}} + g(F_b) \tag{6}$$

This late-stage fusion strategy allows boundary information to directly refine the final feature representation used for classification, without altering the hierarchical feature extraction in the backbone. As a result, the model becomes more sensitive to semantic transitions, improving discrimination near object boundaries while preserving consistency in homogeneous regions. Notably, this design introduces only minimal computational overhead.

3.5 Boundary-aware Optimization

This subsection presents the boundary-aware optimization strategy (C4) illustrated in Figure 2. The network is trained using a joint optimization objective consisting of a boundary-aware segmentation loss and an auxiliary boundary prediction loss:

$$L = L_{\text{seg}}^{\text{boundary-aware}} + \lambda \cdot L_{\text{boundary}} \tag{7}$$

where λ balances the contribution of the boundary supervision.

The segmentation loss is formulated as a boundary-aware weighted cross-entropy loss. Specifically, a per point weight is defined as:

$$w_i = 1 + \beta \cdot y_b(i), \tag{8}$$

where $y_b(i) \in \{0, 1\}$ indicates whether the point i lies on a semantic boundary, and β controls the strength of the boundary emphasis. In homogeneous regions where $y_b(i) = 0$, the weighting is reduced to standard supervision, preserving feature consistency. The segmentation loss is then given by:

$$L_{\text{seg}}^{\text{boundary-aware}} = \frac{1}{N} \sum_{i=1}^{N} w_i \cdot \left(-\log p(y_i \mid x_i)\right) \tag{9}$$

where x_i denotes the input features of point i and $p(y_i \mid x_i)$ denotes the predicted probability of its ground-truth class.

This formulation increases the contribution of boundary points during training, encouraging the network to learn more discriminative features in regions with high semantic ambiguity while

discouraging cross-class feature mixing during neighborhood aggregation.

The auxiliary boundary loss L_{boundary} supervises the boundary prediction head (see Section 3.3), which helps mitigate the class imbalance due to the sparsity of boundary points.

4. Experimental Setup

This section describes the experimental configuration used to evaluate the proposed boundary-aware training strategy. The setup closely follows the implementation details of the PointNeXt-s framework, with additional boundary-aware components integrated as described in Section 3.

4.1 Dataset

Experiments are conducted on the Stanford Large-Scale 3D Indoor Spaces (S3DIS) dataset (Armeni et al., 2016), a widely used benchmark for indoor 3D semantic segmentation. The dataset consists of six large-scale indoor areas spanning office rooms, corridors, and conference spaces, with dense pointwise annotations over 13 semantic classes.

Each point is represented by its 3D coordinates (XYZ) and RGB color. Following common practice, a 6-fold cross-validation protocol is adopted, in which each area is used once as the test set while the remaining five areas are used for training. For example, in one fold, the model is trained on Areas 1–4 and 6 and tested on Area 5. During preprocessing, 3D point clouds are voxelized with a voxel size of 0.04 m to reduce redundancy and ensure computational efficiency.

4.2 Implementation Details

Our implementation is based on the PointNeXt-s architecture. The backbone configuration, feature extraction pipeline, and decoder design remain unchanged from the original model.

Training is conducted using the OpenPoints framework with distributed data parallelism. Each batch contains 32 samples, and each sample includes up to 24,000 points.

4.3 Training Configuration

The network is trained for 100 epochs using the AdamW optimizer with an initial learning rate of 0.006 and weight decay of $1 \cdot 10^{-4}$. A cosine annealing schedule is used with a minimum learning rate of $1 \cdot 10^{-5}$.

Standard data augmentation techniques from PointNeXt-s are used, including random rotation around the vertical axis, scaling, jittering, and color-based augmentations.

For the proposed method, boundary-aware supervision is incorporated during training. Boundary targets are generated on-the-fly using a local neighborhood defined by a ball query with a fixed radius and a k-nearest neighbor constraint ($k = 16$), followed by a thresholding operation with $\tau = 0.3$. The boundary loss is weighted by $\lambda = 0.3$, and boundary points are further emphasized in the segmentation loss using a weighting factor of $\beta = 1.5$. All hyperparameters are determined empirically.

4.4 Evaluation Protocol

We report standard semantic segmentation metrics as follows:

- Overall Accuracy (OA): fraction of correctly classified points over the entire dataset. OA measures global correctness but can be dominated by large classes.
- Mean Class Accuracy (mACC): average per-class accuracy across all semantic categories. mACC provides a class-balanced measure, particularly relevant for smaller or boundary-sensitive classes such as *window*, *door*, and *board*.
- Mean Intersection over Union (mIoU): average IoU across all classes. mIoU accounts for both false positives and false negatives, giving a robust assessment of segmentation quality near interfaces.

Evaluation is performed on full scenes using a voxel-based inference strategy. Predictions from overlapping partitions are aggregated by averaging logits, and final labels are obtained via argmax.

The boundary prediction branch is used only during training and is disabled at inference time, ensuring that the evaluation cost remains identical to the baseline model.

4.5 Reproducibility

All experiments are conducted using the same training pipeline, data preprocessing, and hyperparameters as the original PointNeXt-s implementation. No changes are made to the backbone architecture or training schedule beyond the proposed boundary-aware components.

This controlled setup ensures that any performance gains are solely attributable to the introduced boundary-aware supervision.

5. Results and Discussion

This section presents both quantitative and qualitative evaluations of the proposed boundary-aware training framework on the S3DIS dataset. The method is compared against the PointNeXt-s baseline under identical settings to isolate the impact of boundary-aware supervision.

5.1 Quantitative Results

Table 1 reports the overall performance using the standard 6-fold cross-validation protocol. The proposed method improves mIoU from 67.16% to 69.32% (+3.22% relative) and mACC from 76.50% to 78.68% (+2.85% relative), while maintaining comparable OA.

Metric	Baseline (%)	Boundary-aware (%)
OA	87.58	**87.66**
mACC	76.50	**78.68**
mIoU	67.16	**69.32**

Table 1. **Overall segmentation performance on the S3DIS** dataset using 6-fold cross-validation. The best values are shown in bold.

These improvements indicate that the model benefits from enhanced class-wise discrimination rather than simply increasing

dominant-class accuracy. This observation is further supported by the relatively stable OA, suggesting that gains are primarily achieved in challenging regions rather than already well-classified areas.

Table 2 presents a per-class IoU comparison. The largest improvements are observed for boundary-sensitive and geometrically ambiguous classes such as *window* (+13.46%), *door* (+5.76%), and *board* (+3.93%). These categories typically lie adjacent to walls or other planar surfaces, making them prone to feature mixing in standard point-based aggregation.

In contrast, classes with large homogeneous regions such as *floor* and *table* show marginal decreases or remain stable, suggesting that boundary-aware weighting primarily redistributes learning capacity toward difficult regions, which may slightly reduce the optimization capacity for large homogeneous areas.

Class	Baseline (%)	Boundary-aware (%)
ceiling	94.42	**94.63**
floor	**95.38**	93.19
wall	81.19	**81.65**
beam	39.20	**39.99**
column	**47.65**	47.63
window	53.72	**67.18**
door	72.22	**77.98**
chair	69.15	**72.11**
table	**71.19**	70.23
bookcase	67.07	**68.02**
sofa	62.08	**64.43**
board	59.63	**63.56**
clutter	60.16	**60.55**

Table 2. **Per-class IoU comparison on S3DIS dataset**. Each value shows the mean over 6-fold. The best values are shown in bold.

5.2 Qualitative Analysis

Figure 3 and Figure 4 illustrate qualitative comparisons between the baseline and the proposed method. The most notable improvements occur along object boundaries and in regions with high semantic ambiguity.

In both figures, the baseline model exhibits visible *label bleeding* near class transitions, particularly at wall–door and wall–bookcase interfaces. These errors appear as fragmented predictions caused by feature mixing across neighboring classes. In contrast, the proposed method reduces these effects, resulting in sharper transitions between adjacent classes.

The improvement maps further highlight these effects. A noticeable portion of correctly updated predictions (green) is located along boundaries, thin structures, and regions with multiple nearby classes. This indicates that the boundary-aware supervision helps the model better handle locally ambiguous regions.

For example, in Figure 3, the boundary-aware model more clearly separates the door from the surrounding wall, reducing misclassifications along the edges. Similarly, in Figure 4, improvements are visible in cluttered indoor areas, where objects such as bookcases and windows show better separation from adjacent surfaces.

Red regions (where the baseline performs better) occur mainly in homogeneous or cluttered areas. Gray regions mark locations where both methods struggle, often due to occlusion or inherent ambiguity.

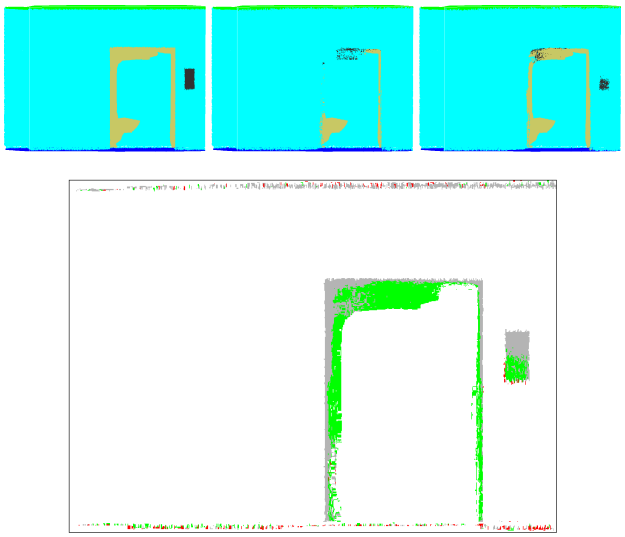

Figure 3. **Qualitative results on S3DIS Area 5.** From left to right: ground truth, PointNeXt baseline, and boundary-aware predictions. Colors indicate semantic classes: cyan (wall), blue (floor), tan (door), light green (ceiling). Bottom: improvement map (green: ours better, red: baseline better, white: both correct, gray: both incorrect).

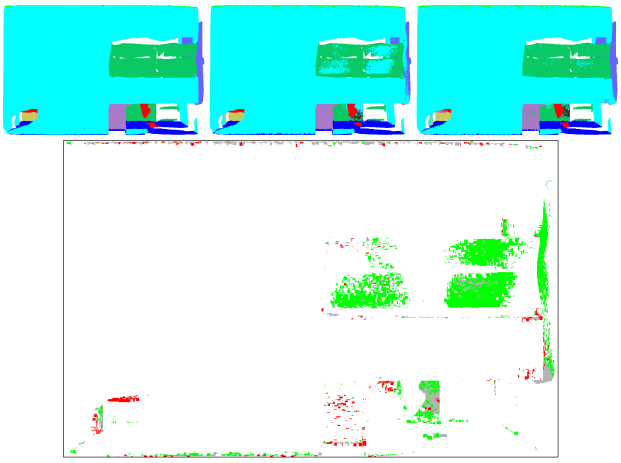

Figure 4. **Additional qualitative comparison on S3DIS.** The boundary-aware model improves segmentation consistency in cluttered indoor scenes, particularly near object interfaces. From left to right: ground truth, PointNeXt baseline, and boundary-aware predictions. Colors indicate the semantic classes: blue (floor), cyan (wall), tan (door), red (chair), light green (ceiling), purple (table), green (bookcase), light blue (window), and black (clutter). Bottom: improvement map (green: ours better, red: baseline better, white: both correct, gray: both incorrect).

5.3 Discussion

The results consistently demonstrate that boundary-aware supervision improves segmentation performance, particularly in regions where multiple semantic classes meet. The gains in mIoU and mACC, together with the qualitative improvements, indicate that the model learns more discriminative representations for boundary points.

Importantly, the improvements are not uniformly distributed

across all regions. Instead, they are concentrated in challenging areas where standard point-based methods tend to oversmooth features due to local aggregation. By emphasizing boundary points during training, the proposed method reduces feature mixing and preserves sharper semantic transitions.

Another key observation is that these improvements are achieved with minimal trade-offs. While slight performance variations are observed for some large homogeneous classes, the overall gains in difficult categories outweigh these effects, leading to a net improvement in segmentation quality.

Finally, since the boundary prediction head is only used during training, the inference cost remains unchanged. This makes the proposed approach an efficient and practical enhancement for existing point-based segmentation models.

Despite its effectiveness, the proposed method has some limitations. First, it relies on ground-truth labels to derive boundary targets, making it sensitive to annotation noise. Second, its performance depends on hyperparameters, such as neighborhood size and disagreement threshold, which may require careful tuning across datasets. Additionally, extremely sparse or heavily occluded regions remain challenging, as reliable neighborhood information is limited. Finally, since boundary points constitute only a small fraction of the data, the overall improvement may be constrained in highly imbalanced scenarios.

6. Conclusions and Outlook

In this work, we introduced a lightweight and backbone-agnostic boundary-aware learning framework for 3D point cloud semantic segmentation. The proposed approach leverages automatically generated boundary supervision derived from local label disagreement, enabling the network to explicitly identify semantically ambiguous regions without requiring additional annotations. A key design choice is the integration of an auxiliary boundary prediction head during training, whose learned features are incorporated into the segmentation process via a simple boundary-guided feature fusion mechanism at the prediction stage.

Notably, the proposed method does not modify the backbone architecture or inference pipeline, preserving the computational efficiency and scalability of the underlying model. This makes the approach readily applicable to existing point-based segmentation frameworks.

Experimental results on the S3DIS dataset using the standard 6-fold cross-validation protocol demonstrate consistent improvements over a strong PointNeXt baseline under identical training conditions. In particular, the method improves mIoU from 67.16% to 69.32% (+3.22% relative) and mACC from 76.50% to 78.68% (+2.85% relative), with notable gains for boundary-sensitive classes such as *window*, *door*, and *board*. Qualitative results further indicate sharper and more coherent predictions at object interfaces, reducing label ambiguity in transition regions.

These results demonstrate that incorporating boundary-aware supervision during training effectively complements standard feature learning and leads to improved segmentation quality without increasing inference complexity.

Limitations: The proposed framework relies on boundary signals derived from local label variation and introduces a small number of additional design choices, e.g., neighborhood definition. While effective in practice, these aspects may require careful adaptation across datasets with different characteristics. Nonetheless, they do not affect the simplicity of the inference pipeline.

Outlook: Future work will focus on improving the robustness and generality of boundary-aware learning. Promising directions include adaptive boundary estimation strategies, multi-scale or continuous boundary representations, and the integration of complementary modalities such as multi-view imagery. Extending the proposed approach to transformer-based architectures and evaluating it on an outdoor dataset with boundary-sensitive metrics may further enhance both performance and interpretability.

Overall, this work demonstrates that boundary-aware supervision can be incorporated in a simple and efficient manner, providing consistent improvements in segmentation accuracy while preserving the practicality of modern point-based architectures.

References

Armeni, I., Sener, O., Zamir, A. R., Jiang, H., Brilakis, I., Fischer, M., Savarese, S., 2016. 3d semantic parsing of large-scale indoor spaces. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 1534–1543.

Chen, Y., Liu, J., Zhang, X., Qi, X., Jia, J., 2023. Voxelnext: Fully sparse voxelnet for 3d object detection and segmentation. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 21674–21683.

Gong, J., Xu, J., Tan, X., Zhou, J., Qu, Y., Xie, Y., Ma, L., 2021. Boundary-Aware Geometric Encoding for Semantic Segmentation of Point Clouds. *arXiv preprint arXiv:2101.02381*.

Guo, Y., Wang, H., Hu, Q., Liu, H., Liu, L., Bennamoun, M., 2023. A Comprehensive Survey on Deep Learning for 3D Point Clouds. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(1), 1–21.

He, Y., Wang, J., Zhou, Z., Xu, H., 2023. Geometric transformer for fast and robust point cloud registration. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 13590–13599.

Hu, Q., Yang, B., Xie, L., Li, Y., Dai, F., Zhang, Y., Zhu, Y., 2020. Randla-net: Efficient semantic segmentation of large-scale point clouds. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 11108–11117.

Kervadec, H., Bouchtiba, J., Desrosiers, C., Granger, E., Dolz, J., Ayed, I. B., 2019. Boundary loss for highly unbalanced segmentation. *Proceedings of the Medical Imaging with Deep Learning (MIDL)*.

Landrieu, L., Simonovsky, M., 2018. Large-scale point cloud semantic segmentation with superpoint graphs. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 4558–4567.

Li, Y., Bu, R., Sun, M., Wu, W., Di, X., Chen, B., 2018. Pointcnn: Convolution on x-transformed points. *Advances in Neural Information Processing Systems (NeurIPS)*, 31.

Niemeyer, J., Rottensteiner, F., Soergel, U., 2024. Deep learning-based 3d semantic segmentation: Recent advances and challenges. *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, XLVIII-2, 100–107.

Qi, C. R., Su, H., Mo, K., Guibas, L. J., 2017a. Pointnet: Deep learning on point sets for 3d classification and segmentation. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 652–660.

Qi, C. R., Yi, L., Su, H., Guibas, L. J., 2017b. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. *Advances in Neural Information Processing Systems (NeurIPS)*, 30.

Qian, G., Li, Y., Peng, H., Mai, J., Hammoud, H. A. A. K., Elhoseiny, M., Ghanem, B., 2022. Pointnext: Revisiting pointnet++ with improved training and scaling strategies. *Advances in Neural Information Processing Systems (NeurIPS)*, 35.

Roynard, X., Deschaud, J.-E., Goulette, F., 2023. Toronto-3d: A large-scale mobile lidar dataset for semantic segmentation of urban roadways. *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, X-1/W1, 123–130.

Takikawa, T., Acuna, D., Jampani, V., Fidler, S., 2019. Gated-scnn: Gated shape cnns for semantic segmentation. *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 5229–5238.

Thomas, H., Qi, C. R., Deschaud, J.-E., Marcotegui, B., Goulette, F., Guibas, L. J., 2019. Kpconv: Flexible and deformable convolution for point clouds. *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 6411–6420.

Wang, Y., Sun, Y., Liu, Z., Sarma, S. E., Bronstein, M. M., Solomon, J. M., 2019. Dynamic Graph CNN for Learning on Point Clouds. *ACM Transactions on Graphics*, 38(5), 1–12.

Wu, X., Jiang, L., Wang, P.-S., Liu, Z., Liu, X., Qiao, Y., Ouyang, W., He, T., Zhao, H., 2024. Point transformer v3: Simpler, faster, stronger. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 24076–24085.

Wu, X., Lao, Y., Jiang, L., Liu, X., Zhao, H., 2022. Point transformer v2: Grouped vector attention and partition-based pooling. *Advances in Neural Information Processing Systems (NeurIPS)*, 35.

Zhao, H., Jiang, L., Jia, J., Torr, P. H. S., Koltun, V., 2021. Point transformer. *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 16259–16268.

Zhao, J., Lu, J., Zhou, J., Zhang, K., 2024. Boundary-aware Dual Edge Convolution Network for Indoor Point Cloud Semantic Segmentation. *Computers Electrical Engineering*, 116, 109219.