EAI Endorsed Transactions
on Scalable Information Systems
Research Article EAI.EU

A Point Cloud Instance Segmentation Framework with Attention Mechanisms and Semantic Refinement

Xi Chen, Meiji Chen and Hao Lin*

School of Information Science and Technology, Southwest Jiaotong University, Chengdu, China

Abstract

INTRODUCTION: Point cloud instance segmentation, a critical 3D computer vision task, faces significant challenges in complex indoor environments. Methods often suffer from insufficient feature extraction and a strong coupling between semantic prediction and instance features, where semantic errors cascade and limit overall accuracy.
OBJECTIVES: This paper proposes an innovative approach using attention mechanisms and semantic refinement to address these limitations. The primary goal is to enhance feature representation and alleviate the strong dependency between semantic prediction and instance segmentation.
METHODS: We introduce three key innovations: 1) A reverse attention mechanism to improve multi-level feature fusion; 2) An instance soft clustering strategy incorporating semantic scores to weaken the semantic-instance coupling; and 3) A self-attention-based instance refinement network. Finally, a dual-branch scoring mechanism, combining classification and mask scores, jointly determines confidence levels to further mitigate semantic errors.
RESULTS: Evaluated on the S3DIS dataset, our model achieved 74.1% mean Precision(mPrec) on the Area 5 test. In the more rigorous six-fold cross-validation, it achieved 76.8% mPrec and 72.3% mean recall rate(mRec), outperforming the state-of-the-art (SOTA) model SoftGroup by 1.5% and 2.5%, respectively.
CONCLUSION: The proposed method significantly improves instance segmentation accuracy and demonstrates stronger robustness in complex scenes. It effectively resolves the strong coupling issue, providing a novel technical pathway for point cloud instance segmentation. While primarily optimized for dense indoor scans, adapting this framework for extremely sparse outdoor point clouds remains a compelling direction for future exploration.

Keywords: point cloud instance segmentation, reverse attention mechanism, self-attention mechanism, feature interleaving.

Received on 07 January 2026, accepted on 24 April 2026, published on 19 May 2026

doi: 10.4108/eetsis.11524

1. Introduction

A Point cloud is a collection of numerous discrete 3D points that capture the spatial coordinates of object surfaces, enabling a complete representation of the target object's geometric shape and spatial structure. Point clouds are inherently unordered, unstructured, and exhibit varying point densities. Point cloud analysis, as a cornerstone of 3D deep learning, encompasses critical tasks such as 3D object detection, semantic segmentation, and the instance segmentation task discussed in this work[1]. These segmentation techniques are applied broadly, from general scene understanding to specialized fields such as 3D defect classification and segmentation in industrial systems[2]. The task is also critical for intelligent robot systems operating in complex indoor environments[3] and 3D LiDAR data processing in autonomous vehicles[4].

Point cloud instance segmentation aims to perform semantic classification for each point while simultaneously distinguishing between individual instances within the same

*Corresponding author. Email: linhao@my.swjtu.edu.cn

category, thereby enabling fine-grained scene understanding. Current deep learning approaches for point cloud instance segmentation can be broadly categorized into three paradigms: proposal-based methods, grouping-based methods, and end-to-end Transformer-based methods.

1.1. Proposal-based Methods

Proposal-based (or top-down) methods first generate a set of class-agnostic object proposals and then perform instance-level segmentation within each proposal. Early works, often inspired by 2D counterparts, relied on 3D bounding boxes as proposals. This paradigm has since evolved, with query-based mechanisms emerging as a prominent direction[5]. For instance, Mask3D[6] pioneered the use of learnable queries to directly predict instance masks, eliminating the need for explicit object detection or bounding boxes. This framework has proven effective and has since been applied to large-scale tasks, such as the segmentation of individual rooms in indoor building point clouds[7]. While intuitive, the performance of these methods is still highly dependent on the quality of the initial proposals or learned queries, and they can struggle with intricate object boundaries.

1.2. Grouping-based Methods

In contrast, grouping-based (or bottom-up) methods have become the dominant approach in 3D instance segmentation. These methods reverse the pipeline: they first predict point-level features (such as semantic labels and geometric offsets) and then cluster points into distinct instances.

This paradigm has seen significant evolution. Early works like MASC[8] explored multi-scale affinity and sparse convolutions to group points. Subsequent methods focused on improving both feature representation and the clustering process. For instance, HAIS[9] introduced hierarchical aggregation to learn instance-aware features. While 3D-SDIS[10] explored novel encoding strategies, using frequency fusion (FFT) and dual-sphere sampling to enhance feature learning across multiple domains.

Simultaneously, others focused on the joint learning and clustering aspects. JSPNet[11] proposed learning joint semantic and instance segmentation by leveraging feature self-similarity. On the clustering front, SoftGroup[12] introduced a "soft" association mechanism, significantly improving robustness to noisy semantic predictions.

However, despite their prevalence, these grouping-based methods suffer from two critical shortcomings:

- Insufficient Feature Representation: Many approaches [8,9,11,12] still employ U-Net backbones based on 3D submanifold sparse convolution (SSCNs)[13] to extract point features. These architectures use simple skip connections, which struggle to effectively fuse multi-scale semantic and geometric features, leading to insufficient feature representation. This limitation is widely recognized: EP-Net[14], for instance, addresses a similar coarse joint processing of geometric and semantic features in point-based methods via Decoupled Feature Aggregation (DFA). Similarly, MTCloud[15] was designed to enhance semantic feature extraction by improving the capture of local and global context. While these works validate the importance of feature enhancement, developing an optimal fusion mechanism remains an active research area.
- Strong Task Coupling: They establish a strong, often detrimental, coupling between semantic prediction and instance segmentation. This rigid reliance on initial semantic predictions means that any semantic errors inevitably cascade and propagate. While methods like JSPNet attempt to mitigate this by jointly learning the tasks, this strong coupling remains the primary obstacle to achieving higher accuracy and robustness. Formally, "strong task coupling" can be mathematically defined by the logical condition for grouping two points p_i and p_j into the same instance:

$$C(p_i, p_j) = \left(\left\|x_i^{shift} - x_j^{shift}\right\|_2 < r\right) \wedge \left(s_i = s_j\right) \quad (1)$$

where x_i^{shift} represents the shifted spatial coordinates, r is the spatial clustering radius, and s is the predicted hard semantic label. Under this strict logical "AND" constraint, any semantic error ($s_i \neq s_j$) directly forces the spatial affinity to be ignored. Consequently, the overall instance segmentation performance is strictly capped by semantic accuracy, allowing errors to cascade irreversibly.

1.3. Transformer-based Methods

To address the challenge of insufficient feature representation, researchers have explored various novel architectures. For instance, some work has proposed an Inception-based deep network (PIG-Net) to effectively characterize the local and global geometric details of point clouds[16]. Concurrently, a third paradigm has recently emerged, leveraging the power of Transformer architectures. The self-attention mechanism is exceptionally adept at capturing long-range dependencies and global contextual information.

Research has explicitly demonstrated that adopting attention mechanisms into existing deep learning networks yields a clear and superior performance advantage[17]. This trend is evident in novel backbone designs; for instance, LCASAFormer[18] employs specialized local cross-attention (LCASA) and Position-dominated self-attention (PDSA) modules to enhance local and global context aggregation, addressing the limitations of traditional pooling operations. Hong et al.[19] introduced dynamic shifting networks for unified 4D panoptic segmentation, showcasing advanced feature encoding. Another, a Global Shape Attention (GSA) module was designed to capture the contours of instances[20].

However, while these Transformer-based models have made significant strides in feature extraction, they are not without their own limitations. First, their complex self-attention mechanisms and deep decoders often introduce

substantial computational overhead and high memory consumption, making them difficult to deploy in real-time applications. Second, they often require massive amounts of training data to converge effectively. Finally, while powerful at capturing global context, they may still struggle with fine-grained geometric details and, most importantly, often do not explicitly address the strong coupling problem, which remains a key bottleneck.

1.4. Our Approach

The limitations of existing methods—insufficient feature representation in grouping-based methods and the unresolved task-coupling problem motivate our work. This paper proposes an innovative approach that simultaneously targets both critical limitations. We introduce a reverse attention mechanism to enhance multi-scale feature representation and a dedicated semantic refinement module to explicitly alleviate the strong dependency between semantic prediction and instance segmentation. Our key contributions are:

- A feature extraction network based on reverse attention mechanism is designed to enhance feature learning for point-wise semantic prediction and center offset vector regression.
- An instance soft clustering strategy incorporating semantic scores is employed to effectively weaken the strong coupling between instance prediction and semantic labels. Furthermore, a self-attention-based instance refinement network is constructed to enhance instance features, thereby further optimizing the expressive capability of instance representations.
- An innovative dual-branch evaluation mechanism, comprising classification scoring and mask scoring branches, is designed to jointly compute confidence scores, thereby further reducing the coupling constraint between instance prediction and semantic prediction.

2. Related Work

2.1. MASC

MASC is a typical example of a three-dimensional instance segmentation method based on semantic prediction. As shown in Figure 1, The network first extracts the point cloud features of the input point cloud through the U-Net backbone network constructed by 3D submanifold sparse convolution, generates the semantic prediction results of each point, and predicts the affinity relationship between adjacent voxels at multiple scales simultaneously. Subsequently, the points are classified into different instance sets through topological clustering based on affinity to form the final instance segmentation result.

The instance prediction of the MASC network strongly relies on the semantic classification results, reflecting the segmentation paradigm of "semantics first, then instances". The advantage of this method lies in the introduction of submanifold sparse convolution for efficient feature learning of voxelized point clouds, significantly improving the computational efficiency of the algorithm. The multi-scale affinity prediction mechanism can capture the instance features at different scales. Meanwhile, replacing the traditional similarity matrix calculation with affinity prediction reduces the computational overhead and improves the segmentation efficiency.

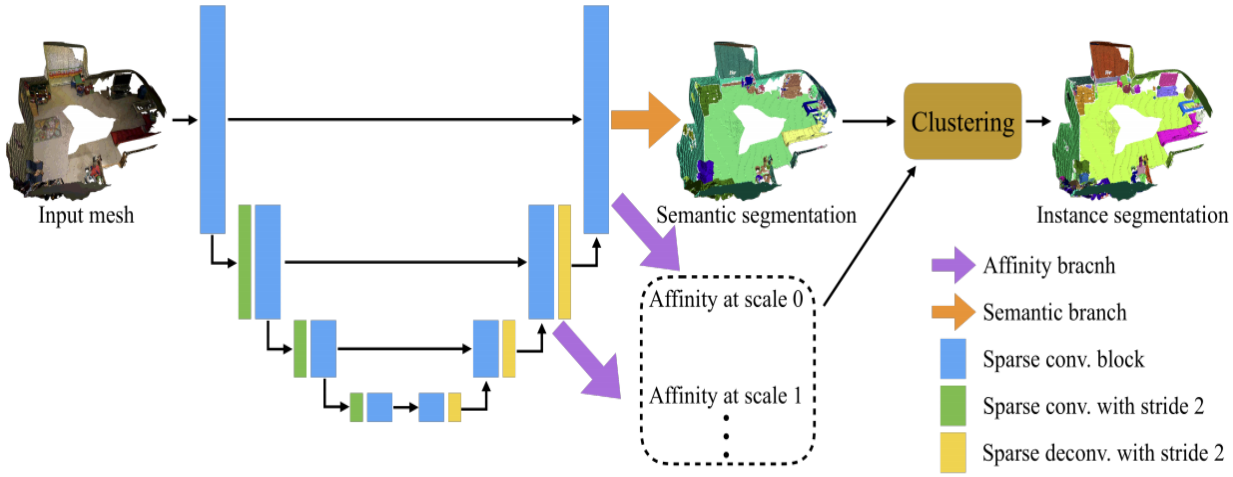

Figure 1. MASC network structure diagram

2.2. HAIS

HAIS[21] proposed a highly innovative and efficient hierarchical aggregation method, which not only improved the segmentation accuracy but also made significant progress in the reasoning speed, providing a new idea for the segmentation of large-scale point cloud instances. As shown in Figure 2, this model consists of four parts: Point-wise Prediction Network, Point Aggregation, Set Aggregation and Intra-instance Prediction Network.

To address the problem that point aggregation cannot ensure the complete aggregation of all points within the same instance, HAIS introduces a aggregation-level aggregation module to optimize the initial instance segmentation prediction. After obtaining the instance prediction, HAIS further introduces the internal prediction network of instances to achieve more refined feature extraction and optimization. This network is based on 3D submanifold sparse convolution, deeply mines the internal features of instances, and performs mask prediction. Meanwhile, it combines the confidence score to adjust the misabsorption points in the set aggregation stage to improve the segmentation accuracy. This module works in synergy with the hierarchical aggregation mechanism to compensate for the possible detail errors that may occur during the hierarchical aggregation process.

While HAIS is efficient, it relies on a "hard" semantic assignment before grouping. This hard logic forces boundary-ambiguous points into a single category prematurely, causing irreversible misgrouping. In contrast, our "soft" clustering method (detailed in Section 3.3) uses continuous semantic scores instead of one-hot labels. By allowing ambiguous points to associate with multiple semantic classes temporarily, we effectively prevent semantic errors from propagating at complex instance boundaries.

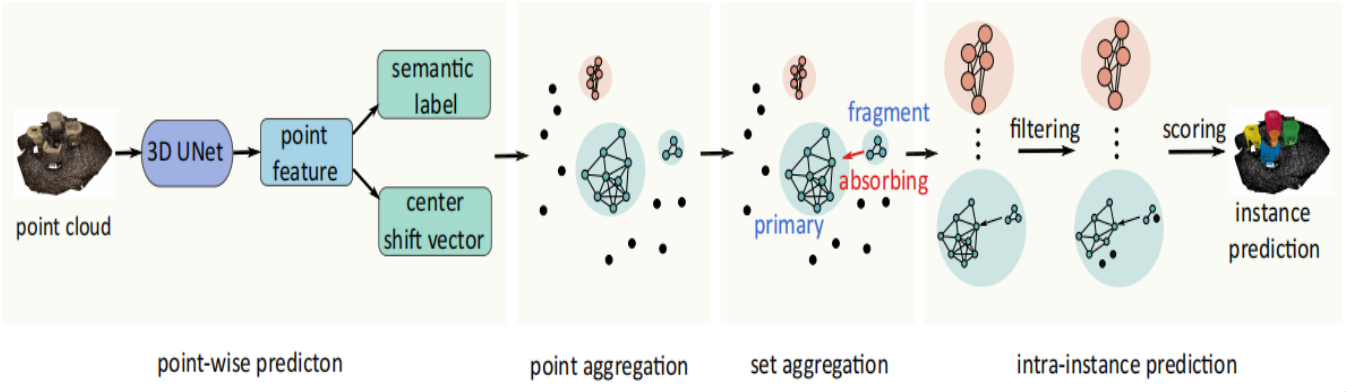

Figure 2. HAIS network structure diagram

2.3.3D U-Net Based on Sparse Convolution

The SSCNs model proposes that the voxel-based featur e extraction method has gradually become another effectiv e means of feature extraction in point cloud deep learning. This model introduces a sparse convolutional neural netw ork structure, effectively overcoming the dual limitations of traditional voxel-based feature extraction methods in te rms of computational efficiency and feature representation performance. Among them, the submanifold sparse convo lution proposed by SSCNs innovatively introduces a dyna mic condition calculation mechanism. Its core lies in deter mining the spatial distribution state of the input features in real time through the spatial activation detection module. In SSCNs, the submanifold sparse convolution and the ful ly convolutional neural network[22] (FCN) or U-Net netw ork[23] architectures are combined to construct two efficie nt network structures, further optimizing the feature extrac tion process. Due to its improvement in the efficiency of f eature extraction, the submanifold sparse convolution of U -Net network has been adopted by many 3D point cloud in stance segmentation models [8,9,11,12], positioning this a pproach as a key application of the Convolutional Neural Network (CNN) paradigm in 3D point cloud processing[2 4].

3. Method

3.1. Overall Structure

The overall structure of the point cloud instance segmentation network based on the attention mechanism and semantic correction is shown in Figure 3.

The point cloud instance segmentation network based on the attention mechanism and semantic correction consists of three parts: Firstly, the feature extraction network based on reverse attention combines the 3D U-Net model based on submanifold sparse convolution with the reverse attention mechanism to extract features from voxelized point clouds and generate point-to-point semantic prediction scores and center point offset vectors. Among them, the reverse attention mechanism is used to fuse the features of the encoding layer and the decoding

layer to enrich semantic and geometric information. Then, combined with the instance aggregation module of semantic scores, referring to the design of the SoftGroup model, instance soft clustering is carried out based on the output of the backbone network. Through the soft semantic threshold screening mechanism, one-point association and multiple categories are achieved to generate the initial instance segmentation prediction. Finally, the instance correction network based on self-attention introduces the self-attention mechanism to optimize the feature representation, draws on the Point Transformer structure to enhance the feature instantiation, uses the micro 3D U-Net to extract local features, and evaluates the instance prediction results through the dual-branch evaluation mechanism combined with the mask prediction quality and classification confidence. Alleviate the correlation between instances and semantic prediction, and output instance-level prediction.

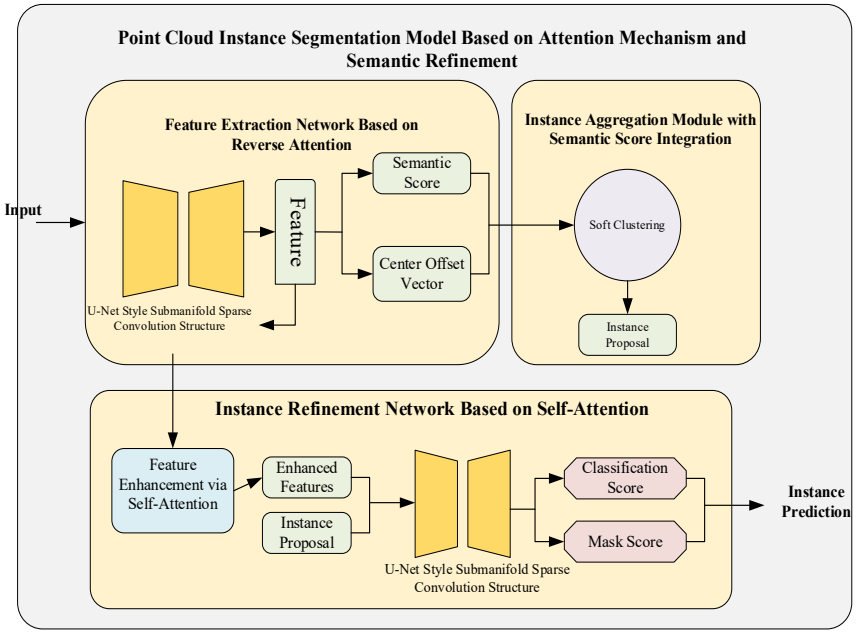

Figure 3. Point cloud instance segmentation model based on attention mechanism and semantic correction

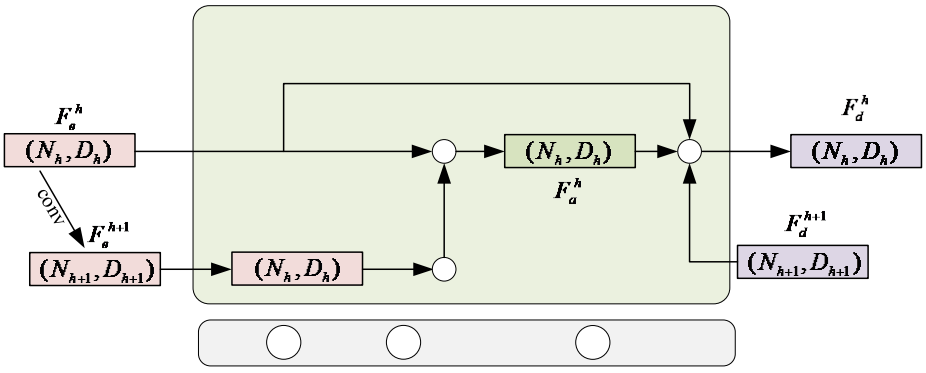

Figure 4. Schematic diagram of the reverse attention mechanism

3.2. Feature extraction network based on reverse attention

The Backward Attentive Fusing Network with Local A ggregation Classifier (BAF-LAC)[25] first introduced the re verse attention scheme for feature fusion. As shown in Fig-

ure 4, it is a schematic diagram of the reverse attention mechanism fusion of the encoder features at the h+1 layer. The encoder features at the h+1 layer are used to enhance the current encoder features after backpropagation. This enables the feature differences between the low-level encoder and the high-level encoder to be taken into account when performing feature stitching with the corresponding decoder, and finally outputs the enhanced encoder features F_a^h for fusion with the decoder features.

As shown in Figure 5, it is the structure diagram of the sparse convolutional 3D U-Net backbone network based on the reverse attention mechanism. The input data is a point collection composed of N points, and each point contains three-dimensional spatial coordinate information x, y, z and color parameters r, g, b. First, the point cloud information is voxelized. The point cloud feature information with an input size of $N \times 6$ is converted into voxel feature information with a size of $K \times 6$. The voxelization processing spatially discretizes the three-dimensional point cloud data according to the preset grid size and decomposes it into K voxel structures of equal volume. The information of each voxel is the average value of the point cloud information in that voxel. In addition, During the voxelization process, it is also necessary to record the mapping relationship between points and voxels for the subsequent restoration of anti-voxelization.

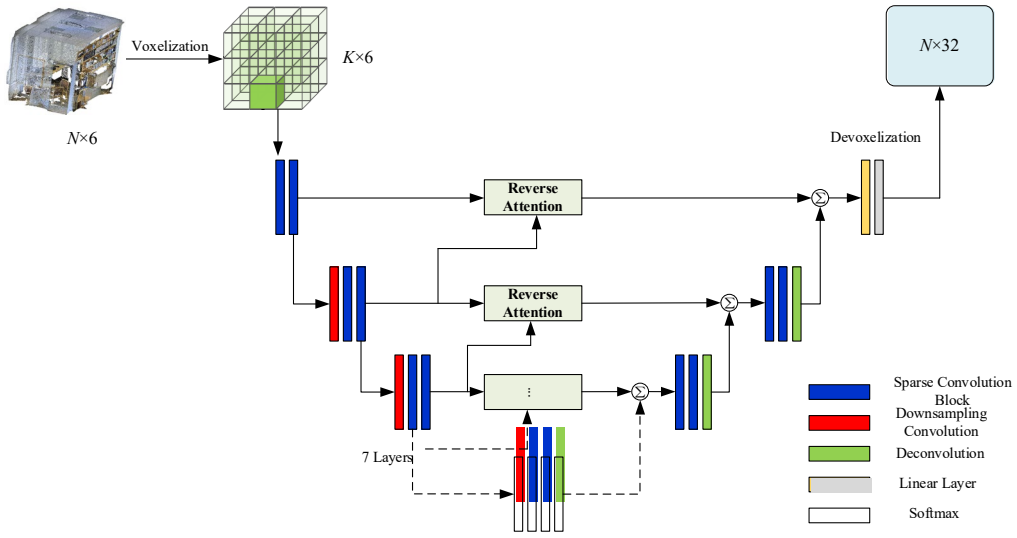

Figure 5. Sparse convolutional 3D U-Net structure based on the reverse attention mechanism

For the decoder features of each layer after feature fusion, as shown in Equation (2), the deepest decoder features and the corresponding encoder of the layer are represented using the same features. For each subsequent deconvolution, the decoder features obtained are summed and calculated with the corresponding reverse attention enhancement features F_a^h. The decoder features fused after enhancing the feature representation ability of the encoder through reverse attention can better represent the geometric and semantic information of the point cloud.

$$F_d^h = \begin{cases} F_e^h, & if \quad h = 7 \\ conv(dconv(F_d^{h+1}) + F_a^h, otherwise \end{cases} \quad (2)$$

The point cloud features with a size of $N \times 32$ extracted through the backbone network will be passed into two parallel prediction branches, namely the offset prediction branch and the semantic prediction branch.

Compared to traditional skip connections that directly concatenate multi-scale features, the reverse attention mechanism explicitly addresses the critical issue of insufficient feature representation. Standard skip connections often inadvertently introduce low-level geometric noise into the decoder. In contrast, our reverse attention leverages the rich semantic context from higher-level encoder layers to dynamically filter and re-weight the lower-level features prior to fusion. By doing so, it effectively suppresses irrelevant background noise and highlights essential fine-grained geometric details. This guided fusion ensures a more robust and discriminative integration of both geometric and semantic information than simple concatenation.

As shown in Figure 6, the offset prediction branch is mainly composed of two layers of MLP, which is used to predict the spatial position offset between each point in the point cloud and the corresponding instance centroid. Firstly, the features are processed through the first MLP. Then, the processed features are normalized through Norm for data normalization and nonlinear factors are introduced through the ReLU activation function to improve the generalization ability of the model. After the second layer of MLP processing, a predicted center offset vector of size $N \times 3$ will be obtained.

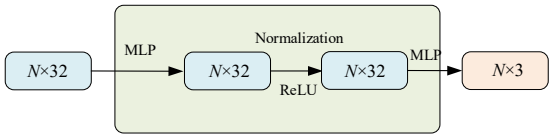

Figure 6. Schematic diagram of the offset prediction branch structure

The loss function for the constraint center bias vector in the offset prediction branch is supervised and trained through L_{shift} as shown in Equations (3) and (4).

$$L_{shift} = \frac{1}{\sum_i^N I(p_i \in P_{fg})} \cdot \sum_{i=1}^{N} L(p_i) \quad (3)$$

$$L(p_i) = \left\| \Delta x_i^{gt} - \Delta x_i^{pred} \right\| \cdot I(p_i \in P_{fg}) \quad (4)$$

Among them, I is the indicator function indicating whether the point belongs to any instance, P_{fg} representing the set of foreground points in the input scene, Δx_i^{gt} and Δx_i^{pred} respectively represent the true center offset vector and the predicted center offset vector of point i. During the model training process, through optimization L_{shift}, the offset prediction branch can accurately generate the prediction result of the center offset vector close to the true value.

Because background points are not considered in instance segmentation, background points such as walls and floors are ignored in L_{shift}. The noise prediction by ignoring background points also makes the center offset prediction more accurate. After the prediction of the center offset vector, the new instance center x_i^{shift} is calculated, making the points of the same instance closer in the multi-dimensional space. x_i^{shift} calculation formula is shown in Equation (5), where x_i^{origin} represents the original center coordinate and Δx_i^{pred} represents the center offset prediction vector.

$$x_i^{shift} = x_i^{origin} + \Delta x_i^{pred} \quad (5)$$

Semantic prediction predicts the semantic score of the semantic category to which the point cloud belongs. As shown in Figure 7, this branch is similar in structure to the offset prediction branch, both adopting a two-layer MLP structure and integrating normalization and ReLU activation functions. The core difference lies in that the output of this branch is the semantic score of each category corresponding to each point.

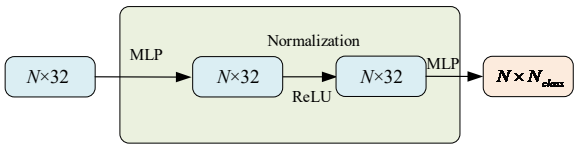

Figure 7. Schematic diagram of the semantic prediction branch structure

3.3. Instance aggregation module combined with semantic scores

The center offset vector can move the points in the original point cloud space to the direction of the instance center to which it belongs, and initially gather the point clouds in the original space around the instance center to which it belongs. Therefore, the Euclidean distance in the offset space and the semantic prediction label are used as the main basis for traditional point cloud instance clustering for instance aggregation.

Table 1. Point Clustering Algorithm Based on Instance-Center Offset and Semantic Label

Algorithm 1: point Clustering Algorithm Based on Instance-Center Offset and Semantic Label

Require:
Offset space point coordinate set $X = \{x_1, x_2, \ldots, x_N\} \in R^{N\times 3}$;
Maximum number of points per cluster N_θ
Semantic label set $S = \{s_1, s_2, \ldots, s_N\} \in R^N$;
Cluster search radius r;
Ensure: cluster set $C = \{c_1, c_2, \ldots, c_M\}$
Initialize a marker array $V = \{v_1, v_2, \ldots, v_N\}$ of length N with all values set to 0
Initialize an empty cluster results set C
for $i = 1$ to N **do**
  if s_i is background class **then**
    $v_i = 1$
  end if
end for
for $i = 1$ to N **do**
  if $v_i = 0$ **then**
    Initialize queue Q and an empty set
    $v_i = 1$, $Q.enqueue(i)$
    while Q is not empty **do**
      $i = Q.dequeue()$;
      for $j = 1$ to N **do**
        if $\|x_i - x_j\|_2 < r$ **and** $s_i == s_j$ **and** $v_j == 0$ **then**
          $\boldsymbol{Q.enqueue(j)}$, Add j *to* Cluster c
        end if
      end for
    end while
    if number of points in cluster $c > N_\theta$ **then**
      Add cluster c to result set C
    end if
  end if
end for
return cluster result set $C = \{c_1, c_2, \ldots, c_M\}$

The details of the point clustering algorithm based on instance center offset and semantic label are shown in Table 1, where N is the number of point clouds and M is the number of clusters. In this way, point clouds within the same instance can be clustered according to the distance in Euclidean space. However, the most serious problem of

this algorithm is that for point $\boldsymbol{i}$ and point $\boldsymbol{j}$ to be clustered to the same instance, not only the Euclidean distance between point clouds is required, but also the point pairs to be clustered must have completely consistent semantic label judgment. Such hard constraints may lead to biased instance segmentation results, especially when there are errors in semantic prediction, which will significantly affect the integrity of the clustering results.

To solve the problem of instance segmentation errors caused by incorrect semantic prediction, this paper adopts a soft clustering method combined with semantic scores to achieve the initial instance aggregation.

Compared with the traditional point clustering, the soft clustering algorithm adopted in this study does not use the semantic labels generated after hard one-hot encoding as the clustering conditions. Instead, by taking the semantic scores as the input of clustering, each point can be associated with multiple classes to alleviate the instance prediction errors caused by incorrect semantic prediction.

Specifically, in this paper, a semantic score is to choose high semantic score point sets for spatial aggregation. Since all the points in each point set during the traversal belong to the same class, fast clustering can be achieved by traversing all the points in the subset and aggregating them only when their geometric distances are less than $\boldsymbol{r}$, that is $\|x_i - x_j\|_2 < r$, when the condition is met. The overall algorithm flow is shown in Table 2.

The instance aggregation module adopted in this paper has the possibility of obtaining correct instance predictions even when there are incorrect semantic predictions. In the subsequent instance correction network, all instance suggestions in L are corrected. The incorrect instance suggestions are suppressed by predicting them as background classes, thereby generating the final point-by-point instance labels.

Table 2. Instance Clustering Algorithm with Semantic Scores

Algorithm 2: Instance Clustering Algorithm with Semantic Scores

Require:

Cluster search radius r;
Semantic score threshold μ;
Semantic size threshold N_θ
Offset space point coordinates $X = \{x_1, x_2, \ldots, x_N\} \in R^N$
Semantic scores $Score = \{S_1, S_2, \ldots, S_N\} \in R^N$,where $S_i = \{s_{i1}, s_{i2}, \ldots, s_{iN_{class}}\}$ represents the predicted scores of point i across semantic classes.

Ensure:

cluster set $L = \{C_1, C_2, \ldots, C_{N_{class}}\}$ where $C_i = \{c_1, c_2, \ldots, c_M\}$ denotes the grouped cluster for each of the N_{class} categories (maximum M clusters per category)
Initialize a partition set $P = \{p_1, p_2, \ldots, p_{N_{class}}\}$ of length N_{class}
for $classId = 1$ to N_{class} **do**
  if $classId$ is background class **then**
    continue
  end if
  for $i = 1$ **to** N **do**
    If Point $i'scores$ for $classId < \mu$ **then**
      continue
    end if
    Add point i to $p_{classId}$
  end for
end for
Initialize empty cluster result set $L = \{C_1, C_2, \ldots, C_{N_{class}}\}$
for $j = 1$ **to** P **do**
  if $length(p_j) = 0$ **then**
    continue
  end if
  Initialize empty cluster set C
  while $length(p_j)! = 0$ **do**
    Initialize queue Q and empty set c , $i \leftarrow p_j$.first()
    $Q.enqueue(i)$; Remove i from p_j {mark as processed}
    while Q is not empty **do**
      $i = Q.dequeue()$;
      for $l = 1$ **to** N **do**
        if $\|x_i - x_l\|_2 < r$ **then**
          $Q.enqueue(l)$, add l to cluster c, Remove l from p_j {Mark as processed}
          if cluster c size $> N_\theta$ **then**
            Add cluster c to result set C
          end if
        end if
      end for
    end while
    Add result set C to cluster collection L
  end while
end for
Return: final cluster set L

3.4. Instance correction network based on self-attention

In this paper, after instance aggregation, a self-attention-based instance correction network is added to correct the aggregation results, as shown in Figure 8. The point-by-point features generated by the backbone network pass through a self-attention-based feature extractor to enhance the features of the points. Then, feature fusion is carried out with each set in the instance suggestions to obtain the corresponding instance features for each instance suggestion. The extracted instance features are sent to a tiny U-Net structure with only two layers for feature transformation. Finally, instance correction is conducted jointly by the classification scoring branch and the mask scoring branch to filter out low-quality instance predictions, and finally point-by-point instance labels are generated.

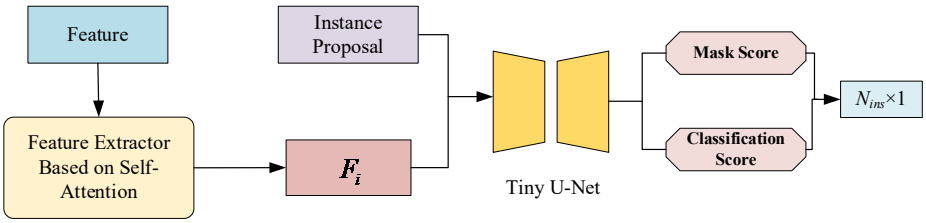

Figure 8. Instance Correction Network based on self-attention

Self-attention mechanism design

Self-attention operators can be divided into two types: scalar attention[26] and vector attention[27]. As shown in Equation (6), scalar attention determines the attention weights by calculating the scalar product between features.

$$y_i = \sum_{x_j \in \chi} \rho(\phi(x_i)^T \psi(x_j) + \delta)\alpha(x_j) \quad (6)$$

Among them, ϕ, ψ and α are the feature space transfor mation methods, δ is the geometric encoding module of sp atial coordinates, ρ is the feature standardization unit, x_i is the feature of the input point i, x_j is the feature of the neig hborhood points of point i. Specifically, the local neighbor hood χ for each point i is constructed using the k-Nearest Neighbors (k-NN) algorithm, where k is empirically set to 16. The scalar attention operator first transforms the featu res, then calculates the scalar product, and uses it as the att ention weight to aggregate the features.

The attention weights generated by the vector attention mechanism are vectors and can adaptively modulate a single feature channel. Its calculation method is shown in Equation (7)

$$y_i = \sum_{x_j \in \chi} \rho(\gamma(\beta(\phi(x_i), \psi(x_j)) + \delta)) \odot \alpha(x_j) \quad (7)$$

Among them, β is the relationship function, γ is the mapping function. The vector attention mechanism first calculates the relationship of the features through the relationship function, then obtains the attention vector through the mapping function, and finally multiplishes it element by element with the transformed features. Compared with the scalar attention mechanism, the vector attention mechanism can achieve more accurate feature modeling capabilities by differentially allocating the weight coefficients of each feature channel. This dynamic weight adjustment method can effectively improve the capture effect of data detail features.

The self-attention designed in this paper belongs to vector attention, as shown in Equation (8). Referring to the design of self-attention in Point Transformer[28], subtraction is adopted as the relational function β in this paper. Considering that, compared with semantic features, instance features also require the offset information of points to further determine the instance to which the points belong, in this paper, the instance center coordinates enhanced by the offset vector output by the offset prediction branch are taken as the geometric encoding information. After the MLP transformation, the relation function β transformation and α transformed features are added simultaneously to enhance the instance features of the points. Enable it to be more effectively used for point-by-point instance segmentation. Figure 9 shows the transformation process of the instance features of the points after self-attention enhancement in this paper.

$$MT = \gamma\left(\left(\phi(f_i) - \psi(f_i^k)\right) + MLP(x_i^{shift})\right)$$

$$MA = \alpha(f_i^k) + MLP(x_i^{shift}) \quad (8)$$

$$F_i = \sum_k^K softmax(MT \odot MA)$$

Among them, the features f_i representing the input points p_i, the features f_i^k representing the neighbourhood points of the input points p_i, and the position encoding uses the offset vector x_i^{shift} fused with the original coordinates after the offset prediction branch output processed by MLP, $\odot$ representing the element-by-element product operation, F_i is the final output feature.

To ensure efficiency in large-scale scenes, our self-attention module operates locally rather than globally. By restricting the attention computation for each of the N points to its k local neighbors, the theoretical complexity is significantly reduced from quadratic $\mathcal{O}(N^2)$ to $\mathcal{O}(N \cdot k)$. Since $k \ll N$ is a small constant, the overall complexity scales linearly, $\mathcal{O}(N)$, demonstrating its practical applicability for large-scale point clouds.

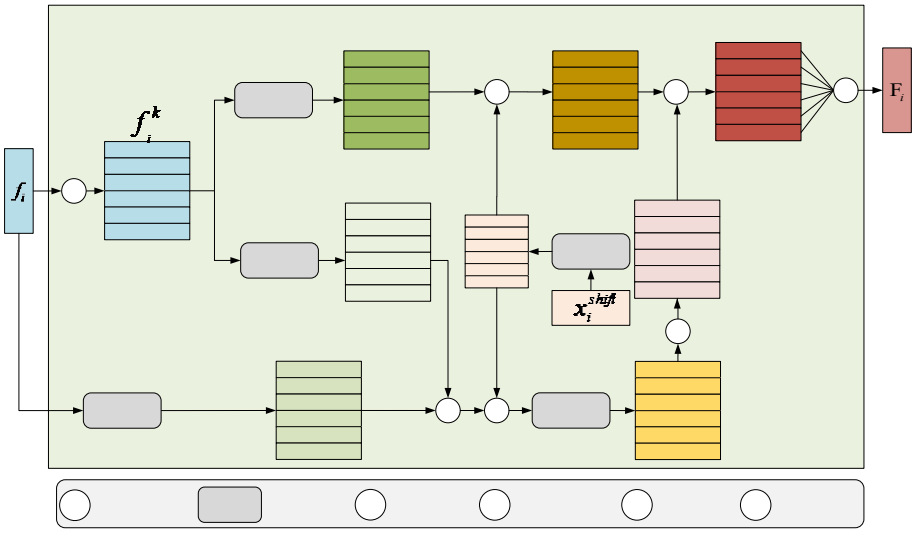

Figure 9. Schematic Diagram of Feature Extraction Based on Self-attention

Dual-branch evaluation mechanism

The existing instance segmentation methods based on s emantic prediction tend to rely on the semantic prediction of objects to determine their categories. However, a potent ial problem of this method is that it may wrongly take som e actually inaccurate or noisy semantic predictions as the b asis for judgment, thereby affecting the accuracy of the fin al category labels. Therefore, after extracting the instance features of self-attention in this paper, a dual-branch evalu ation mechanism of classification prediction scoring and mask prediction scoring is designed.

The classification prediction branch is used for categor y prediction. This branch first aggregates the features of al l points in the instance by means of attention aggregation, and then predicts the classification score of the instance th rough MLP. The classification label generated by one-hot encoding is used as the category annotation of the instanc e. Compared with semantic labels, using classification lab els to uniformly label instances can filter out category inte rference caused by inaccurate local features or environme ntal noise more effectively, and provide more reliable pred iction results. The loss function used in the classification s coring branch is the cross-entropy loss function L_{class}, as s hown in Equation (9).

$$L_{class} = \frac{1}{K}\sum_{k=1}^{K}\sum_{i=1}^{C} y_{ki}\, log(p_{ki}) \tag{9}$$

Among them, K represents the number of instance sets, C represents the number of categories, y_{ki} and p_{ki} respectively represent the true label of the sample k in the category i and the predicted probability.

The function of the mask prediction scoring branch is to screen the instance prediction results with high confidence. This branch adopts a two-layer and multi-layer perceptron architecture to process the input features and output the scoring vector with a dimension of $N \times 1$. For each predicted instance set, by calculating the intersection-to-union ratio (IoU) between them and the true instance set, instance samples that do not reach the set IoU threshold Thr_{iou} are regarded as negative samples. The threshold set in this paper is 0.5. Meanwhile, only the positive sample set is involved in the training of mask prediction because negative samples of poor quality are difficult to optimize the model training. The loss of mask prediction L_{mask} is shown in Equation (10).

$$L_{mask} = \frac{1}{\sum_{i=1}^{K} I(iou_i > Thr_{iou}) \cdot N_i} \cdot \sum_{i=1}^{K} (I(iou_i > Thr_{iou}) \cdot \sum_{j=1}^{N_i} (y_j \cdot \log(p_j) + (1-y_j) \cdot \log(1-p_j))) \tag{10}$$

Here, K represents the number of instance sets, N_i represents the number of points contained in the instances, $I(iou_i) > 0.5$ is the indicator function, is the IoU between the predicted instance set and the corresponding target true value set, y_i and p_j respectively represent the mask prediction results and mask true values of the point j in the instance i .

After the mask prediction is completed, the parts belonging to the background points in the mask are filtered. The remaining foreground features are sent to a layer of MLP with Sigmoid activation for instance confidence prediction. In the instance confidence prediction stage, all instance sets with scores lower than the threshold Thr_{score} will be filtered again. For each instance set, the product of the score $scores_c$ in its classification prediction, the mask prediction, and the calculated truth set of the target instance iou_i is taken as the confidence score of that instance set. The predicted loss of confidence level L_{score} is shown as Equation (11).

$$L_{score} = -\frac{1}{K}\cdot\sum_{i=1}^{K}(iou_i \cdot score_{ci}\cdot\log(score_i) + (1-iou_i \cdot score_{ci})\cdot\log(1-score_i)) \quad (11)$$

Among them, $score_{ci}$ represents the classification prediction score of the instance, and $score_i$ is the confidence prediction score of the instance i.

4. Experiments

4.1. Dataset

In the field of 3D point cloud understanding, several public datasets are available to evaluate algorithm performance, covering a variety of indoor and outdoor scenes. For example, SemanticKITTI[29] is a well-known large-scale dataset focused on semantic scene understanding of LiDAR sequences for autonomous driving. Concurrently, large-scale datasets for indoor environments, such as S3DIS[30], also exist.

To evaluate the semantic segmentation performance of the proposed algorithm, this study selects the large-scale indoor public dataset S3DIS, released by Stanford University. The dataset comprises complete scans of six building areas (Area 1 to Area 6), containing over 270 million 3D points with instance-level semantic annotations, covering 50 types of indoor scenes. Each point includes spatial coordinate (X, Y, Z) and RGB color information. The dataset consists of 13 semantic categories, such as ceiling, floor, and table.

4.2. Evaluation indicators

To validate the effectiveness of our method, we conducted experiments using two evaluation protocols: testing on Area 5 as the validation set, and six-fold cross-validation across all areas. For evaluation metrics, we adopt mean Precision (mPrec), mean Recall (mRec), mean Coverage (mCov), and mean Average Precision (mAP). As follows specifically.

mPrec

The average precision mPrec is a commonly used evaluation metric for the S3DIS dataset. This metric is derived based on the precision Prec. The definition of precision is as shown in Equation (12).

$$Prec = \frac{TP}{TP+FP} \quad (12)$$

Here, TP and FP represent true examples and false positives respectively. The precision can reflect the reliability of the model's prediction results. In instance segmentation, the average precision mPrec is usually determined by setting different IoU thresholds to assess the accuracy of the prediction, thereby obtaining the evaluation indicators under different IoU threshold conditions. The calculation of IoU is shown in Equation (13), defined as the proportion of the intersection of the predicted set and the true value set to their union. The larger the value of this indicator, the more significant the spatial overlap between the model's prediction result and the true annotation.

$$IoU = \frac{TP}{TP+FP+FN} \quad (13)$$

mRec

The average recall rate mRec is calculated based on the recall rate Recall. The definition of recall rate is as shown in Equation (14).

$$REC = \frac{TP}{TP+FN} \quad (14)$$

The recall rate can reflect the model's ability to correctly identify positive samples. The average recall rate (mRec) is still an indicator obtained by averaging the recall rates (Recall) of various categories after setting a given IoU threshold. In this paper, with the IoU threshold T_{iou} set to 0.5, the average recall rate $mRec_{50}$ under this threshold is obtained.

mCov

The average coverage mCov is derived based on the coverage Cov. The definition of coverage is as shown in Equation (15), and it is mainly used to measure the average value of IoU matching between the true instance of the example and the predicted instance. For each real instance in each category, by finding the predicted instance with the largest IoU, the average value of these maximum IoUs is calculated.

$$Cov = \sum_{k=1}^{K}\sum_{i=1}^{N_{ins}}\frac{1}{N_{ins}}\max_j IoU(I_{k,i}^{g}, I_{k,j}^{p}) \quad (15)$$

And K indicates the number of categories, N_{ins} is the number of actual instances, $I_{k,i}^{g}$ and $I_{k,j}^{p}$ respectively represent the true value instances i and predicted instances j of the category k.mCov is obtained by averaging the total Cov values calculated for each category. Additionally, considering that the number of points in different categories is uneven, this paper also introduces mWCov, which is obtained through weighted averaging while taking into account the weights of each category

mAP

In the ablation experiment section of this chapter, the metric of mean average precision (mAP) was also introduced. This metric takes into account both recall rate and precision rate comprehensively. For a single category, the precision is calculated under different recall rates, and the area under the PR (Precision-Recall) curve is obtained. In the PR curve, the precision is usually plotted on the vertical axis and the recall rate on the horizontal axis. The closer the curve is to the upper right corner, the higher the precision and recall rates of the model at different thresholds, indicating better performance. The calculation method of mAP is shown in Equation (16), in this paper, $mAP_{50}$was calculated when T_{iou} was set to 0.5.

$$mAP = \frac{1}{K}\sum_{k=1}^{K}\frac{1}{N_k}\sum_{i=1}^{N_k} Precision(i)\cdot recall(i) \quad (16)$$

In addition, on the basis of mAP, more stringent indicators AP have also been introduced. The AP calculation method is to start with a threshold T_{iou} of 0.5,

with a step size of 0.05, until the average value of all mAPs up to 0.95 is obtained, as shown in Equation (17).

$$AP = \frac{1}{10}(mAP_{50} + mAP_{55} + \ldots + mAP_{95}) \quad (17)$$

4.3. Implementation details

The proposed model was implemented in PyTorch. During training, the model was configured with the following parameters: the number of training epochs was set to 128, the learning rate was 0.004, and the batch size was 4. The Adam optimizer was used. In the soft clustering process, the semantic score threshold μ was set to 0.2, the IoU threshold Thr_{iou} for matching predicted and ground-truth instances was set to 0.5, and the confidence score threshold Thr_{score} was set to 0.1

4.4. Comparative experiments

Performance evaluation on the Area 5.

Table 3 presents the comparative experimental results on the Area 5 test set. The best results are highlighted in bold, while the second-best results are underlined.

According to the experimental results, our method achieved 74.1%, 67.3%, 67.3%, and 69.1% in terms of mean Precision, mean Recall, mean Coverage, and mean Weighted Coverage, respectively — all surpassing the performance of the compared models. These results demonstrate the effectiveness and technical advantages of the proposed method, offering a new perspective for research on point cloud instance segmentation.

Table 3. Comparative Experimental Results on the Area 5 Test Set (Unit: %)

Model	$mPrec$	$mRec_{50}$	$mCov$	$mWCov$
SGPN[31]	36.0	28.7	32.7	35.5
ASIS[32]	55.3	42.4	44.6	47.8
JSPNet[11]	59.6	48.0	50.7	53.5
HAIS[9]	71.7	65.0	64.3	66.0
SoftGroup[12]	<u>73.6</u>	<u>66.6</u>	<u>66.1</u>	<u>68.0</u>
Ours	**74.1**	**67.3**	**67.3**	**69.1**

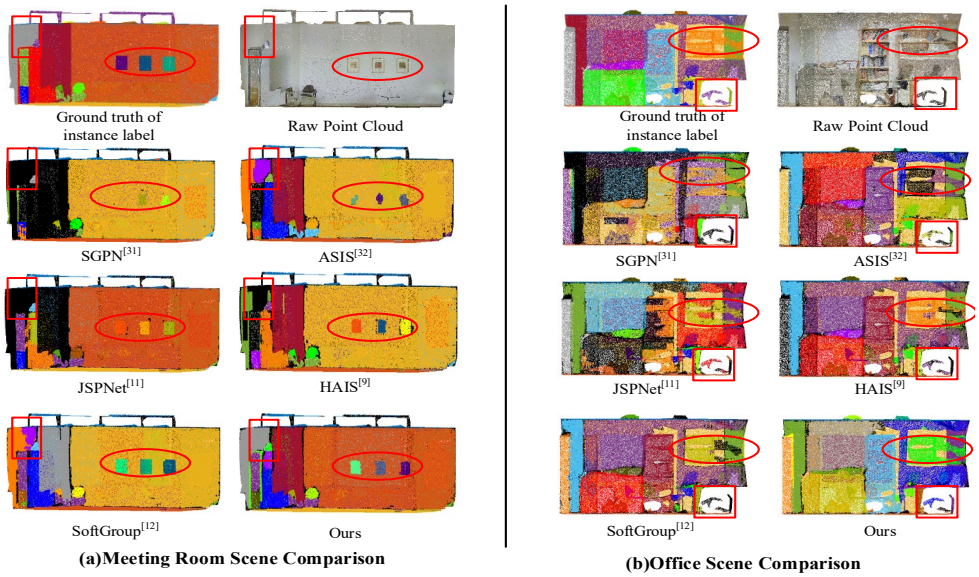

Figure 10. Instance Segmentation Comparison on Area 5 Scenes

Cross-Area robustness evaluation

Table 4 presents the comparative results under six-fold cross-validation. The best results are highlighted in bold, while the second-best results are underlined.

The experimental results show that the proposed method consistently outperforms several state-of-the-art models across various evaluation metrics under six-fold cross-validation. Specifically, our model achieves 76.8% mean Precision, 72.3% mean Recall, 70.7% mean Coverage, and 72.3% mean Weighted Coverage. Compared to the second-best model, SoftGroup, our method achieves improvements of 1.5%, 2.5%, 1.4%, and 0.6% on these four metrics, respectively. These results further demonstrate the effectiveness of the proposed method in

point cloud instance segmentation and highlight its strong generalization ability and robustness.

Table 4. Comparative Experimental Results under Six-Fold Cross-Validation (Unit: %)

Model	$mPrec_{50}$	$mRec_{50}$	$mCov$	$mWCov$
SGPN[31]	38.2	31.2	37.9	40.8
ASIS[32]	63.6	47.5	51.2	55.1
JSPNet[11]	66.5	55.0	54.9	58.8
HAIS[9]	73.2	69.4	67.0	70.4
SoftGroup[12]	75.3	69.8	69.3	71.7
Ours	**76.8**	**72.3**	**70.7**	**72.3**

Performance comparison

To further intuitively demonstrate the effectiveness of the proposed model in point cloud instance segmentation, we select two representative scenes from Area 5 and visualize the instance segmentation results for comparison. As shown in Figure 10, we present the segmentation results in a meeting room and an office scene using our proposed model, alongside comparisons with SGPN, ASIS, HAIS, JSPNet, and SoftGroup.

By examining the instance segmentation comparison results, it can be observed that the proposed model achieves superior performance in capturing fine-grained details within the scene. Moreover, it demonstrates better instance integrity and clearer distinction between different instances compared to other methods.

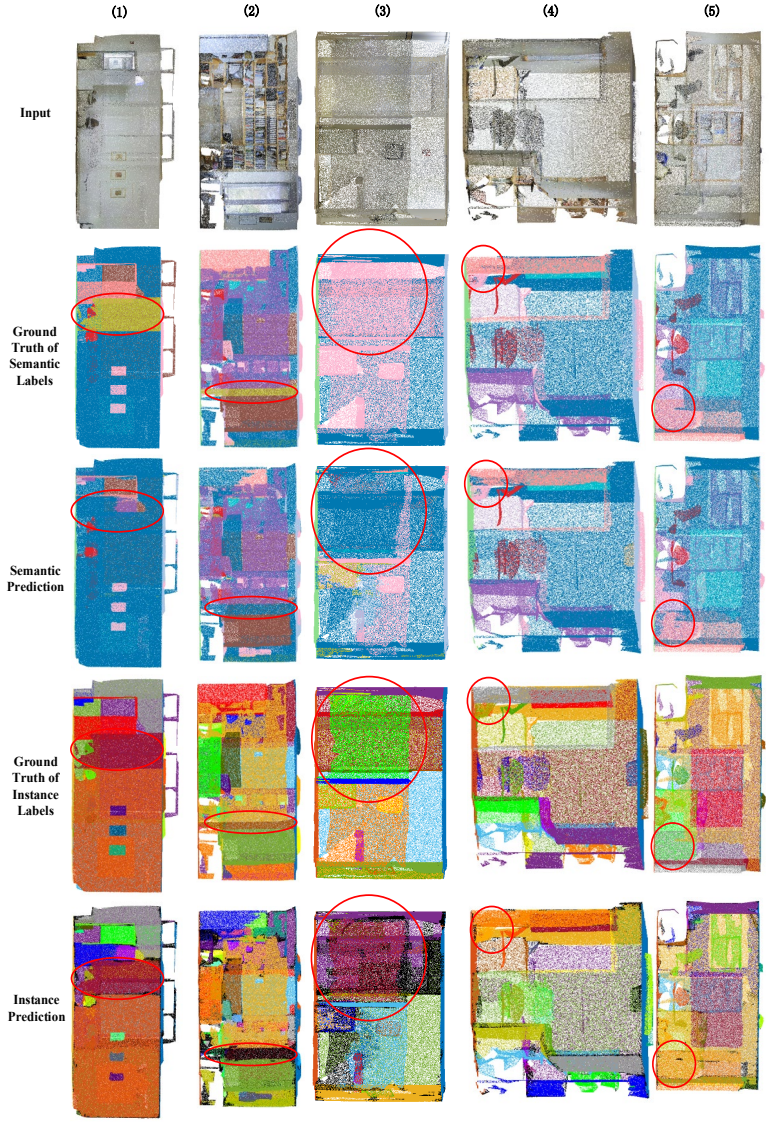

Figure 11. Visualization of Segmentation Results on the S3DIS Area 5 Indoor Real-World Dataset

Analyse of Instance Refinement Effectiveness

To more intuitively demonstrate the instance refinemen t capability of the proposed model and verify its ability to accurately predict instances even when semantic segmenta tion errors occur, a visual analysis of the model's instance refinement performance was conducted. Figure 11 illustrat es the visualization of instance refinement results in select ed scenes from Area 5.

Observing the visualization results, it can be seen that:

- (1) and (2) scenes contain columns (highlighted by red circles) that were incorrectly predicted as walls in the semantic segmentation, yet the instance prediction successfully identified them as separate column instances.
- (3) in the highlighted red box of the third scene, the ground truth category is clutter, which was mistakenly predicted as the same semantic category as the surrounding walls during semantic segmentation, but was still correctly separated as an individual instance in the instance prediction.
- (4) and (5) scenes contain doors (marked by red circles) that were misclassified as walls in semantic segmentation, while the instance prediction was able to preserve the door as a whole instance, effectively mitigating the error propagation from incorrect semantic labels to instance segmentation.
- These visual improvements align directly with the quantitative ablation results in Table 5. The correct segmentation of misclassified instances (e.g., columns and doors) in Figure 11 is attributed to the soft clustering strategy (Group 3) mitigating semantic errors, and the self-attention network (Group 8) further refining geometric boundaries. 4.5. Ablation experiments.

4.5. Ablation Experiments

To systematically verify the contribution of each module proposed in this paper, including the backbone network with reverse attention mechanism, the feature extractor with self-attention mechanism, the dual scoring branches, and the soft clustering with semantic scores, to improving point cloud instance segmentation, ablation experiments were conducted by removing some modules and scoring branches. All experiments were carried out using the S3DIS dataset and compared through six-fold cross-validation.

Sub-module ablation experiment

The experimental setup of this group is designed to veri fy the effectiveness of the reverse attention mechanism, th e soft clustering module, and the self-attention mechanism in the model presented in this paper. Corresponding exper imental groups were set up on the S3DIS dataset and evalu ated through a six-fold cross-validation method. The resul ts of the ablation experiments for different strategy combi nations are shown in Tables 5. In each experiment, the con fidence score is jointly determined by the double-score bra nch combining the mask prediction and classification scor e proposed in this paper. The "×" in the table indicates th e removal of the relevant module, and the " √ " indicates t he addition of the relevant module. The experiments witho ut the soft clustering combined with semantic scores use tr aditional point clustering for instance clustering.

Analysis of the ablation experiment results shows that t he introduction of the reverse attention mechanism, self-at tention mechanism, and instance soft clustering proposed i n this paper all have an enhancing effect on the model perf ormance.

The group 8 that used the combined application of the three modules achieved the best results compared to the other experimental groups. This indicates that the instance segmentation model proposed in this paper, which combines the application of the above three modules, can achieve more accurate instance predictions

Table 5. Comparison of ablation experiment results of the S3DIS six-fold cross-validation sub-module (unit: %)

Group	Reverse attention mechanism	Soft clustering	Self-attention mechanism	$mPrec_{50}$	$mRec_{50}$	$mCov$	$mWCov$	AP	mAP_{50}
1	×	×	×	65.7	63.9	64.3	64.6	50.9	61.3
2	√	×	×	66.9	65.3	62.9	65.2	51.3	61.8
3	×	√	×	69.2	66.8	65.1	67.7	52.1	62.9
4	×	×	√	67.9	64.3	64.5	66.2	51.1	61.7
5	√	√	×	73.8	69.9	67.2	70.9	53.9	67.1
6	√	×	√	72.7	67.4	67.9	69.8	52.6	66.5
7	×	√	√	74.9	69.7	68.6	70.9	53.7	68.3

8	√	√	√	**76.8**	**72.3**	**70.7**	**72.3**	**55.8**	**70.1**

Theoretically, the performance drop from Group 8 to Group 7 demonstrates the specific value of the reverse attention mechanism. By effectively filtering low-level geometric noise using high-level semantics, this mechanism specifically enhances the segmentation of small objects and fine-grained boundaries, which directly corroborates the superior detail preservation observed in the qualitative results of Figure 10.

Score Branch Abandonment Experiment

In order to verify the effectiveness of using the classification prediction results as the category labels and adopting the dual-branch evaluation mechanism to jointly determine the confidence score in the model proposed in this paper for instance prediction, corresponding ablation experiments were conducted on the S3DIS dataset for evaluation. In all the grouped experiments, the reverse attention mechanism, soft clustering, and self-attention mechanism were also incorporated simultaneously. Table 6 shows the ablation experiments using different score branch combinations.

Group 1 indicates using the classification prediction results as the category labels, but using the mask prediction scores alone for confidence scoring; Group 2 indicates using the semantic prediction results as the category labels, and using the mask prediction scores alone for confidence scoring; Group 3 adopts the dual-branch evaluation mechanism proposed in this paper, that is, using the classification prediction results as the category labels for instance prediction, and combining classification scoring with mask scoring to jointly perform confidence scoring. It can be observed that the combination of using the classification prediction results as instance labels and the dual scoring branches for confidence scoring can achieve the best effect.

Table 6. Experimental results of different scoring branch ablation (unit: %)

Group	Classification prediction	Mask prediction	$mPrec_{50}$	$mRec_{50}$	$mCov$	$mWCov$	AP	mAP_{50}
1	√	×	72.3	69.3	69.1	70.1	52.9	65.8
2	×	√	71.8	68.9	68.6	69.7	52.3	62.7
3	√	√	**76.8**	**72.3**	**70.7**	**72.3**	**55.8**	**70.1**

5. Conclusions

This paper proposes an innovative point cloud instance segmentation framework that fundamentally addresses the theoretical bottlenecks of strong task coupling and insufficient feature extraction in existing bottom-up methods. Technically, we achieve this through three key innovations: a reverse-attention feature extraction network for multi-scale context fusion, a soft clustering strategy to mitigate semantic errors, and a self-attention-based refinement network coupled with a dual-branch scoring mechanism. Experimentally, our method achieves significant accuracy improvements over representative baseline methods on the S3DIS dataset and demonstrates remarkable robustness in complex, ambiguous scenes. Despite these advancements, the self-attention mechanism inevitably introduces certain computational overhead. Future work will focus on designing lightweight attention modules to improve real-time efficiency and extending the framework to sparse, large-scale outdoor datasets.

Acknowledgement

This research is supported by the Joint Funds of the National Natural Science Foundation of China for Railway Basic Research under Grant No. U2468201; the Science and Technology Program Projects of the Sichuan Provincial Department of Science and Technology under Grant Nos. 2025ZDZX0009 and 2026YFHZ0225.

References

[1] Guo Y, Wang H, Hu Q, et al. Deep learning for 3d point clouds: A survey[J]. IEEE transactions on pattern analysis and machine intelligence, 2020, 43(12): 4338-4364.

[2] Rani A, Ortiz-Arroyo D, Durdevic P. Advancements in point cloud-based 3D defect classification and segmentation for industrial systems: A comprehensive survey[J]. Information Fusion, 2024, 112: 102575.

[3] Yasir S M, Sadiq A M, Ahn H. 3D Instance Segmentation Using Deep Learning on RGB-D Indoor Data[J]. Computers, Materials and Continua, 2022, 72(3): 5777-5791.

[4] Nunes L, Chen X, Marcuzzi R, et al. Unsupervised class-agnostic instance segmentation of 3d lidar data for autonomous vehicles[J]. IEEE Robotics and Automation Letters, 2022, 7(4): 8713-8720.
[5] Li X, Ding H, Yuan H, et al. Transformer-based visual segmentation: A survey[J]. IEEE transactions on pattern analysis and machine intelligence, 2024.
[6] Schult J, Engelmann F, Hermans A, et al. Mask3D: Mask Transformer for 3D Semantic Instance Segmentation[C]//2023 IEEE International Conference on Robotics and Automation (ICRA). IEEE, 2023: 8216-8223.
[7] Brunklaus M, Kellner M, Reiterer A. Three-Dimensional Instance Segmentation of Rooms in Indoor Building Point Clouds Using Mask3D[J]. Remote Sensing, 2025, 17(7): 1124.
[8] Liu C, Furukawa Y. MASC: Multi-scale affinity with sparse convolution for 3D instance segmentation. arXiv preprint. 2019; arXiv:1902.04478.
[9] Chen S, Fang J, Zhang Q, et al. Hierarchical aggregation for 3d instance segmentation[C]//Proceedings of the IEEE/CVF International Conference on Computer Vision. 2021: 15467-15476.
[10] Cong B G, Wang X H, Zhao X, et al. 3D-SDIS: enhanced 3D instance segmentation through frequency fusion and dual-sphere sampling[J]. The Visual Computer, 2025: 1-14.
[11] Chen F, Wu F, Gao G, et al. JSPNet: Learning joint semantic & instance segmentation of point clouds via feature self-similarity and cross-task probability[J]. Pattern Recognition, 2022, 122: 108250.
[12] Vu T, Kim K, Luu T M, et al. Softgroup for 3d instance segmentation on point clouds[C]//Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2022: 2708-2717.
[13] Graham B, Engelcke M, Van Der Maaten L. 3d semantic segmentation with submanifold sparse convolutional networks[C]//Proceedings of the IEEE conference on computer vision and pattern recognition. 2018: 9224-9232.
[14] Deng H, Chen S, Zhu X, et al. EP-Net: Improving Point Cloud Learning Efficiency Through Feature Decoupling[J]. IEEE Transactions on Instrumentation and Measurement, 2024.
[15] Du J, Cai G, Wang Z, et al. MTCloud: Multi-type convolutional linkage network for point cloud instance segmentation[J]. Expert Systems with Applications, 2025, 270: 126432.
[16] Hegde S, Gangisetty S. PIG-Net: Inception based deep learning architecture for 3D point cloud segmentation[J]. Computers & Graphics, 2021, 95: 13-22.
[17] Vanian V, Zamanakos G, Pratikakis I. Improving performance of deep learning models for 3D point cloud semantic segmentation via attention mechanisms[J]. Computers & Graphics, 2022, 106: 277-287.
[18] Guo S, Cai J, Hu Y, et al. LCASAFormer: Cross-attention enhanced backbone network for 3D point cloud tasks[J]. Pattern Recognition, 2025, 162: 111361.
[19] Hong F, Kong L, Zhou H, et al. Unified 3d and 4d panoptic segmentation via dynamic shifting networks[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2024, 46(5): 3480-3495.
[20] Xv J, Deng F. 3D point cloud instance segmentation considering global shape contour constraints[J]. Remote Sensing, 2023, 15(20): 4939.
[21] Chen S, Fang J, Zhang Q, et al. Hierarchical aggregation for 3d instance segmentation[C]//Proceedings of the IEEE/CVF International Conference on Computer Vision. 2021: 15467-15476.
[22] Shelhamer E, Long J, Darrell T. Fully Convolutional Networks for Semantic Segmentation[J]. IEEE transactions on pattern analysis and machine intelligence, 2017, 39(4): 640-651.
[23] Ibtehaz N, Rahman M S. MultiResUNet: Rethinking the U-Net architecture for multimodal biomedical image segmentation[J]. Neural networks, 2020, 121: 74-87.
[24] Hazer A, Yildirim R. Deep learning based point cloud processing techniques[J]. IEEE Access, 2022, 10: 127237-127283.
[25] Shuai H, Xu X, Liu Q. Backward attentive fusing network with local aggregation classifier for 3D point cloud semantic segmentation[J]. IEEE Transactions on Image Processing, 2021, 30: 4973-4984.
[26] Vaswani A, Shazeer N, Parmar N, et al. Attention is all you need[J]. Advances in neural information processing systems, 2017, 30.
[27] Zhao H, Jia J, Koltun V. Exploring self-attention for image recognition[C]//Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2020: 10076-10085.
[28] Zhao H, Jiang L, Jia J, et al. Point transformer[C]//Proceedings of the IEEE/CVF international conference on computer vision. 2021: 16259-16268..
[29] Behley J, Garbade M, Milioto A, et al. Towards 3D LiDAR-based semantic scene understanding of 3D point cloud sequences: The SemanticKITTI Dataset[J]. The International Journal of Robotics Research, 2021, 40(8-9): 959-967.
[30] Armeni I, Sener O, Zamir A R, et al. 3d semantic parsing of large-scale indoor spaces[C]//Proceedings of the IEEE conference on computer vision and pattern recognition. 2016: 1534-1543.
[31] Wang W, Yu R, Huang Q, et al. Sgpn: Similarity group proposal network for 3d point cloud instance segmentation[C]//Proceedings of the IEEE conference on computer vision and pattern recognition. 2018: 2569-2578.
[32] Wang X, Liu S, Shen X, et al. Associatively segmenting instances and semantics in point clouds[C]//Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2019: 4096-4105.