MUSF-SSA: Multi-Scale Umbrella Feature with Spatial Self-Attention Model for Semantic Segmentation of Point Clouds

Linfu Xie[1, 2], Rutao Zhang[1, 2], Tianyi Xu[2], Weixi Wang[1, 2, *], Xiaoming Li[1, 2], Shengjun Tang[1, 2], Renzhong Guo[1, 2]

[1]State Key Laboratory of Subtropical Building and Urban Science, Shenzhen University, Shenzhen, Guangdong, 518060, China
[2]Research Institute for Smart Cities, School of Architecture and Urban Planning, Shenzhen University, Shenzhen 518060, China-
(linfuxie,wangwx, lixming, shengjuntang, guorz)@szu.edu.cn, (2410114011, xutianyi21)@email.szu.edu.cn

Keywords: Point clouds; Semantic segmentation; Feature extraction; Machine learning; Spatial self-attention.

Abstract

Efficiently extracting discriminative local features and exploiting long-range spatial correlations remain critical challenges in point cloud semantic segmentation. Existing methods often struggle to balance complex surface topology resolution with computational efficiency, frequently ignoring critical spatial correlations during training. To address these limitations, this paper proposes the Multi-Scale Umbrella Feature Network with Spatial Self-Attention (MUSF-SSA) to efficiently encode irregular 3D surface geometry. MUSF-SSA employs a k-d tree search for rapid neighboring point identification, followed by multi-scale umbrella feature construction to integrate key information across various spatial scales. An encoder-decoder structure is introduced to further refine these features, while a spatial self-attention mechanism explicitly models long-range spatial correlations. Evaluations on the S3DIS dataset demonstrate that MUSF-SSA achieves a mean Intersection over Union (mIoU) of 74.8%, a mean Accuracy (mAcc) of 82.9%, and an Overall Accuracy (OA) of 91.2%, surpassing state-of-the-art (SOTA) methods with comparable parameter complexity.

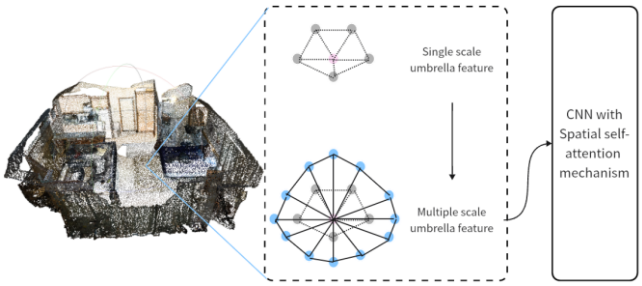

Fig. 1. An overview of points cloud segmentation with MUSF.

1. Introduction

Point cloud semantic segmentation partitions 3D points into meaningful subsets for complex scene understanding (Eisl et al., 2025). Current data-driven paradigms have transitioned from traditional methodologies (Wang et al., 2023) to neural networks, primarily multi-view, voxel-based, and point-based architectures (Guo et al., 2020; Landrieu and Simonovsky, 2018). While multi-view (Jaritz et al., 2019) and voxel-based (Li et al., 2023; Ye et al., 2023) methods achieve robust accuracy, they suffer from prohibitive computational overhead or inherent spatial resolution limits (Ye et al., 2022).

Consequently, computationally efficient point-based methods dominate raw point cloud processing. Following the pioneering direct processing of unstructured points by PointNet and PointNet++ (Qi et al., 2017a; Qi et al., 2017b), subsequent advancements enhanced local geometric feature extraction through CNNs (Thomas et al., 2019; Wu et al., 2019), Graph Neural Networks (Wang et al., 2019; Shi et al., 2020), self-attention models (Zhao et al., 2021; Yue et al., 2025; Wang et al., 2023), and recent multi-modal LLM-guided approaches (R. Xu et al., 2025; Diao et al., 2025).

Despite these advancements, critical limitations persist in fundamental point cloud representation. Existing "black box" models (Gawlikowski et al., 2022) rely on rudimentary initial features (e.g., coordinates or normals) that are insufficient for complex 3D surface topologies (Ceccarelli et al., 2025). Integrating robust geometric priors remains essential for boosting model generalization (Luo et al., 2025). Furthermore, current algorithms suffer from low construction efficiency, poor robustness to density variations, and a frequent failure to model long-range spatial correlations during training (Ran et al., 2022; Fujiwara et al., 2020; Nezhadarya et al., 2020).

To address these limitations, drawing on RepSurf (Ran et al., 2022), we propose the Multi-Scale Umbrella Feature Network with Spatial Self-Attention (MUSF-SSA) to efficiently encode irregular 3D surfaces and explicitly model spatial dependencies (Fig. 1). Our primary contributions are fourfold:

1. A multi-scale umbrella surface representation that extends standard Umbrella RepSurf (Ran et al., 2022) to resolve complex topological ambiguities across varying spatial scopes.
2. A multi-scale umbrella feature fusion optimization algorithm for adaptive geometric encoding.
3. A lightweight MLP-based encoder-decoder module for hierarchical feature dimensionality refinement.
4. An integrated spatial self-attention mechanism that leverages multi-layer perceptrons to explicitly model spatial correlations among local geometric features.

2. II. Overview of the proposed method

2.1 Overall process of the proposed method

As illustrated in Figure 2, the MUSF-SSA comprises a multi-scale umbrella feature construction module and a hierarchical feature extraction module. It processes raw 7-dimensional point cloud data (3D coordinates, RGB, and semantic labels) in batches. In the feature construction layer, a k-d tree retrieves neighboring points,

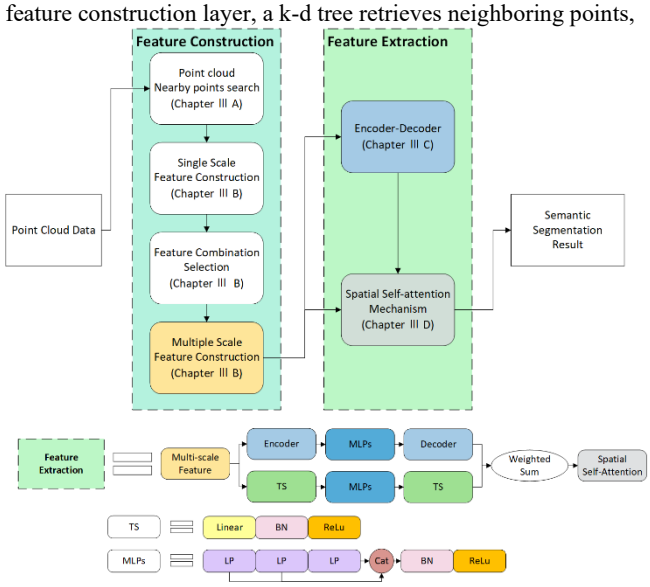

Figure 2. The overall workflow of the proposed model.

which are sorted by Euclidean distance and segmented into multiple spatial scales. Within each scale, points are transformed into a spherical coordinate system, ordered counterclockwise by azimuth, and used to construct sequential triangular surfaces for umbrella feature extraction.To unify dimensions and mitigate noise, candidate features undergo min-max normalization. We compute feature variance per scale and concatenate the three lowest-variance scales to form the final multi-scale umbrella features, effectively filtering out disordered neighborhoods that degrade discriminability.These final features feed into two parallel MLP-based branches: a feature extraction branch for dimensionality upscaling and an encoder-decoder branch. Their shallow and deep features are fused via weighted summation and reduced by additional MLPs to efficiently retain critical information.The refined features then enter the spatial self-attention module to explicitly model long-range spatial dependencies. Features are mapped into Query, Key, and Value matrices, where dot-product attention scores weight the final feature output. Finally, a fully connected layer performs per-point semantic classification to complete the segmentation task.

2.2 Baseline model introduction

Raw point features often lack the local discriminative power required for complex, highly irregular surface analysis (Fujiwara et al., 2020). To address this, Triangular RepSurf (Ran et al., 2022) computes explicit geometric features by constructing local triangular surfaces. Building on this framework, Umbrella RepSurf (Ran et al., 2022) is adopted as the core baseline for our MUSF-SSA model. Inspired by umbrella curvature (Foorginejad et al., 2014), it extends the triangular representation by aggregating adjacent facets in a local neighborhood. This dimensionally stable, rotation- and pose-invariant prior captures irregular topologies significantly better than conventional normal-based descriptors. Using this robust representation as a foundation, our MUSF-SSA optimizes it via multi-scale construction and explicitly models spatial feature correlations with an integrated self-attention mechanism.

3. Key Methods for MUSF

Despite its advantages, the original umbrella feature's representation degrades in messy or incomplete point clouds, suffers from slow construction, and ignores spatial feature correlations (Ran et al., 2022). To overcome these issues, we employ a k-d tree to accelerate construction and propose an optimized multi-scale umbrella feature. Parallel MLPs and an encoder-decoder structure then extract hierarchical features. Finally, a spatial self-attention mechanism explicitly captures spatial relations, outputting segmentation results via a fully connected layer.

3.1 Detailed explanation of process

Standard k-nearest neighbors (KNN) searches for initial point cloud feature construction incur high computational costs, significantly impeding training and testing efficiency. To accelerate this process, we replace KNN with a k-d tree neighbor search algorithm.

The k-d tree recursively partitions the dataset along selected spatial dimensions to construct corresponding tree nodes (Brown et al., 2014). Once built, efficiently traversing and backtracking the tree from the root node enables rapid retrieval of adjacent points, substantially improving overall feature construction speed.

3.2 Construction and combination optimization of multi-scale umbrella features

We find that in the process of constructing umbrella features, the original model only uses a small number of neighborhood points to establish umbrella features in the minimum scale space. In actual situations, especially for locations where the nature or structure of the point cloud changes, umbrella features in a single scale space may not be able to represent local features with high accuracy.

In view of this, we propose a multi-scale umbrella feature. In our opinion, using multi-scale spatial perspectives to construct features can alleviate the degradation of representational capabilities caused by the chaotic distribution of adjacent points. The use of multi-scale umbrella features can further improve the representation ability of point cloud spatial features, thereby improving the accuracy of point cloud segmentation tasks.

Given a point $X_i = (x_i, y_i, z_i)$ and normal vector $V_i = (a_i, b_i, c_i)$. Previous work defined the surface position as $p_i = a_i x_i + b_i y_i + c_i z_i$. Triangular RepSurf is defined as **(1)**.

$$t_i = (a_i, b_i, c_i, p_i) \tag{1}$$

To feed point cloud into models, they replace the original point X_i with re-compute centroid X_i' of the triangle. Then the Umbrella

RepSurf is defined as **(2)**. Note that $\mathcal{A}$ is an aggregation function and $\mathcal{T}$ is a transformation function. x'_{ij} is the normalized coordinate compute by x_{ij}. K is the number of neighbors.

$$u_i = \mathcal{A}(\{\mathcal{T}([x'_{ij}, t_{ij}]), \forall j \in \{1, \dots, K\}\}) \quad (2)$$

Based on previous work **(1)** and **(2)**, we proposed a calculation formula for umbrella features in multi-scale spaces as **(3)**.

$$MU_i = \mathcal{L}(\mathcal{O}(u_{ij}, \forall j \in \{1, \dots, N\})) \quad (3)$$

Denote that N is the number of scale spaces needed to construct features. u_{ij} is the umbrella feature that construct in scale space j. $\mathcal{O}$ is a combination optimization function used to make trade-offs between features constructed in multiple scale spaces. $\mathcal{L}$ is a link function that connects the features. Specially, function $\mathcal{O}$ will firstly use maximum and minimum normalization to unify the dimensions of the eigenvalues calculated in different scale spaces. For each set of normalized eigenvalues, calculate its meaning and variance. After that, function will sort the scale space from small to large variance and select the scale space with smaller differences between the first three features as the final feature output. As for $\mathcal{L}$ link function. It will use matrix splicing to concatenate those features.

For the detailed process of multi-scale umbrella features, please see **Algorithm 1**.

Algorithm 1 Natural Language Style Pseudocode of **Multi-scale Umbrella RepSurf**

K: number of neighbors N: number of scale space

points: point coordinates set normal: point normal vectors set

neighbors = k-d_tree_search(points, k = K)

sort_neighbors = sort_by_distance(neighbors)

$S_1, S_2, \dots, S_N$ = cut_by_ratio(sort_neighbors) # usually 1:2: … :N

$S_1, S_2, \dots, S_N$ = sort_by_clock($S_1, S_2, \dots, S_N$)

for $S_1, S_2, \dots, S_N$: {$pairs_i$ = make_pairs(S_i, one_ahead(S_i))}

for $pairs_1, pairs_2, \dots, pairs_N$: {$normals_i$ = count_normal($pairs_i$)}

for $pairs_1, pairs_2, \dots, pairs_N$: {$centriods_i$ = count_centriods ($pairs_i$)}

for $normals_1, normals_2, \dots, normals_N$: {$position_i$ = sum ($normals_i$ * $centriods_i$)}

for $normals_1, normals_2, \dots, normals_N$: {$features_i$ = concatenate($centriods_i$, $normals_i$, $position_i$)}

for $features_1, features_2, \dots, features_N$:{$features_i$ = MLPs($features_i$)}

for $features_1, features_2, \dots, features_N$:{$features_i$ = Pooling($features_i$)}

for $features_1, features_2, \dots, features_N$:{$features_i$_copy = normalization($features_i$)}

for $features_1$_copy, $features_2$_copy, …, $features_i$_copy:{$variance_i$ = count_variance($features_i$_copy)}

sort_space = sort_by_variance($variance_1, variance_2, \dots, variance_i$)

$feature\_space_1, feature\_space_2, feature\_space_3$ = clip_sort_space(sort_space)

Multi-scale feature = link($features_{feature\_space1}$, $features_{feature\_space2}$, $features_{feature\_space3}$)

return multi-scale feature

3.3 Encoder-Decoder and Spatial self-attention mechanism

Encoder-decoder architecture is fundamental to hierarchical feature representation learning (Vaswani et al., 2017). It can convert the input data into fixed-length vectors, encoding contextual information and improve the representation ability of features. As for our multi-scale umbrella features, the feature dimensions of each point are not exactly the same, which may cause errors when training the model. The encoder-decoder module can help model to normalize the dimension of features, establish initial global information for features.

The proposal of the spatial self-attention mechanism essentially allows the model to pay special attention to certain areas when processing images or matrix. Its purpose is to improve the feature expression of key areas and also allows the model to capture the space relation between various parts of the input data. The local spatial features of point clouds have strong spatial correlation, that is, the features of adjacent points have strong similarities. Therefore, introducing a spatial self-attention mechanism into the model allows the convolutional neural network to pay more attention to the features of the current neighboring points during the feature extraction and feature classification process, which is beneficial to improving the accuracy of the model for feature segmentation. However, many models (Qi et al., 2017a; Qi et al., 2017b; Li et al., 2018) often ignore this correlation when learning features. There are also many models (Zhao et al., 2021; Wang et al., 2019) that have noticed this relationship and achieved superior results. Therefore, we introduce spatial self-attention mechanism to our model, which can help the model learn spatial relations.

Specifically, the spatial self-attention mechanism module uses linear connection layers to convert the input data into three parts: Query Q, Key K and Value V. Then, the model dots product of query and key to calculate attention score, which determines the importance of each value. After that, the model will apply softmax function to the attention score, so that the weights sum of each value is 1. Finally, the values are weighed and summed by the attention score to obtain the output of the attention mechanism.

$$Attention(Q, K, V) = softmax(\frac{QK^T}{\sqrt{d_k}})V \quad (4)$$

Note that, $\sqrt{d_k}$ is the scaling factor, usually the square root of the key vector dimension.

The spatial self-attention mechanism can be described by **(4)**. The detailed calculation process of the spatial self-attention mechanism can be found in **Algorithm 2**.

Algorithm 2 Natural Language Style Pseudocode of **Spatial Self-Attention Mechanism**

```
# multi-scale feature: feature output in Algorithm 1
# points: point coordinates set
# normal: point normal vectors set
Query = transform_to_query(multi-scale feature)
Key = transform_to_key(multi-scale feature)
Value = transform_to_value(multi-scale feature)
attention_score =  dot(Query, Key)
attention_score = softmax(zoom(attention_score))
out = sum(dot(attention_score, Value))
return out
```

4. Experiments and Analysis

4.1 Experimental data and evaluation methods

We employ Mean Intersection over Union (mIoU), Mean Accuracy (mAcc), and Overall Accuracy (OA) as the primary evaluation metrics, adhering to standard benchmarks. We employ the open-source dataset, S3DIS (Armeni et al., 2016), as the primary source at indoor scene for training and testing our point cloud segmentation task model. We also conducted tests on ScanNet, a larger-scale open-source indoor point cloud dataset.

We evaluate five models on S3DIS 6-fold and S3DIS Area-5: the original model (both result in the paper and under our settings), a multi-scale umbrella feature model (MUSF), a multi-scale umbrella feature model with encoder-decoder (MUSF-ED), a multi-scale umbrella features model with encoder-decoder and spatial self-attention mechanism (MUSF-SSA). These models are trained and tested on the aforementioned datasets, with their results displayed in **Table 1**. For specific performance of each category is shown in **Table 2**. Furthermore, in the tests on the ScanNet dataset, we evaluated the performance of MUSF-SSA under the environment of a single NVIDIA RTX 4090 GPU, and the test results are shown in **Table 3**.

When we conduct model performance evaluation analysis, we hope to be able to control the uniformity of the experimental environment. Therefore, we retrain the Repsurf-U (Ran et al., 2022) in a single A100 environment. Unfortunately, we do not achieve the same performance as the model in the original paper, and even the performance of the model dropped significantly. We attribute this performance discrepancy to limitations in our available computational resources (specifically, GPU memory constraints compared to the original study). Still, we achieved better results in this case. The further experiments are under our experimental environment.

4.2 Experimental results and analysis

We retrain the Repsurf-U (Ran et al., 2022) model in our own environment and use the results of this model to compare with our three models. Unfortunately, due to the experimental environment, we are unable to completely reproduce the superior performance of the model. However, in order to ensure the rigor of the experimental comparison, all subsequent comparisons are based on the same experimental environment.

Judging from the results presented in **Table 1**, **Table 2**, and **Table 3**, the model performance is basically consistent with our expectations, with only minor fluctuations. Experimental results show that multi-scale umbrella features, encode-decoder structure and spatial self-attention mechanism all have a positive impact on the performance of the improved model in task results. To verify the effectiveness of the Multi-scale Umbrella Feature (MUSF), we also applied MUSF feature processing to the PointNet++ and Point Transformer models and conducted point cloud segmentation experiments on the ScanNet dataset. The results show that compared with inputting raw point cloud data, the performance of the models with MUSF features as input has been improved.

For the S3DIS 6-fold test results, our MUSF-SSA has outstanding performance, with an increase of only 11% in the number of parameters.

We represent five scenes in **Figure 4**, which greatly show our improvement in most cases. For example, our MUSF-SSA can better identify simple geometric objects in the scene, such as doors or special walls. Such results show that in addition to the characteristics of the point cloud itself, the model can also learn the spatial relationship between point clouds and the geometric features corresponding to the segmentation categories, thereby having better performance capabilities for objects with obvious geometric shapes. In complex scenes like offices or storage areas, MUSF-SSA demonstrates robustness against point cloud sampling artifacts and noise.

Specially, in the test results of S3DIS-Area 5, we find that multi-scale umbrella features can improve the segmentation performance of the model in most categories. At the same time, the encoder-decoder structure can also improve the performance of the model to a certain extent, especially for the overall accuracy of the model. The spatial self-attention mechanism module has the greatest ability to improve the model and has a strong positive impact on the model in almost all categories.

For specific segmentation categories, the performance of MUSF-SSA at simple objects such as ceiling, floor and wall have huge improvement compared to our setting. Which shows the effectiveness of our model improvements both in feature characterization and deeper feature learning. As for the complex objects, for instance, bookcase and sofa, our MUSF-SSA achieves significant improvements compared to the baseline configuration of Repsurf-U, even exceeds the optimal model in most categories. The qualitative comparisons of these scene refinement results across different models are further illustrated in Figure 5.

4.3 Discussion

Limitation. Although effective, multi-scale umbrella features can cause computational redundancy when segmenting simple geometries like floors and ceilings. Furthermore, MUSF-SSA performance may decline in regions with missing or highly messy points, as noise can lead to poor feature construction.

Additionally, computing attention scores in the spatial self-attention module requires significant computational resources. To mitigate this, block attention calculations could be employed to reduce resource consumption. However, this approach introduces implementation challenges, such as the strict requirement for consistent block sizes. It may also lead to loss of information at boundaries, compromising sequence integrity and the model's ability to capture long-range feature relationships without further complex processing logic.

Method	S3DIS 6-fold			S3DIS Area-5			# Param
	mIoU	mAcc	OA	mIoU	mAcc	OA	
PointNet (C. R. Qi et al, 2017)	47.6	66.2	78.5	41.1	48.9	-	1.7M
PointNet++ (C. R. Qi et al, 2017)	-	-	-	56.0	61.2	86.4	0.969M
PointCNN (Y. Li et al, 2018)	65.4	75.6	88.1	57.3	63.9	85.9	0.6M
PointTrans (H. Zhao et al, 2021)	73.5	81.9	90.2	**70.4**	76.5	**90.8**	4.9M
PAConv (M. Xu et al, 2021)	69.3	78.6	-	66.5	73.0	-	-
KPConv (H. Thomas et al, 2019)	70.6	79.1	-	67.1	72.8	-	14.9M
Repsurf-U (H. Ran et al, 2022)	74.3	82.6	90.8	68.9	76.0	90.2	0.976M
Repsurf-U (our setting)	71.2	78.9	88.6	64.2	72.7	87.8	0.976M
MUSF	72.4	80.5	88.9	66.1	75.1	88.7	0.976M
MUSF-ED	74.3	81.7	89.8	66.2	73.8	89.5	1.038 M
MUSF-SSA	**74.8**↑0.5	**82.9**↑0.3	**91.2**↑0.4	68.6	**76.5**↑0.5	90.4	1.088 M

Table 1: The test situation of the model on the S3DIS data set (using Area 5 as the test area). We use the average intersection-to-union ratio (mIoU, %), average accuracy (mAcc, %), overall accuracy (OA, %), and number of parameters (Params). Bold indicates that the current indicator performs best on the current method. The green arrow indicates how much the model proposed in this article is improved compared to the original model. We train and test the models with one NVIDIA A100 Tensor Core GPU 40G. The batch size is set to 10.

Method	ceiling	floor	wall	beam	column	window	door	chair	table	bookcase	sofa	board	clutter
RepSurf-U	93.3	98.5	85.7	0.0	38.1	61.5	71.8	80.1	**90.4**	**80.3**	71.2	**68.2**	**56.0**
RepSurf-U (our setting)	90.8	97.4	83.3	0.0	30.7	56.8	71.6	74.5	86.4	58.6	69.2	62.1	53.2
MUSF	91.2	97.8	84.5	0.0	**40.2**	60.2	73.1	78.8	87.6	60.2	70.5	65.7	51.5
MUSF-ED	93.7	98.3	84.7	0.0	32.6	59.3	70.8	78.4	88.2	64.9	71.1	62.0	56.6
MUSF-SSA	**94.3**	**98.6**	**86.2**	0.0	39.7	**62.1**	**74.4**	**81.2**	89.0	71.3	**71.8**	68.1	55.3

Table 2: The performance of the model in each category on Area 5 of the S3DIS dataset. We use the average intersection-to-union ratio (mIoU, %) to determine the performance of different models. Bold indicates that the current indicator performs best on the current method.

Method	ScanNet			Param
	mIoU	mAcc	OA	
MUSF-SSA	60.30	91.11	65.33	1.088M
PointNet++	33.90	-	-	0.969M
PointCNN	49.80	-	-	0.6M
KPConv	67.12	94.37	72.43	14.900M
PTv1+MUSF	67.69	97.35	67.16	7.768M
PTv1	66.77	97.17	67.05	7.767M
PointNet2+MUSF	56.14	87.81	63.87	0.972M
PointNet2	33.95	61.73	43.69	0.969M

Table 3: Test Results of Models on the ScanNet Dataset (Validation Set Accuracy) Note: All models were trained and tested on a single NVIDIA RTX 4090 GPU, with a batch size set to 4; this table adopts the same evaluation metrics and units as Table 2. In the lower part of the table, "PTv1+MUSF" denotes the Point Transformer (PTv1) model with MUSF features as input.

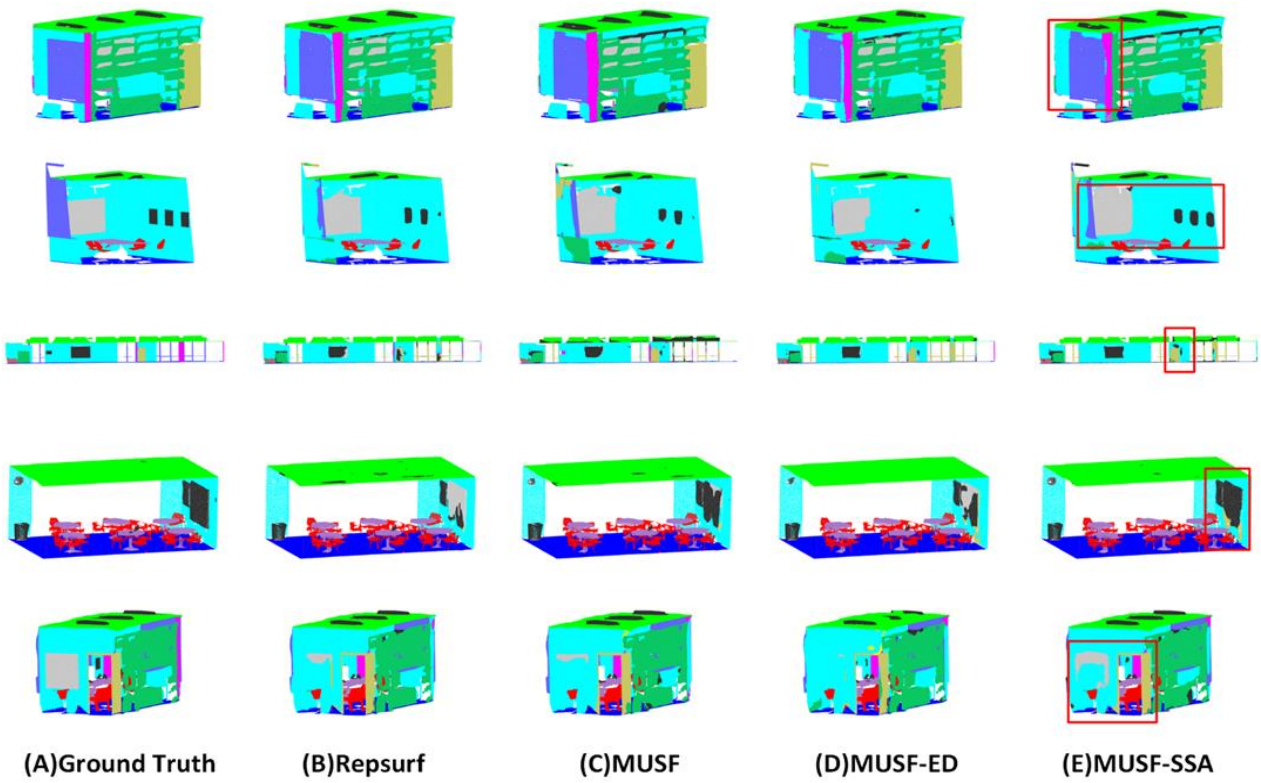

Figure 4. Model scene visualization results.

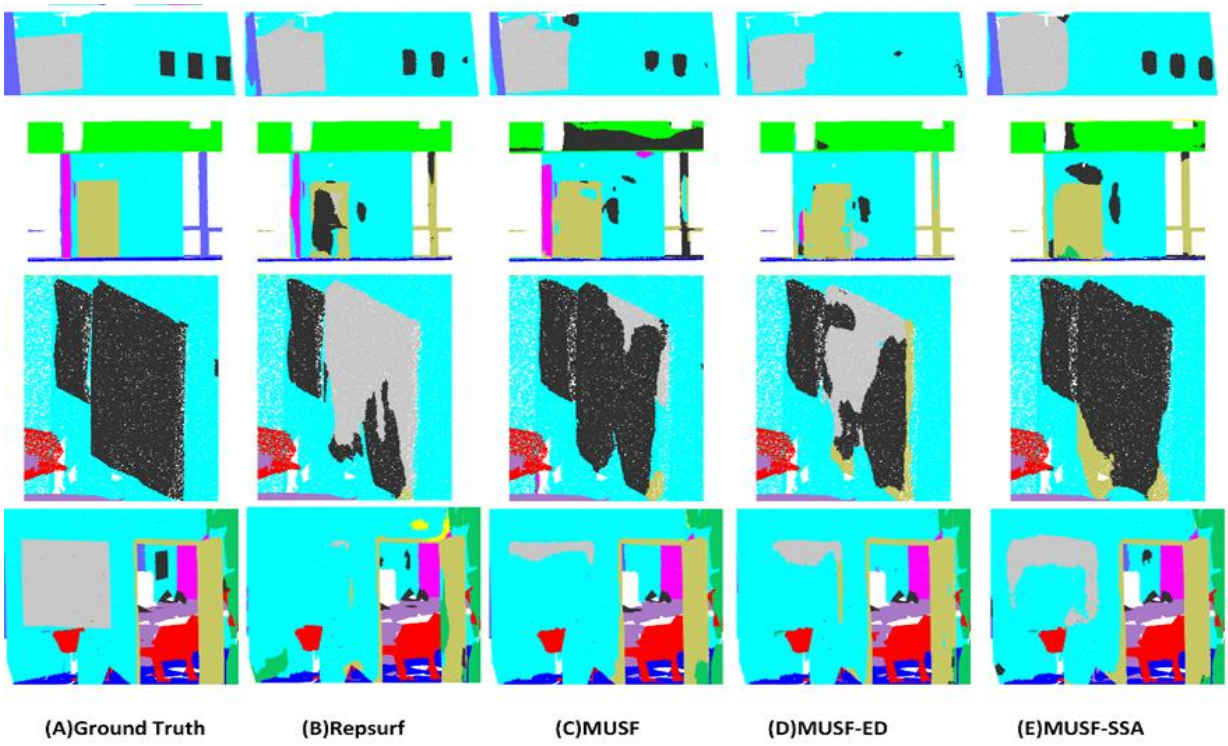

Figure 5. Model scene refinement results.

5. Conclusions

Our goal is to refine point cloud feature extraction and improve model efficiency with a lightweight feature extraction backbone (Zhang et al., 2020). We implement four core optimizations: k-d tree accelerated point cloud neighbor search, multi-scale umbrella feature construction, encoder-decoder structure, and spatial self-attention mechanism.

1) k-Dimension tree neighbor point search

For a 50,000-point point cloud during model training, the original proximity-based KNN search took 1.4593 s, while the k-d tree-based neighbor search reduces the time to 0.1987 s. This confirms that k-d tree outperforms KNN for point cloud neighbor search, accelerating initial feature construction for both model training and test set inference.

2) Multi-scale umbrella feature

We extend single-scale umbrella curvature and features to multi-scale umbrella feature construction, fully integrated into the model training pipeline. Experimental results show that even with our lightweight model, multi-scale umbrella features achieve better segmentation performance than the original baseline. This validates that multi-scale umbrella features better characterize complex point cloud surface geometries, thus improving model segmentation performance.

3) Encoder-decoder structure

The encoder-decoder structure is often used in sequence-to-sequence models. In our model, we simplify this structure and successfully perform feature encoding and feature extraction of encoded content. After this, we successfully achieve feature decoding and weighting with the original features, which helps the model to learn the contextual relation between features.

4) Spatial self-attention mechanism

Leveraging the inherent spatial correlation of point cloud features, we integrate a spatial self-attention module into the model for training and testing. Experimental results confirm that this mechanism enables the model to capture spatial correlations in point cloud features, which is critical for improving model segmentation performance.

In general, our proposed multi-scale umbrella features combined with the spatial self-attention mechanism outperform most mainstream existing models on point cloud semantic segmentation. This work validates that multi-scale umbrella features effectively improve point cloud feature representation, while the spatial self-attention mechanism captures intrinsic spatial feature dependencies, greatly boosting model segmentation performance.

Acknowledgements

This research was supported by the National Natural Science Foundations of China under Grant numbers 42571499, the Shenzhen Science and Technology Program under Grant number KJZD20230923115508017, the Guangdong Basic and Applied Basic Research Foundation under Grant number 2024A1515010777, The Guangdong Science and Technology Strategic Innovation Fund (the Guangdong-Hong Kong-Macau Joint LaboratoryProgram, Project No. 2020B1212030009).

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. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 1534-1543.

Brown, R. A., 2014. Building a Balanced k-d Tree in O(kn log n) Time. arXiv preprint arXiv:1410.5420.

Ceccarelli, G., Gao, W., Peters, R., 2025. Semantic segmentation of point clouds with the 3D medial axis transform. ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences, X-4/W6-2025, 33–40.

Diao, L., Song, S., Qian, Y., Ren, D., 2025. ZigzagPointMamba: Spatial-Semantic Mamba for Point Cloud Understanding. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR).

Eisl, N., Halperin, D., 2025. Point Cloud Based Scene Segmentation: A Survey. arXiv preprint arXiv:2503.12595.

Foorginejad, A., Khalili, K., 2014. Umbrella Curvature: A New Curvature Estimation Method for Point Clouds. Procedia Technology, 12, pp. 347-352.

Fujiwara, K., Hashimoto, T., 2020. Neural Implicit Embedding for Point Cloud Analysis. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR).

Gawlikowski, J., et al., 2022. A Survey of Uncertainty in Deep Neural Networks. arXiv preprint arXiv:2107.03342.

Guo, Y., Wang, H., Hu, Q., Liu, H., Liu, L., Bennamoun, M., 2020. Deep Learning for 3D Point Clouds: A Survey. IEEE Transactions on Pattern Analysis and Machine Intelligence, 43(12), pp. 4338-4364.

Jaritz, M., Gu, J., Su, H., 2019. Multi-view PointNet for 3D Scene Understanding. In: Proceedings of the IEEE International Conference on Computer Vision (ICCV).

Landrieu, L., Simonovsky, M., 2018. Large-scale Point Cloud Semantic Segmentation with Superpoint Graphs. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR).

Li, H., et al., 2023. MVPNet: A multi-scale voxel-point adaptive fusion network for point cloud semantic segmentation in urban scenes. International Journal of Applied Earth Observation and Geoinformation, 122, p. 103391.

Li, Y., Bu, R., Sun, M., Wu, W., Di, X., Chen, B., 2018. PointCNN: Convolution On X-Transformed Points. In: Advances in Neural Information Processing Systems (NeurIPS).

Luo, H., Chen, Z., Ye, F., Huang, T., He, H., Hu, W., 2025. Cross-sensor adaptive semantic segmentation for mobile laser scanning point clouds based on continuous potential scene surface reconstruction. ISPRS Journal of Photogrammetry and Remote Sensing, 228, 537–551.
Nezhadarya, E., Taghavi, E., Razani, R., Liu, B., Luo, J., 2020. Adaptive hierarchical down-sampling for point cloud classification. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR).

Qi, C. R., Su, H., Mo, K., Guibas, L. J., 2017a. PointNet: Deep Learning on Point Sets for 3D Classification and Segmentation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR).

Qi, C. R., Yi, L., Su, H., Guibas, L. J., 2017b. PointNet++: Deep Hierarchical Feature Learning on Point Sets in a Metric Space. In: Advances in Neural Information Processing Systems (NeurIPS).

Ran, H., Liu, J., Wang, C., 2022. Surface Representation for Point Clouds. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR).

Shi, W., Rajkumar, R., 2020. Point-GNN: Graph Neural Network for 3D Object Detection in a Point Cloud. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR).

Thomas, H., Qi, C. R., Deschaud, J., Marcotegui, B., Goulette, F. O., Guibas, L. J., 2019. KPConv: Flexible and Deformable Convolution for Point Clouds. In: Proceedings of the IEEE International Conference on Computer Vision (ICCV).

Vaswani, A., et al., 2017. Attention Is All You Need. In: Advances in Neural Information Processing Systems (NeurIPS).

Wang, Y., Hu, Y., Kong, Q., Zeng, H., Zhang, L., Fan, B., 2023. 3D point cloud semantic segmentation: state of the art and challenges. Chinese Journal of Engineering, 45(10), pp. 1653-1665.

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), pp. 1-12.

Wu, W., Qi, Z., Fuxin, L., 2019. PointConv: Deep Convolutional Networks on 3D Point Clouds. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR).

Xu, M., Ding, R., Zhao, H., Qi, X., 2021. PAConv: Position Adaptive Convolution with Dynamic Kernel Assembling on Point Clouds. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR).

Xu, R., Yang, S., Wang, X., Wang, T., Chen, Y., Pang, J., Lin, D., 2025. PointLLM-V2: Empowering Large Language Models to Better Understand Point Clouds. IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI).

Ye, D., et al., 2023. LidarMultiNet: Towards a Unified Multi-Task Network for LiDAR Perception. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR).

Ye, X., Li, H., Tan, H., et al., 2022. DRINet++: Efficient Voxel-as-point Point Cloud Segmentation. IEEE Transactions on Geoscience and Remote Sensing, 60, pp. 1-15.

Yue, Y., Robert, D., Wang, J., Hong, S., Wegner, J. D., Rupprecht, C., Schindler, K., 2025. LitePT: Lighter Yet Stronger Point Transformer. arXiv preprint arXiv:2512.13689.

Zhang, F., Fang, J., Wah, B., Torr, P., 2020. Deep FusionNet for Point Cloud Semantic Segmentation. In: Proceedings of the European Conference on Computer Vision (ECCV), pp. 644-663.

Zhao, H., Jiang, L., Jia, J., Torr, P., Koltun, V., 2021. Point Transformer. In: Proceedings of the IEEE International Conference on Computer Vision (ICCV).