LLM-Enhanced Semantic Segmentation of Large-Scale Urban LiDAR Point Clouds via Contextual Prompting

Jinlong Wang[1], Tao Shen[1], Liang Huo[1], Fulu Kong[1], Jiahui Wang[1], Xuejia Wei[1]

[1] School of Geomatics and Urban Spatial Informatics, Beijing University of Civil Engineering and Architecture, No. 15, Yongyuan Road, Huangcun Town, Daxing District, Beijing, 102627, China

Keywords: LiDAR, Point Cloud, Semantic Segmentation, Large Language Models, Contextual Prompting.

Abstract

As a key carrier of 3D spatial information, the semantic segmentation of urban LiDAR point clouds directly impacts the reliability of applications such as autonomous driving and digital twins. However, existing methods face two core bottlenecks: firstly, insufficient adaptation to scene-specific semantics, and secondly, an inference gap between LiDAR structured semantics and segmentation instructions, which makes it difficult to effectively combine the reasoning ability of large language models with LiDAR geometric semantics. To address these issues, this paper proposes a contextual cue framework of "LiDAR semantics-large language model-retrieval-enhanced generation". The framework first designs a lightweight semantic mapping module to convert the structured information inherent to LiDAR into natural language cues that can be understood by LLMs. Secondly, it constructs a LiDAR semantics-text vector library, utilizing the RAG mechanism to retrieve fine-grained knowledge of similar scenes in real-time, generating precise segmentation cues that include geometric features and contextual relationships. Finally, through a three-stage progressive training strategy, it guides LLMs to gradually learn the mapping relationship from semantic understanding to segmentation instruction generation.Ablation experiments verify the effectiveness of each module, and the inference efficiency meets the real-time processing requirements of large-scale urban data. This study provides a new technical path for high-precision, fine-grained semantic segmentation of urban LiDAR point clouds and also offers a theoretical reference for promoting the in-depth application of large language models in 3D spatial intelligence.

1. Introduction

Urban LiDAR point clouds, with their high-precision and high-density three-dimensional geometric measurement capabilities, have become a key data support in fields such as autonomous driving environmental perception, urban digital twin modeling, and infrastructure operation and maintenance. Compared to traditional 2D images, LiDAR point clouds not only provide precise spatial coordinates (X, Y, Z) but also contain rich geometric structure information (such as normal vectors, point cloud density, local curvature) and semantic labels preset in the dataset (such as "road", "curb", "sidewalk"). However, in large-scale urban applications, semantic segmentation of LiDAR point clouds[2] still faces significant challenges, mainly reflected in the following two aspects.Firstly, there is an inadequate adaptation to scene-specific semantics. In urban LiDAR scenes, there exist a large number of special semantic categories that are not covered by training data. For instance, in mainstream datasets such as nuScenes, "curb" is a unified semantic label. However, in actual urban scenes, there are significant differences between "permanent curbs" (usually made of concrete, with a height of 10-15cm, and uniform point cloud density) and "temporary curbs" (usually made of plastic or rubber, with a height of 5-8cm, lower point cloud density, and often appearing in construction areas). Another example is that "bus-specific curbs" often have special geometric shapes (such as higher point cloud reflectance intensity corresponding to yellow painted areas) and spatial locations (adjacent to bus lanes). The coarse-grained semantic labels that come with existing LiDAR datasets[1] cannot cover these fine-grained details. If relying solely on large language models (LLMs) for prompt tuning, due to the lack of understanding of the geometric features of specific scenes, LLMs often generate vague and non-guiding segmentation instructions (for example, only prompting to "segment curbs" without specifying the differences in point cloud density, z-axis range, and spatial correlation between temporary and permanent curbs).The second bottleneck is the disconnect between LiDAR semantics and segmentation instructions. LiDAR data itself carries rich structured semantic information, which exists in precise numerical form, such as "the 3D bounding box coordinates of the curb are (x_min, y_min, z_min, x_max, y_max, z_max)", "the point cloud density of this area is 20 points/㎡", "the surface normal vector is (0.01, 0.02, 0.999)", etc. However, this information exists in the form of structured data and cannot be directly converted into natural language prompts that LLM can understand. Existing research, such as LiDAR-LLM, attempts to map point clouds to the embedding space of LLM through an encoder, but this approach requires complex cross-modal alignment modules and is prone to losing the precise geometric semantics of LiDAR. As a result, the powerful reasoning ability of LLM is difficult to combine with the precise geometric semantics of LiDAR, limiting the further improvement of segmentation accuracy. To address these two major bottlenecks, this paper proposes an innovative "LiDAR semantics-large language model-retrieval-enhanced generation" contextual prompt framework. The core idea of this framework is: instead of directly inputting LiDAR point clouds into LLM, its inherent precise semantic information is used as "prior knowledge", and through retrieval-enhanced methods, highly scene-adaptive fine-grained segmentation prompts are generated to guide downstream segmentation networks to achieve more accurate recognition.

The main contributions of this paper are as follows:

1.Propose a LiDAR semantic mapping module: Design a lightweight regularized template to convert the structured semantics (3D bounding box, point cloud density, normal vector, category label) inherent to LiDAR into natural language text clues that are comprehensible to LLMs. Establish a direct association between "data-semantics-instructions" to avoid complex cross-modal alignment.

2.Constructing a RAG-enhanced context cue generation mechanism[3]: For the first time, we introduce retrieval-augmented generation technology into the field of LiDAR point

cloud semantic segmentation, construct a LiDAR semantic-text vector library, and supplement the inherent semantic limitations of LiDAR by retrieving fine-grained knowledge of similar scenes in real-time, generating precise and interpretable segmentation cues.
3.Design a three-stage progressive training strategy: Addressing the shortcomings of LLMs in LiDAR semantic understanding, we devise a progressive training strategy consisting of "semantic-text association training → semantic perception enhancement training → segmentation instruction fine-tuning". This strategy gradually guides LLMs to learn geometric semantic understanding, spatial relationship reasoning, and segmentation instruction generation, ensuring the effectiveness and reliability of prompts.
4.Comprehensive experiments conducted on public datasets: The effectiveness of the method was verified on the nuScenes extended dataset. Through quantitative analysis, qualitative demonstration, and ablation experiments, it was demonstrated that the proposed method exhibits significant advantages in segmentation accuracy, cue quality, and computational efficiency.
The organizational structure of this paper is as follows: Section 2 reviews relevant research work; Section 3 elaborates on the proposed methodological framework; Section 4 introduces experimental setup, result analysis, and ablation experiments; and Section 5 summarizes the full paper and outlines future work.

2. Related Work

2.1 Point cloud semantic segmentation based on deep learning

After years of development, point cloud semantic segmentation methods based on deep learning have evolved into multiple technical paths.
Projection-based methods project 3D point clouds onto 2D planes (such as spherical projection, bird's-eye view projection) and utilize mature 2D convolutional networks for segmentation. These methods are computationally efficient, but the projection process introduces geometric information loss, resulting in lower segmentation accuracy for small objects (such as curbs, poles).
Voxel-based methods divide irregular point clouds into regular voxel grids and process them using 3D convolutional neural networks. This type of method can preserve 3D spatial structure, but the computational complexity increases exponentially with voxel resolution, making it difficult to handle large-scale urban scenes.
Point-based methods directly process the original point cloud coordinates and extract features point by point through multi-layer perceptrons. However, these methods rely on large-scale, high-quality labeled data and are difficult to generalize to fine-grained categories that do not appear in the training set. When faced with categories such as "temporary curbs" that are scarce in the training data, point-based methods often perform poorly.

2.2 Application of large language models[4] in 3D vision

With the emergence of large language models such as ChatGPT and LLaMA, researchers have begun to explore their potential applications in three-dimensional visual tasks.
Multimodal alignment methods attempt to encode point clouds into the embedding space of LLMs. Point-Bind aligns the embedding spaces of point clouds, images, and text through contrastive learning, achieving zero-shot 3D classification. Point-LLM further projects point cloud features into the input space of LLMs, leveraging the reasoning capabilities of LLMs for 3D understanding. However, such methods require complex cross-modal alignment modules (such as Q-Former, projection layers), and the alignment process may lose precise geometric information of point clouds.
Instruction tuning methods, such as LL3DA, construct large-scale 3D instruction datasets to fine-tune LLMs to enable them to understand 3D scenes and execute visual instructions. Such methods demonstrate potential in open vocabulary 3D understanding, but their inputs are typically downsampled point clouds or voxel features, making it difficult to preserve fine-grained geometric details.
LiDAR-LLM represents a recent direct attempt to apply LLM to the understanding of LiDAR point clouds. This approach encodes LiDAR point clouds and inputs them into LLM, leveraging its reasoning capability to generate segmentation instructions. However, its input remains the original point cloud features, failing to fully utilize the precise structured semantic information inherent to LiDAR, such as 3D bounding boxes and point cloud density. Furthermore, when dealing with special scenarios like construction areas, the prompts generated by LLM still appear rough and lack an understanding of the specificity of the scene.

2.3 Retrieval-augmented generation[5]

Retrieval Augmented Generation (RAG) enhances the accuracy and factuality of the content generated by Large Language Models (LLM) by retrieving relevant information from external knowledge bases. RAG has achieved remarkable results in open-domain question answering and knowledge-intensive tasks by combining a retriever and a generator.
This paper introduces the RAG mechanism into the field of urban LiDAR point cloud semantic segmentation for the first time. By constructing a LiDAR semantic-text vector library, it achieves real-time retrieval and integration of scene-specific knowledge. Compared with existing work, the innovations of this study lie in: (1) the retrieval object is the fine-grained geometric semantic features of LiDAR point clouds, rather than image textures or text descriptions; (2) the retrieval results are used to generate precise segmentation cues, rather than directly answering questions; (3) the retrieval and generation processes are closely integrated, with LLM integrating basic semantics and retrieval knowledge to form segmentation instructions with high scene adaptability.

3. Methodology

The "LiDAR Semantics-LLM-RAG" contextual cue framework proposed in this study is illustrated in Figure 1. This framework comprises three core modules: the LiDAR Semantics Mapping Module, the RAG-Enhanced Contextual Cue Generation Module, and a three-stage progressive training strategy. Each of these modules will be elaborated on below.

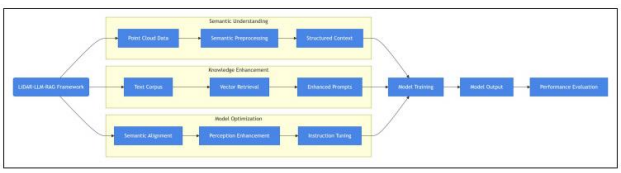

Figure 1. Technical Methodology

3.1 LiDAR Semantic Mapping Module

This module aims to address the "gap" between LiDAR structured semantics[6] and LLM natural language input. Instead of directly inputting point cloud coordinates into the LLM, we utilize the precise semantic information inherent in LiDAR data as a bridge. The core assumption of this design is that the

structured semantics inherent in LiDAR (such as 3D bounding box and point cloud density) already provide a highly abstracted and precise description of the scene. The LLM does not need to learn the original coordinates of the point cloud from scratch; it only needs to understand the relationship between these semantics and the segmentation task.

3.1.1 LiDAR semantic preprocessing[7]

The input is a large-scale urban LiDAR point cloud L \in \mathbb{R}^{n \times 3}L ∈ R n × 3, where nn represents the number of point clouds. Based on the annotation information of datasets (such as nuScenes), we extract structured semantic attributes of target categories (such as "curb", "road", "pedestrian walkway"). The specific extracted attributes include:
3D bounding box: For each semantic object, extract the coordinate range of its minimum circumscribed cube (x_{\min}, y_{\min}, z_{\min}, x_{\max}, y_{\max}, z_{\max})(x min ,y min ,z min ,x max ,y max ,z max), as well as the center point coordinates (x_c, y_c, z_c)(x c ,y c ,z c) and dimensions (w, h, l)(w,h,l).
Geometric features: Calculate the point cloud density of the object as \rho = N / A, where NN represents the number of points contained in the object and AA denotes the projected area of the object on the horizontal plane. Calculate the average normal vector as \vec{n} = (\bar{n}_x, \bar{n}_y, \bar{n}_z) x , n ¯ y , n ¯ z) is used to describe the surface orientation.
Coarse classification labels: Semantic labels preset in the dataset, such as "curb", "road", "vehicle", etc.

3.1.2 Semantic-to-text mapping

The structured semantics mentioned above are transformed into natural language fragments through a preset regularized template. The design of the template follows the following principles: (1) Accuracy: strictly preserving the numerical accuracy of LiDAR semantics; (2) Readability: converting into descriptions that conform to natural language habits; (3) Completeness: covering all key attributes that may affect segmentation.
Table 1 presents examples of mapping templates designed for different semantic attributes. For instance, for a curb object, the generated text description is: "The 3D bounding box of the curb has a z-axis range of [-0.1m, 0.05m], a point cloud density of 20.5 points/㎡, a normal vector approximately perpendicular to the ground, and belongs to the category of traffic facilities."

LiDAR semantic attributes	Map text template
3D Bounding Box (z-range)	"The z-axis range of the object is [min_z, max_z] meters."
3D Bounding Box (xy-range)	"The coverage area of the object on the horizontal plane is x ∈ [min_x, max_x], y ∈ [min_y, max_y]."
point cloud density	"The point cloud density in this area is approximately {density} points per square meter."
normal vector	"The normal vector of this surface is approximately perpendicular/parallel to the ground."
coarse-grained label	"This object belongs to the category of {category}."

Table 1. Example of LiDAR semantic mapping template

The mapping process strictly avoids introducing 2D visual noise (such as color, texture, and other information unrelated to LiDAR), ensuring that the LLM performs reasoning solely based on geometric semantics.

3.1.3 Semantic feature input

The mapped text segments are directly input as the initial context for the LLM. Since the text segments are clearly associated with the geometric and semantic attributes of LiDAR, the LLM can directly learn the correspondence between "semantic-segmentation instructions" based on this, without requiring complex cross-modal alignment modules (such as VAT, Q-Former). This significantly reduces computational complexity while avoiding errors that may be introduced during feature alignment.

3.2 Contextual cue generation module

This module utilizes the RAG mechanism to address the issue of insufficient scene adaptation for LiDAR-inherent semantics, generating fine-grained and highly instructive segmentation cues.

3.2.1 Build a LiDAR semantic-text vector database

The vector library serves as the knowledge source for the RAG mechanism. We construct the vector library based on the nuScenes extended dataset, and the specific steps include:
1.Collecting fine-grained LiDAR-text pairs: Based on the nuScenes dataset, we have provided additional annotations for special scenes such as construction zones and bus lanes. For fine-grained objects within each special scene (such as temporary curbs and bus lane curbs), we record their LiDAR semantic attributes (generated by the semantic mapping module) along with their corresponding natural language descriptions. For instance, a temporary curb might have LiDAR semantic attributes like "z-axis range [-0.08m, -0.02m], point cloud density 12 points/㎡", with the corresponding text description being "This object is a temporary curb, typically found in construction zones, and has a lower point cloud density than permanent curbs.".
2.Text embedding: Using the word embedding layer of LLaMA-7B, convert each text description into a dense vector \(\mathbf{v}\) \(\in\) \(\mathbb{R}^{4096}\).
3.Index construction: The IVF_FLAT index is established using the FAISS library, dividing the vectors into n_{\text{list}} = 100 clusters to support efficient similarity retrieval. This index structure strikes a good balance between retrieval speed and accuracy.
Ultimately, the vector library comprises approximately 5,000 LiDAR-text pairs, encompassing fine-grained semantic categories in common urban scenes..

3.2.2 Real-time prompt generation

When new LiDAR data is inputted, the real-time prompt generation process is as follows:
1.Generate basic textual cues: Through the semantic mapping module, convert the currently input LiDAR point cloud into basic textual cues T_{\text{base}}. For example: "The current scene contains a curb object: bounding box with a z-axis range of [-0.1m, 0.05m], and a point cloud density of 20 points/㎡."
2.Encoding user instructions: The user's segmentation instructions (such as "segment curbs") are also converted into a

query vector \(\mathbf{q}\) through the word embedding layer of LLaMA-7B.
3.Similarity Search: In the FAISS index, calculate the cosine similarity between \(\mathbf{q}\) and all stored vectors, and select the Top-5 most similar text segments as auxiliary clues \(T_{\text{ret}}\). For example, the search results may include: "The point cloud density of temporary curbs is lower than that of permanent curbs, and there are construction signs in adjacent areas", "The z-axis range of bus-only road curbs is usually [-0.12m, -0.03m], with a higher point cloud density".
4.LLM integration: The basic cue T_{\text{base}}T base and the auxiliary cue T_{\text{ret}}T ret are concatenated to form the input context for the LLM. Based on this information, the LLM generates a comprehensive and precise segmentation prompt PP. For example: "Segment the temporary curbs of adjacent construction vehicles in the current scene with a z-axis range of [-0.05m, 0.02m] and a point cloud density of 12-15 points/㎡."
This design supplements scene-specific semantics through RAG, enabling the generated prompts to adapt to the geometric features of different scenes and avoiding the coarse-grained limitations of relying solely on the built-in semantics of LiDAR.

3.3 Three-stage progressive training strategy

To ensure that LLM can effectively utilize LiDAR semantics and generate high-quality prompts, we have designed a three-stage progressive training strategy. This strategy draws on the progressive logic of LiDAR-LLM, but has been adjusted to suit the semantic utilization characteristics of our framework.

Stage 1: Semantic-Text Association Training
The goal of this stage is to enable the LLM to understand the correspondence between LiDAR geometric parameters and semantic categories. The training data consists of LiDAR semantic-text pairs, with each pair containing a set of LiDAR structured attributes and their corresponding textual descriptions. For example, if the input is "z-axis range [-0.1m, 0.05m], density 20 points/ ㎡ , normal vector perpendicular to the ground", the output would be "curb".
We employ the cross-entropy loss function to supervise the LLM to generate text descriptions that match the input semantics. During the training process, only the adapter layer of the LLM is fine-tuned, while the main parameters are frozen. This stage involves training for 2 epochs with a learning rate of 1e-4.

Stage 2: Semantic Perception Enhancement Training
The training data is in the form of (structured semantic description, spatial relationship description). The LLM needs to generate the correct spatial relationship description based on the input semantic description. In this way, the LLM learns the spatial association patterns of different semantic categories, enabling it to consider contextual constraints when generating prompts in the future. This stage is trained for 2 epochs with a learning rate of 5e-5.

Stage 3: Segmentation instruction fine-tuning
The goal of this stage is to enable the LLM to generate imperative prompts that meet the requirements of downstream segmentation models. We utilize a dedicated instruction dataset for urban LiDAR segmentation, encompassing tasks such as "segmenting bus-only road edges", "distinguishing between temporary and permanent road edges", and "extracting road edges at the intersection of sidewalks and roads".
The training data is in the form of (scene semantic description + user task, segmentation prompt). The LLM needs to generate a precise and executable segmentation prompt based on the input scene information and user task. This stage is trained for 2 epochs with a learning rate of 1e-5.
Training configuration: The Adam optimizer (β1=0.9, β2=0.999) is used, with an initial learning rate of 1e-4, which is halved every 2 epochs, and a total of 6 epochs are trained. The main parameters of the LLM (LLaMA-7B) are frozen throughout the process, and only the adapter and semantic mapping layers are fine-tuned to ensure training efficiency.

4. Experimental Evaluation

4.1 Experimental setup

4.1.1 Dataset
This study is based on the nuScenes dataset for expansion. nuScenes is a large-scale autonomous driving dataset containing 1000 scenes, each with approximately 20 seconds of LiDAR point cloud data. We selected data that includes rich urban scenes (commercial areas, construction zones, transportation hubs) and utilized its built-in 19 semantic annotations to construct LiDAR semantic-text pairs.
To evaluate fine-grained segmentation performance, we additionally labeled two categories, "Temporary Curb" and "Bus Lane Curb", in certain scenes (mainly construction areas and bus lane areas). The labeling was independently completed by three professional annotators, and the final labels were determined through a voting mechanism.

4.1.2 Model Configuration
LiDAR semantic mapping module: Utilizes a regularized template, with the specific template detailed in Table 1.
LLM: LLaMA-7B, with the main parameters frozen, only the Adapter layer is fine-tuned (with approximately 2 million parameters).
RAG vector library: FAISS IVF_FLAT index, retrieve Top-5 texts, vector dimension 4096.
Downstream segmentation network: To evaluate prompt quality, we use PointNet++ as a fixed segmentation network, with only the prompt generated by the LLM as an additional input to guide its segmentation. The input to PointNet++ includes point cloud coordinates and the embedding vector of the LLM prompt.
Training environment: NVIDIA Tesla A100 (40GB) GPU × 2, CUDA 11.8, PyTorch 2.0.

4.1.3 Evaluation Metrics
Segmentation accuracy: Overall mean Intersection over Union (mIoU), category mIoU for difficult-to-classify categories (curbs, temporary curbs, bus lane curbs, sidewalks), and accuracy.
Prompt quality: The matching accuracy (Precision) between prompts and the true semantics of LiDAR, the recall rate (Recall) covering fine-grained features, and the F1 score. We calculate these through a combination of manual evaluation and automated metrics (BLEU, ROUGE).
Efficiency: single-scenario inference time (seconds), total training duration (hours).

4.1.4 Comparative Method
We have selected the following methods as the baseline:
PointNet++: A classic point cloud segmentation network, serving as a baseline for segmentation accuracy.
LiDAR-LLM (Zero-shot): After encoding the LiDAR point cloud, it is inputted into the LLM for zero-shot generation of segmentation prompts.
LiDAR-LLM (Fine-tuned): LiDAR-LLM after fine-tuning on nuScenes.

2D MLLM-based RAG: An RAG method based on 2D multimodal LLMs (such as Flamingo), which projects LiDAR point clouds into 2D images for retrieval.

4.2 Segmentation accuracy analysis

Table 2 shows the segmentation accuracy comparison of various methods on the test set. The experimental results indicate that:
Overall performance: Compared to the baseline PointNet++ (with an overall mIoU of 68.5%), the proposed method in this paper elevates the overall mIoU to 84.2%, representing a significant improvement of 15.7 percentage points. This enhancement is notably superior to that of LiDAR-LLM (78.4% after fine-tuning) and 2D MLLM-based RAG (75.9%). This underscores the effectiveness of the proposed method in harnessing the structured semantics of LiDAR and augmenting knowledge through RAG.
Significant improvement in difficult-to-classify categories: For temporary curbs, the accuracy of our method has increased from 48.7% in PointNet++ to 71.7%, an improvement of 23 percentage points; the mIoU for curb categories has risen from 52.1% to 75.3%. For bus-only curbs, our method achieves an accuracy of 68.4%, compared to only 41.2% for PointNet++. These improvements stem from: (1) the semantic mapping layer directly linking LiDAR semantics with text, avoiding alignment errors; (2) RAG supplements scene-specific knowledge for fine-grained categories such as temporary curbs and bus-only curbs, enabling LLMs to generate discriminative prompts.
Comparison with LiDAR-LLM: Compared to LiDAR-LLM (zero-shot), the method proposed in this paper improves the accuracy of road edges by 15.5 percentage points; compared to LiDAR-LLM (fine-tuned), it improves by 9.2 percentage points. This proves that directly utilizing the inherent semantics of LiDAR (rather than encoding the raw point cloud and inputting it into LLM) can more accurately preserve geometric information, thereby improving segmentation accuracy.

method	Overall mIoU (%)	Roadside mIoU (%)	Temporary curb accuracy (%)	Sidewalk mIoU (%)
PointNet++	68.5	52.1	48.7	70.2
LiDAR-LLM (Zero-shot)	71.2	55.3	52.4	73.1
LiDAR-LLM (Fine-tuned)	78.4	59.8	62.1	76.5
LiDAR-LLM (Fine-tuned)	75.9	57.2	58.9	74.8
Proposed Method	84.2	75.3	71.7	85.6

Table 2 Comparison of main segmentation accuracies

4.3 Prompt quality analysis

Table 3 presents the evaluation results of prompt quality. We adopted a combination of manual evaluation and automated metrics. For manual evaluation, three annotators scored the generated prompts for 100 random scenarios on a scale of 1-5, and the average score was taken as the final metric.

The enhancement effect of RAG is significant: After introducing RAG, the F1 score of the prompts increased from 87.2% to 92.3% for a pure LLM (with semantic mapping only), the precision rose from 88.8% to 93.5%, and the recall increased from 85.7% to 91.1%. This indicates that the contextual knowledge retrieved by RAG makes the generated prompts more aligned with the actual semantics of LiDAR, and can cover fine-grained features such as "temporary curb-construction vehicle association" and "bus lane curb-lane line distance".
Specific case: For the LiDAR data of the construction area, the pure LLM prompt is only "segment curb", while the method proposed in this paper generates "segment z-axis range [-0.05m, 0.02m], point cloud density 12-15 points/㎡, temporary curb of adjacent construction vehicles". The latter includes specific geometric parameters and contextual relationships, significantly enhancing the guidance for downstream segmentation networks.

method	Manual scoring (1-5)	Accuracy (%)	Recall (%)	F1 value (%)
Pure LLM (Semantic Mapping Only)	3.8	88.8	85.7	87.2
LiDAR-LLM	3.5	86.2	82.4	84.3
This paper's method (+RAG)	4.6	93.5	91.1	92.3

Table 3: Quality Assessment of Prompts

4.4 Efficiency Analysis

Table 4 presents a comparison of the training and inference efficiency of various methods. The efficiency advantage primarily stems from two aspects:
Training Efficiency: Due to the removal of complex cross-modal alignment modules (such as VAT, Q-Former), the total training duration of our method (42 hours) is reduced by approximately 22% compared to LiDAR-LLM (54 hours). This is because the semantic mapping module is a lightweight regularized template that requires no additional training, and the construction of the RAG vector library is a one-time offline operation.
Inference Efficiency: The inference time for a single scene is approximately 0.7 seconds, which is 41.7% faster than the 2D MLLM-based RAG (1.2 seconds) and 22.2% faster than LiDAR-LLM (0.9 seconds). The RAG retrieval optimized by FAISS introduces almost no additional delay (the retrieval takes approximately 0.05 seconds). This inference speed can meet the real-time processing requirements of large-scale LiDAR data in cities (typically requiring <1 second per frame).

method	Training duration (hours)	Single-scene inference time (seconds)	Parameter count (M)
PointNet++	12	0.15	1.5
LiDAR-LLM	54	0.9	7200+

2D MLLM-based RAG	48	1.2	8000+
Proposed Method	42	0.7	7200+

Table 4 Comparison of Training and Inference Efficiency

4.5 Ablation Experiment

To verify the effectiveness of each module, we designed four sets of ablation experiments, and the results are presented in Table 5.

E1: No semantic mapping. Directly inputting the original LiDAR point cloud to the LLM (via an encoder). The overall mIoU decreased to 72.5%, and the accuracy of temporary road edges decreased to 59.3%. This proves that the semantic mapping layer is crucial for the LLM to understand point clouds, and directly inputting the original point cloud will prevent the LLM from effectively extracting geometric semantics.

E2: Without RAG. Only semantic mapping + LLM are used, without retrieving the vector library. The F1 score drops to 86.8%, and the overall mIoU decreases to 79.1%. This verifies the important role of the RAG mechanism in supplementing scene-specific semantics, especially in fine-grained categories such as temporary curbs.

E3: No stage 2 training. Only stage 1 and stage 3 training are conducted, skipping the semantic perception enhancement training. The mIoU of the curb category decreases to 73.2%, indicating that the spatial relationship learning in stage 2 is crucial for improving segmentation and localization accuracy. Without this stage, LLMs find it difficult to understand spatial constraints such as "curbs are closely adjacent to roads".

E4: All modules. Complete method for achieving optimal performance.

method variant	Overall mIoU (%)	Roadside mIoU (%)	Temporary curb accuracy (%)	F1-score (%)
E1: No semantic mapping	72.5	58.1	59.3	84.5
E2: No RAG	79.1	68.5	64.2	86.8
E3: No stage 2 training	80.3	73.2	66.9	89.1
E4: Our method	84.2	75.3	71.7	92.3

Table 5 Comparison Results of Four Ablation Experiments

4.6 Discuss

The necessity of semantic mapping: Ablation experiment E1 demonstrates that directly inputting the original point cloud into LLM yields poor results. This is because the training data of LLM primarily consists of natural language, lacking semantic understanding of 3D point cloud coordinates[8]. Through semantic mapping, we convert the point cloud into a natural language format familiar to LLM, establishing an effective communication bridge.

RAG and Generalization Ability: The RAG mechanism not only enhances the segmentation accuracy of fine-grained categories but also strengthens the model's generalization ability to unknown scenarios. For instance, in novel construction areas that appear in the test set but not in the training set, RAG can retrieve temporally similar knowledge of temporary road edges, thereby generating effective segmentation cues. This demonstrates that RAG is an effective "lifelong learning" mechanism.

Limitations: The method proposed in this paper relies on the built-in LiDAR semantic annotations (such as 3D bounding boxes) in the dataset. For raw LiDAR data without annotations, these structured semantics need to be generated first through object detection methods, adding a preprocessing step. Future work could explore how to integrate object detection and segmentation cue generation in an end-to-end manner.

5. Conclusions and Prospects

This paper proposes an innovative "LiDAR Semantics-LLM-RAG" contextual cue framework to address two core bottlenecks in fine-grained semantic segmentation of urban LiDAR point clouds: insufficient scene-specific semantic adaptation and a disconnect in the association between LiDAR semantics and segmentation instructions. The core idea of this framework is to "use semantics as a bridge and retrieval as a wing". It leverages the structured information inherent in LiDAR through a lightweight semantic mapping module, supplements scene-specific knowledge through the RAG mechanism, and ultimately enables the LLM to generate accurate and interpretable fine-grained segmentation cues.

Experiments conducted on the nuScenes extended dataset demonstrate that the method proposed in this paper has achieved the following key results:

1.Significant improvement in segmentation accuracy: The overall mIoU reached 84.2%, an increase of 15.7 percentage points compared to PointNet++; the accuracy of temporary road edges increased from 48.7% to 71.7%, and the accuracy of bus lane edges reached 68.4%.

2.Prompt quality optimization: After RAG enhancement, the F1 score of the prompt reaches 92.3%. The generated prompt contains specific geometric parameters and contextual relationships, providing stronger guidance for downstream segmentation networks.

3.The efficiency advantage is evident: training time is reduced by 22%, and single-scene inference time is 0.7 seconds, meeting the real-time processing requirements of large-scale LiDAR data in cities.

4.Module effectiveness verification: Ablation experiments demonstrate the significant contributions of semantic mapping, RAG mechanism, and three-stage training strategy.

This study not only provides a new technical approach for high-precision semantic segmentation of urban LiDAR point clouds, but also offers valuable theoretical references and practical paradigms for promoting the in-depth application of large language models in three-dimensional spatial intelligence.

Future work outlook:

1.End-to-end integration: Exploring the integration of object detection (generating 3D bounding boxes) and segmentation cue generation in an end-to-end manner, enabling the method to directly process raw LiDAR point clouds.

2.Multimodal extension: Extend the RAG vector library to a multimodal knowledge base encompassing images, text, and LiDAR, further enriching the sources of retrieval knowledge.

3.Active learning: Leveraging the reasoning capabilities of LLMs, identifying uncertain areas in segmentation results, and proactively requesting manual annotation to achieve efficient data augmentation.

4.Real-time deployment: Optimize the inference efficiency of LLM (such as using LoRA, quantization technology) to enable its deployment on in-vehicle computing platforms and support real-time autonomous driving applications.

References

[1] Xu Han, Chong Liu, Yuzhou Zhou, Kai Tan, Zhen Dong & Bisheng Yang. (2024). WHU-Urban3D: An urban scene LiDAR point cloud dataset for semantic instance segmentation. *ISPRS Journal of Photogrammetry and Remote Sensing, 209,* 500-513. https://doi.org/10.1016/J.ISPRSJPRS.2024.02.007.

[2] Xiaokai Sun, Baoyun Guo, Cailin Li, Na Sun, Yue Wang & Yukai Yao. (2024). Semantic Segmentation and Roof Reconstruction of Urban Buildings Based on LiDAR Point Clouds. *ISPRS International Journal of Geo-Information, 13* (1), 19-. https://doi.org/10.3390/IJGI13010019.

[3] Yan Zhou, Yichao Fan, Haibin Zhou & Richard Irampaye. (2025). MSCSeg: Multi-scale contextual network for LiDAR point cloud semantic segmentation. *Signal, Image and Video Processing, 19* (13), 1074-1074. https://doi.org/10.1007/S11760-025-04464-2.

[4] Arega Denekew, Tesfahun Berehane & Molalign Adam. (2025). Integrating Large Language Models and CNN-LSTM for Enhanced Portfolio Optimization. *Operations Research Forum, 6* (4), 157-157. https://doi.org/10.1007/S43069-025-00564-4.

[5] Shuai Guo, Ting Chen, Penghui Wang, Jun Ding, Junkun Yan & Hongwei Liu. (2026). Knowledge embedding fusion based on language model for enhanced radar target recognition. *Signal Processing, 238,* 110199-110199. https://doi.org/10.1016/J.SIGPRO.2025.110199.

[6] Jinpeng Li, Yuan Li, Yiping Chen, Hongchao Fan & Ruisheng Wang. (2026). City-scale building instance segmentation from LiDAR point clouds via structure-aware method. *International Journal of Applied Earth Observation and Geoinformation, 146,* 105086-105086. https://doi.org/10.1016/J.JAG.2026.105086.

[7] Kai Fischer, Martin Simon, Stefan Milz & Patrick Mäder. (2025). MagneticPillars++: Efficient LiDAR Odometry Via Deep Frame-To-Keyframe Point Cloud Registration. *Applied Artificial Intelligence, 39* (1), https://doi.org/10.1080/08839514.2025.2472105.

[8] Yifan Zhang, Junhui Hou, Siyu Ren, Jinjian Wu, Yixuan Yuan & Guangming Shi. (2025). Self-Supervised Learning of LiDAR 3D Point Clouds via 2D-3D Neural Calibration.. *IEEE transactions on pattern analysis and machine intelligence, 47* (10), 9201-9216. https://doi.org/10.1109/TPAMI.2025.3584625.