Review │ Received 9 December 2024; Accepted 27 February 2025; Published 12 March 2025
https://doi.org/10.55092/aias20250003

A survey on deep learning-based lidar place recognition

Weizhong Jiang [1], Shubin Si [1,2], Hanzhang Xue [3], Yiming Nie [1], Zhipeng Xiao [1], Qi Zhu [1,*] and Liang Xiao [1,*]

[1] Unmanned Systems Technology Research Center, Defense Innovation Institute, Beijing, China
[2] College of Intelligent Systems Science and Engineering, Harbin Engineering University, Harbin, China
[3] Test Center, National University of Defense Technology, Xi'an, China

* Correspondence authors; E-mails: zhuqiwk@126.com; xiaoliang@nudt.edu.cn.

Highlights:

- Proposes coarse-to-fine DL-LPR classification framework via data structure and model architecture.
- Reviews datasets, metrics, and performance comparisons of representative DL-LPR methods.
- Analyzes challenges in complex environments (long-term, large-scale, dynamic) and future trends.

Abstract: LiDAR-based place recognition (LPR) technology processes 3D LiDAR point clouds and encodes them into feature descriptors, enabling mobile robots to recognize previously visited locations. This capability supports critical tasks such as loop closure detection and re-localization. With the rapid advancements in deep learning, deep learning-based LiDAR place recognition (DL-LPR) has emerged as the dominant research direction in this field. However, existing reviews on DL-LPR remain limited in scope. To address this gap, this paper focuses on DL-LPR, introducing its core concepts, system structures, and applications. It presents a coarse-to-fine classification framework to systematically categorize and review existing methods, based on two dimensions: input data structure and model architecture. Furthermore, this paper summarizes commonly used datasets and performance evaluation metrics, along with performance comparisons of representative methods. Finally, it provides an in-depth analysis of the challenges faced by DL-LPR in complex environments, such as long-term, large-scale, and dynamic settings, and offers insights into future development trends.

Keywords: place recognition; LiDAR; deep learning; mobile robots; navigation; re-localization; loop closure detection

1. Introduction

The core principle of Place Recognition (PR) involves identifying a location within a global place features database or map, based on environmental data captured by sensors. As a subset of global localization technologies for mobile robots, PR enhances navigation and localization systems, playing a crucial role in tasks such as loop closure detection in Simultaneous Localization and Mapping (SLAM) and re-localization in long-term navigation [1]. These functions help mitigate cumulative localization errors and ensure robust localization [2].

PR technologies can be categorized by sensor type, mainly including Visual-based Place Recognition (VPR), Radar-based Place Recognition (RPR), and LiDAR-based Place Recognition (LPR). This paper focuses on outdoor applications of PR. While VPR leverages cameras to capture rich environmental features such as color and texture, its sensitivity to lighting variations limits its effectiveness in outdoor settings. In contrast, millimeter-wave radar is more resilient to adverse weather conditions like rain, fog, and dust. However, radar-based systems suffer from sparse data coverage and noise susceptibility, hindering the maturity of RPR for outdoor applications. LiDAR, with its longer range, high distance accuracy, and robustness to lighting fluctuations, has emerged as a more reliable solution for outdoor place recognition in mobile robots. Therefore, this paper focuses on recent advancements in LPR from a practical perspective. For detailed discussions on VPR and RPR, readers are referred to comprehensive reviews [1,3–9], which are beyond the scope of this work.

In 2009, Magnusson *et al.* [10] introduced a 3D LPR solution using Normal Distributions Transform (NDT) [11], a feature-based approach relying on handcrafted features. Following the resurgence of deep learning in 2012, marked by the success of the ImageNet [13] image classification challenge, significant progress was made in fields like image processing and natural language processing. However, deep learning applications in LPR lagged behind due to the fact that early neural networks, particularly those based on 2D convolutions, were designed for structured data (e.g., images and text), which are relatively regular. These networks were not directly applicable to sparse and unordered 3D LiDAR point clouds. As a result, most LPR research before 2017 focused on traditional methods, such as histogram-based features [10,14–15], keypoint-based features [16–17], and segment-based features [18], which relied on mathematical models for statistical analysis or data structure transformations to generate local or global place descriptors.

In 2017, Yin *et al.* [19] employed neural networks to extract features from 2D projections of 3D point clouds for loop closure detection, marking one of the first deep learning-based solutions for LPR. The same year, Qi *et al.* [20] introduced PointNet, a neural network model designed for processing 3D point clouds. Originally intended for 3D object detection and segmentation, PointNet provided an efficient framework for feature extraction from 3D point clouds and addressed the challenge of point cloud permutation invariance, advancing deep learning applications in 3D point cloud processing. In 2018, Mikaela *et al.* [21] introduced PointNetVLAD, the first DL-LPR model capable of processing 3D point clouds directly. By using PointNet for local feature extraction, PointNetVLAD demonstrated comparable performance to traditional methods. Since then, DL-LPR techniques have evolved, surpassing traditional methods in efficiency and accuracy across several benchmark datasets. While classic handcrafted LPR methods such as Scan Context [22–23], LiDAR-iris [24], Semantic Topological Descriptors [25], and Binary Image Fingerprints [26] were proposed after 2018, DL-LPR has become the dominant approach in LPR research, with numerous deep learning models emerging.

Despite these advances, there remains a notable lack of comprehensive review articles on DL-LPR. Existing reviews [1–2,27–28] fail to provide a complete overview of the technology. To fill this gap, this paper presents a systematic review of DL-LPR methods, offering an in-depth examination of the various technical branches and their current research status.

Table 1 The data structures and their characteristics of the inputs of the place encoding network.

Structure Type	Data Volume	Structural Characteristics	Contained Information	Other
3D Point Cloud	Large	Sparse, unordered	Fine-grained scene information	Sensitive to viewpoint variations
2D Projection	Small	Compact, ordered	Preserves geometric structure information, with some information loss	Facilitates rotation-invariant design
3D Voxel	Small	Sparse, ordered, efficient for retrieval	Information loss present	Sensitive to viewpoint variations
Semantic Information	Small	Sparse, suitable for graph construction	Contains high-level information, loses low-level information	Accuracy and efficiency depend on semantic extraction methods

A DL-LPR system consists of several key components: input data, neural network models, loss functions, evaluation metrics, and datasets. Among these, the design of the neural network model architecture is closely tied to the structure of the input data. For example, Yin *et al.* [19] use 2D convolutions because their input data is structured as 2D images, while PointNetVLAD [21] can process 3D point clouds directly due to its feature extraction module based on PointNet. In tasks such as 3D object detection [29] and classification [30], common preprocessing techniques, including 2D multi-view projection and voxelization, are used to reduce data size and regularize the data. Additionally, semantic segmentation techniques [31] can extract more detailed semantic information from 3D point clouds. These point clouds, being sparse, are better represented as graph structures, which are well-suited for processing by graph neural networks. These preprocessing methods have been adopted in the DL-LPR domain, expanding the range of input data types beyond raw 3D point clouds. The characteristics of different input data structures are summarized in Table 1.

Therefore, this paper classifies existing DL-LPR methods based on input data structures into four main categories: methods based on raw 3D point clouds, 2D projections, 3D voxelization, and semantic data. The corresponding data flow variations are depicted in Figure 1. However, classifying methods purely by input data structure oversimplifies the diversity of DL-LPR models, as methods within the same category can differ significantly in their network architectures. Understanding these variations is essential for designing more efficient models. To address this, the paper further refines the categorization by considering differences in network architectures, providing a comprehensive review of DL-LPR research from a coarse-to-fine perspective.

The main contributions of this paper are as follows:

(1) We propose a coarse-to-fine classification strategy, systematically summarizing and analyzing existing DL-LPR methods at two levels: input data structure and model network architecture.

(2) We outline the core components of DL-LPR systems, including key research challenges, commonly used datasets, and their characteristics. We also provide a detailed analysis and evaluation of selected DL-LPR methods, focusing on place recognition accuracy, generalization ability, and real-time performance.

(3) We discuss challenges in future DL-LPR research, particularly for long-term, large-scale, and high-dynamic applications, and explore development trends by incorporating advancements in emerging technologies and methodologies.

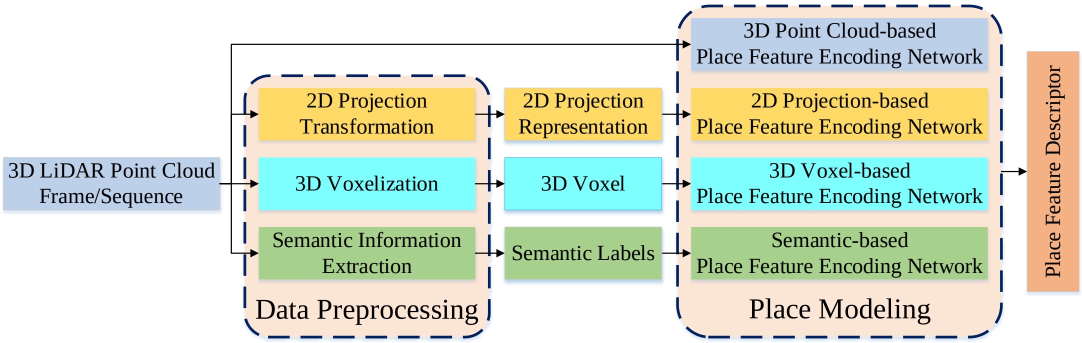

3D LiDAR Point Cloud Frame/Sequence
2D Projection Transformation
3D Voxelization
Semantic Information Extraction
Data Preprocessing
2D Projection Representation
3D Voxel
Semantic Labels
3D Point Cloud-based Place Feature Encoding Network
2D Projection-based Place Feature Encoding Network
3D Voxel-based Place Feature Encoding Network
Semantic-based Place Feature Encoding Network
Place Modeling
Place Feature Descriptor

Figure 1. Data flow and corresponding technical route in DL-LPR. In DL-LPR research, 3D LiDAR point clouds are typically preprocessed into various data structures, such as 2D projections, 3D voxels, and semantic segmentation labels. Corresponding neural network models need to be designed for feature encoding based on different types of input data, thereby giving rise to four distinct technical approaches.

The paper is structured as follows: section 1 provides the background of DL-LPR. Section 2 introduces the proposed classification approach, summarizes key research challenges, and systematically organizes and analyzes DL-LPR methods based on the coarse-to-fine classification strategy. Section 3 reviews commonly used datasets and performance evaluation metrics. Section 4 discusses challenges and development trends in long-term, large-scale, and high-dynamic environments. Finally, Section 5 concludes the paper.

2. Related background on DL-LPR research

This section summarizes the definitions of PR, LPR, and DL-LPR within the context of this research, following the typical process flow of a DL-LPR system (Figure 2). The components of the DL-LPR system are abstracted, and the relationships and distinctions between DL-LPR and related applications are discussed.

2.1. Concept of "Place"

The core focus of DL-LPR research is the concept of "place". Within the academic community, two primary perspectives on the definition of "place" exist.

The first definition is inspired by "place cells" in the hippocampus. O'Keefe and Dostrovsky [32] discovered that these cells are activated when an animal revisits a previously encountered location. These cells can update place information using external visual landmarks and self-motion estimation, even in changing environments. When the animal returns to the same position, the cells are reactivated, leading to their designation as "place cells". Drawing from this mechanism, Lowry *et al*. [3] proposed that "place" can be defined either as a precise point (e.g., a GPS coordinate) or as a continuous or discretized region, where the region boundaries are determined by criteria such as the robot's time step, travel distance, or scene appearance. A new place can be identified when the robot travels a certain distance or when the currently observed scene differs significantly from the previous one [9].

The second definition is based on the concept of "spatial view cells" in the animal brain. Research indicates that when an animal observes a specific region of its environment, "spatial view cells" are activated if the field of view overlaps with a prior observation, regardless of the spatial position. Garg *et al*. [7] applied this concept to VPR, defining place recognition as follows: if the field of view between two observations overlaps beyond a certain threshold — considering both metric distance and observation direction—the two locations are considered the same.

In DL-LPR research, these two definitions lead to two distinct ground-truth measurement standards: distance-based and overlap-based metrics.

(1) Distance metric: in works such as PointNetVLAD [21], distance is used as the measurement standard. During training, point cloud pairs within 10 meters (in Universal Transverse Mercator (UTM) coordinates) are considered positive, while pairs with coordinates more than 50 meters apart are negative. In the place recognition phase, if the UTM distance between the retrieved point cloud and the query frame is less than or equal to 25 meters, the recognition is considered successful. This method aligns with the first definition of “place”.

(2) Overlap metric: in works such as OverlapNet [33–34], the authors use a custom overlap rate to define relationships between samples. During training, point cloud pairs with an overlap rate greater than 0.3 are considered positive, while those below 0.3 are negative. In the recognition phase, if the overlap rate between the retrieved point cloud and the query frame exceeds 0.3, place recognition is deemed successful. This method aligns with the second definition of “place”.

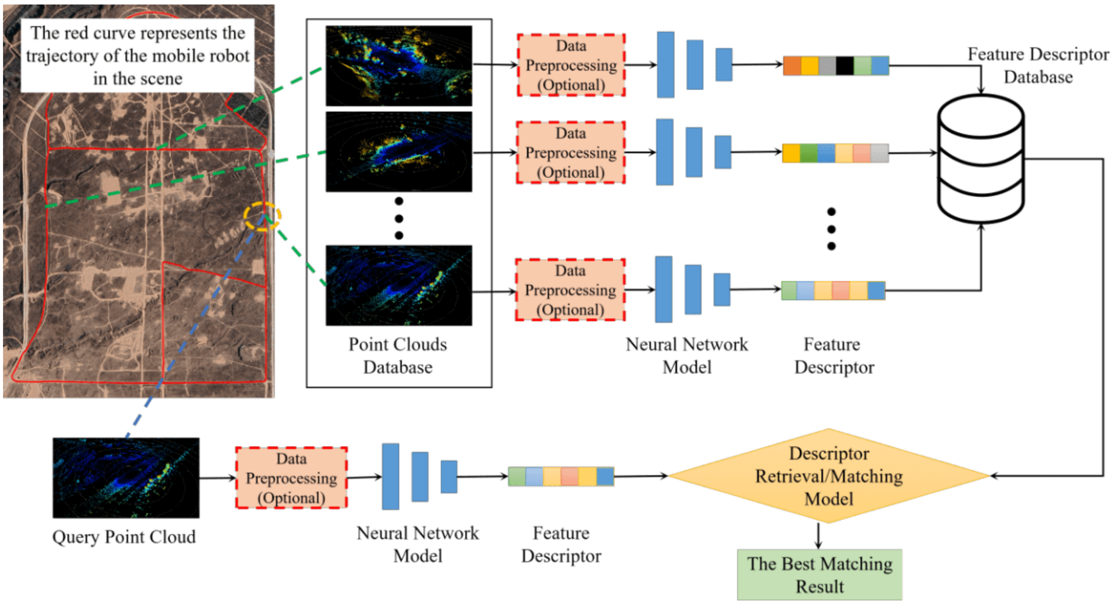

The red curve represents the trajectory of the mobile robot in the scene
Data Preprocessing (Optional)
Data Preprocessing (Optional)
Data Preprocessing (Optional)
Point Clouds Database
Neural Network Model
Feature Descriptor
Feature Descriptor Database
Query Point Cloud
Data Preprocessing (Optional)
Neural Network Model
Feature Descriptor
Descriptor Retrieval/Matching Model
The Best Matching Result

Figure 2. Example of typical DL-LPR system. A typical DL-LPR system comprises core modules including point cloud database construction, data preprocessing, feature encoding, and feature matching.

1.2. *Definition of DL-LPR*

Building on the definitions of PR and LPR, we define DL-LPR as follows: a mobile robot uses an onboard LiDAR sensor to observe its environment and processes the data with deep learning techniques. The robot learns global feature descriptors that effectively represent places in the scene, enabling it to identify previously visited locations by retrieving or matching these descriptors in the feature space.

1.3. System architecture of DL-LPR

Drawing from the VPR system architecture by Barros *et al*. [9] and incorporating Figure 2, the DL-LPR system architecture is derived, with the logical relationships between the system modules illustrated in Figure 3.

(1) Data preprocessing module: this module converts 3D point clouds into a structure suitable for subsequent network processing. Common preprocessing methods are illustrated in Figure 1.

(2) Place modeling module: this module maps sensor data or preprocessed data to the feature descriptor space. Its core is the place feature encoding network, typically comprising two submodules: local feature extraction and local feature aggregation.

(3) Place mapping module: this module organizes and stores global feature descriptors output by the place modeling module, constructing a scene feature map. The map can take various forms, such as a database, topological map, or topological-metric map. In DL-LPR, database-based feature maps are most commonly used.

(4) Confidence generation module: this module performs matching or nearest-neighbor searches on global feature descriptors. Based on similarity measures (e.g., distance or overlap rate), it identifies the descriptor in the feature map closest to or most overlapping with the query frame. The corresponding place is considered a candidate match.

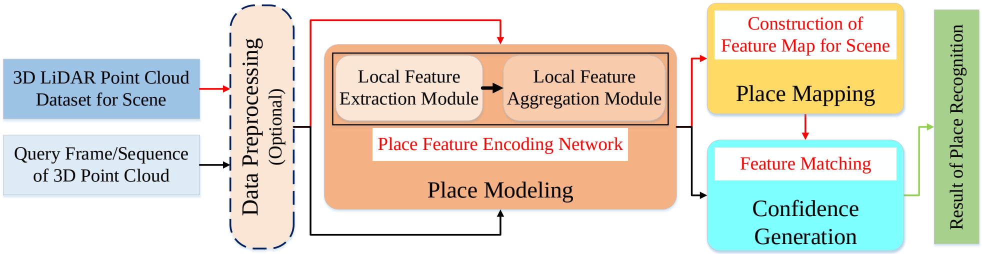

3D LiDAR Point Cloud Dataset for Scene
Query Frame/Sequence of 3D Point Cloud
Data Preprocessing (Optional)
Local Feature Extraction Module
Local Feature Aggregation Module
Place Feature Encoding Network
Place Modeling
Construction of Feature Map for Scene
Place Mapping
Feature Matching
Confidence Generation
Result of Place Recognition

Figure 3. System composition of DL-LPR. Drawing upon the canonical architecture of visual location recognition systems, a DL-LPR system can be partitioned into four core modules: data preprocessing (including dataset construction), location modeling (centered on feature encoding), scene mapping (constructing feature database and query library), and confidence generation (based on feature matching).

1.4. Relationship between DL-LPR and other tasks

At both the algorithmic and application levels, DL-LPR is closely related to tasks such as retrieval, regression, loop closure detection [151], and re-localization. The relationships between these tasks are illustrated in Figure 4.

At the algorithmic level, when the place feature map constructed by the DL-LPR system is represented as a database, the DL-LPR task essentially becomes a data retrieval task. Research in retrieval tasks can be applied to search for DL-LPR feature descriptors. Conversely, when the output of the DL-LPR model consists of continuous values, such as similarity scores, the task can be treated as a regression task.

At the application level, DL-LPR is used in position estimation for loop closure detection and re-localization in mobile robot autonomous navigation and localization. The key distinction is that place recognition typically does not estimate the robot's current pose, while loop closure detection and re-localization require both position and accurate pose information.

Additionally, for place recognition and re-localization tasks, the search/matching space is the pre-constructed scene feature map, which has a fixed spatial scale. In contrast, the search/matching space for loop closure detection tasks is within the historical frames, and its size increases as the task progresses [35].

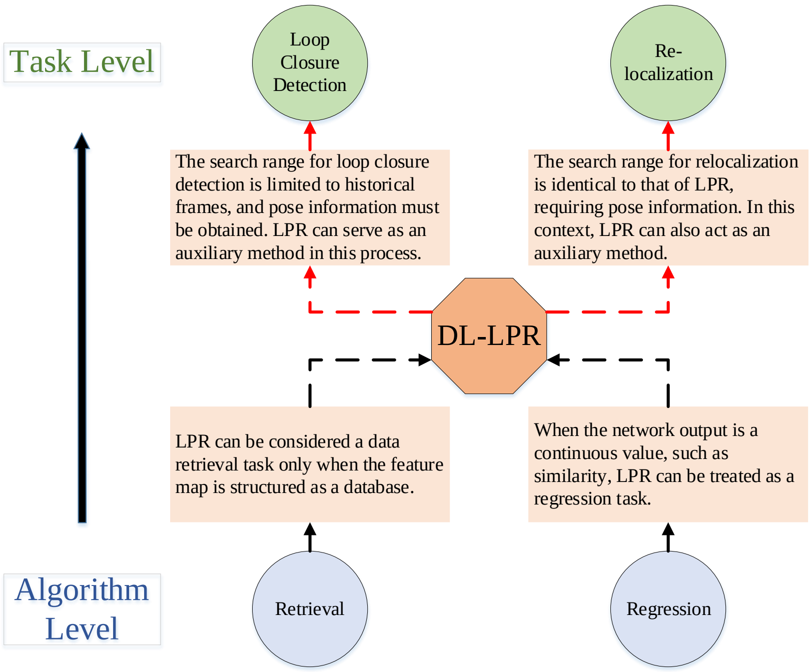

Task Level
Loop Closure Detection
Re-localization
The search range for loop closure detection is limited to historical frames, and pose information must be obtained. LPR can serve as an auxiliary method in this process.
The search range for relocalization is identical to that of LPR, requiring pose information. In this context, LPR can also act as an auxiliary method.
DL-LPR
LPR can be considered a data retrieval task only when the feature map is structured as a database.
When the network output is a continuous value, such as similarity, LPR can be treated as a regression task.
Algorithm Level
Retrieval
Regression

Figure 4. The relationship between DL-LPR and other tasks. At the algorithmic design level, DL-LPR involves the selection between retrieval and regression methodologies, a decision governed by the system's output format — whether employing discrete feature descriptors or continuous similarity metrics. At the application level, DL-LPR serves as the computational foundation for critical robotic navigation tasks including loop closure detection and re-localization.

2. Research status of DL-LPR

This section introduces the proposed classification method for the literature and outlines the key issues addressed in current studies before summarizing existing research according to this classification.

As shown in Figure 3, the core of the DL-LPR system lies in the place modeling module, which centers on the place feature encoding network. This network typically consists of two main components: the local feature extraction module and the local feature aggregation module. The input data is first processed by the local feature extraction module, and then aggregated into a global feature descriptor by the local feature aggregation module.

As previously mentioned, the design of the network structure is closely linked to the input data. Variations in DL-LPR at the network design level are primarily reflected in the local feature extraction module, while methods for local feature aggregation are generally more standardized, as demonstrated in Tables 2, 3, 4, and 5. Therefore, the coarse-to-fine classification method proposed in this paper primarily focuses on the local feature extraction module of the DL-LPR model.

2.1. Classification method

This paper classifies current DL-LPR research into two levels: the data structure level and the model network structure level, with a coarse-to-fine progression between them.

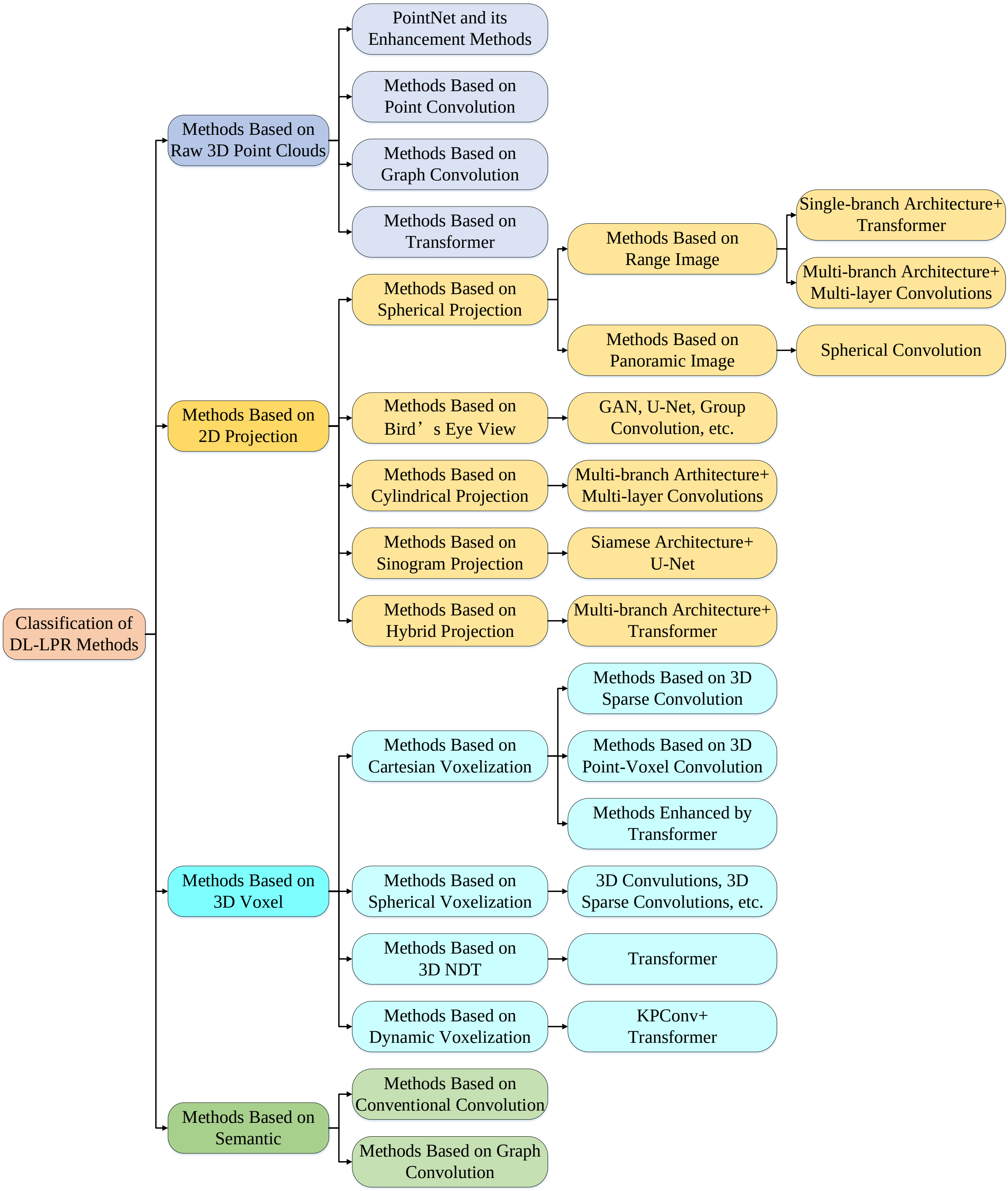

Classification of DL-LPR Methods
Methods Based on Raw 3D Point Clouds
PointNet and its Enhancement Methods
Methods Based on Point Convolution
Methods Based on Graph Convolution
Methods Based on Transformer
Methods Based on 2D Projection
Methods Based on Spherical Projection
Methods Based on Range Image
Single-branch Architecture+ Transformer
Multi-branch Architecture+ Multi-layer Convolutions
Methods Based on Panoramic Image
Spherical Convolution
Methods Based on Bird' s Eye View
GAN, U-Net, Group Convolution, etc.
Methods Based on Cylindrical Projection
Multi-branch Arthitecture+ Multi-layer Convolutions
Methods Based on Sinogram Projection
Siamese Architecture+ U-Net
Methods Based on Hybrid Projection
Multi-branch Architecture+ Transformer
Methods Based on 3D Voxel
Methods Based on Cartesian Voxelization
Methods Based on 3D Sparse Convolution
Methods Based on 3D Point-Voxel Convolution
Methods Enhanced by Transformer
Methods Based on Spherical Voxelization
3D Convulutions, 3D Sparse Convolutions, etc.
Methods Based on 3D NDT
Transformer
Methods Based on Dynamic Voxelization
KPConv+ Transformer
Methods Based on Semantic
Methods Based on Conventional Convolution
Methods Based on Graph Convolution

Figure 5. Classification of DL-LPR methods.

At the data structure level, DL-LPR methods are categorized based on the structure of the input data, including methods based on raw 3D point clouds, 2D projections, 3D voxels, and semantic data. Further subdivisions within these categories are made based on specific implementation details of the data transformation methods. At the network structure level, models within each major category are further refined according to similarities and differences in their network architectures. This two-level classification approach allows for a comprehensive and systematic summary of existing DL-LPR methods. A detailed classification scheme is presented in Figure 5.

Table 2. Representative methods based on original 3D point clouds.

Category	Method	Time	Local Feature Extraction Network	Local Feature Aggregation Network	Loss Function	Ground Truth	Source Code	Dataset
PointNet and its Enhancement Methods	PointNet-VLAD [21]	2018	PointNet [20]	NetVLAD [150]	Lazy-Triplet/ Quadruplet Loss	Distance	PointNet-VLAD	Oxford In-house
	PCAN [37]	2019	PointNet	Attention+ NetVLAD	Lazy-Quadruplet Loss	Distance	PCAN	
	LPD-Net [39]	2019	PointNet +GNN	NetVLAD	Lazy-Quadruplet Loss	Distance	LPD-Net	
	SeqLPD [40]	2019	PointNet +GNN	NetVLAD	Lazy-Quadruplet Loss	Distance	-	
	SOE-Net [45]	2019	PointOE (Modified PointNet)	Attention+ NetVLAD	Hard Plsitive-Hard Negative-Quadruplet Loss	Distance	SOE-Net	
	RPR-Net [48]	2022	ARIConv (Based on PointNet)	GeM	Triplet Loss	Distance	RPR-Net	
	Zhou [43]	2022	PointNet	-	Hard Contrastive Loss	Distance		KITTI
Methods Based on Point Convolution	DH3D [51]	2020	Multi-layer FlexConv [49]+ SENet [52]	Attention+ NetVLAD	N-Tuple Loss	Distance	DH3D	Oxford
	SE(3)-Equivariant [53]	2022	EPN [54]/E2PN [55] (Based on KPConv [50])	NetVLAD/GeM	Lazy-Quadruplet Loss	Distance	se3-equivariant	Oxford In-house KITTI
	KPPR [56]	2022	Point Cloud Compression Network+KPConv	NetVLAD	Contrastive Loss	Distance	KPPR	Oxford In-house
Methods Based on Graph Convolution	DAGC [38]	2020	ResGCN (Introduce EdgeConv into PointNet)	NetVLAD	Lazy-Quadruplet Loss	Distance	-	Oxford In-house
	EPC-Net [59]	2022	ProxyConv（Modified EdgeConv）	Grouped VLAD	Lazy-Quadruplet Loss	Distance	EPC-Net	
	HiBi-GCN [36]	2023	T-Net+Hierarchical Bidirectional Graph Convolution	NetVLAD	Lazy-Quadruplet Loss	Distance	-	
Methods Based on Transformer	PPT-Net [58]	2021	Graph Embedding+Pyramid Point-Transformer	Pyramid VLAD+ Context Gatting	Lazy-Quadruplet Loss	Distance	PPT-Net	Oxford In-house
	HiTPR [61]	2022	Hierarchical Transformer	Max Pooling	Lazy-Quadruplet Loss	Distance	-	
	FPET-Net [60]	2022	Point-Transformer	Attention	Binary Cross-Entropy Loss	Distance	-	KITTI

2.2. Key issues

Based on the literature review, current DL-LPR methods primarily address the following key challenges: viewpoint variation, appearance changes of the same place, perceptual ambiguity across different places, occlusion, and real-time performance. Solving these challenges is crucial for improving the overall performance of DL-LPR systems.

(1) Viewpoint variation. When mobile robots operate in real-world environments, they may observe the same place from different angles. Since the information captured by the observation data varies with the viewpoint, and the training data cannot cover all possible angles, the model may misidentify the same place from different viewpoints as distinct places. This issue is referred to as viewpoint variation.

(2) Appearance change of the same place. Factors such as seasonal changes, weather conditions, and road construction can cause the appearance of the same place to change. These changes may lead the model to incorrectly classify the same place as a different one, thereby increasing demands on the model's expressiveness and robustness.

(3) Perceptual confusion across different places. In environments like highways, corridors, and bridges, the geometric structure of places may appear highly similar, increasing the likelihood of the model misidentifying different places as the same. This issue highlights the importance of the model's ability to distinguish between structurally repetitive environments.

(4) Occlusion. Dynamic objects within the scene can cause occlusion by blocking certain areas from being scanned by LiDAR sensors. These objects may change position, leading to shifts in the distribution of observation data, which may not have been encountered during training. This imposes additional demands on the model's robustness.

(5) Real-time performance: Real-time performance is crucial for deploying models on real-world platforms. Complex network architectures often exhibit higher computational complexity, which may hinder the ability to meet real-time requirements. Thus, reducing model structural complexity without sacrificing accuracy has become a key area of research.

2.3. Methods based on raw 3D point clouds

A key characteristic of these methods is their use of raw 3D point clouds as input. However, due to the sparse, independent, and unordered nature of 3D point clouds, applying traditional 2D convolution techniques directly is challenging [36]. In the context of DL-LPR, this challenge has been addressed by leveraging PointNet [20] and its improved variants as backbone networks for local feature extraction. Moreover, various deep learning methods have been introduced to enhance feature extraction from 3D point clouds. Based on the core components of the local feature extraction network, methods based on raw 3D point clouds can be further categorized into sub-branches, including PointNet and its enhanced variants, point convolution-based methods, graph convolution-based methods, and Transformer-based methods. Representative models and their key attributes are summarized in Table 2.

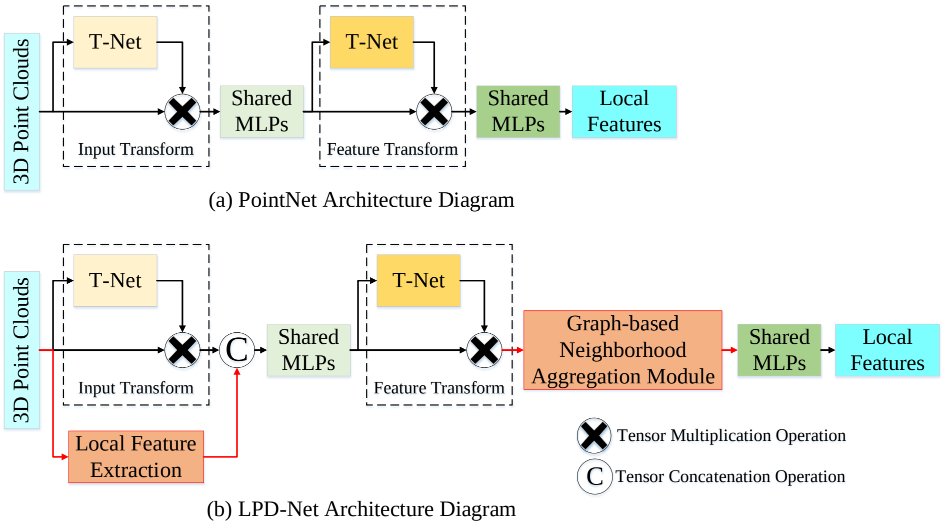


Figure 6. Examples of local feature extraction method based on PointNet.

2.3.1 PointNet and its enhancements

The core architecture of PointNet [20] consists of a multi-layer perceptron (MLP) and a max pooling layer, with its T-Net design ensuring invariance to the arrangement of 3D point clouds. Despite its strengths, PointNet has limitations, such as difficulties in cross-scale local feature extraction, lack of viewpoint invariance, and relatively high computational complexity.

PointNetVLAD [21] uses PointNet for local feature extraction from 3D point clouds but inherits many of PointNet's limitations. To address these, several models have introduced improvements while retaining the overall PointNet architecture. For example, PCAN [37] incorporates an attention mechanism after PointNet to weight local features, but it overlooks the relationships between points and their neighborhoods, resulting in higher computational complexity [38]. LPD-Net [39] and SeqLPD [40] embed adaptive local feature extraction modules within the PointNet architecture to enhance fine-grained feature extraction. Figure 6 compares their architectures with that of PointNet.

Studies show that local 3D feature descriptors typically offer better generalization than global features [41–42]. Building on this, Zhou *et al*. [43] used PointNet to learn 3D local deep descriptors for loop closure detection. SOE-Net [45], inspired by SIFT's direction encoding [44] in image processing, integrates a directional encoding unit from PointSIFT [46] into PointNet and employs a self-attention mechanism [47] to capture long-range dependencies. This improves the model's point-wise feature expression and robustness to viewpoint variations.

Some models, such as DAGC [38] and RPR-Net [48], deviate from the original PointNet architecture. DAGC retains only the T-Net structure and combines a dual-attention mechanism with residual graph convolution, enhancing the exploration of point relationships. In contrast, RPR-Net keeps the MLP from PointNet and introduces an ARIConv module with rotational invariance and attention mechanisms.

2.3.2 Point convolution-based methods

In DL-LPR, point-wise convolution methods, such as FlexConv [49] and KPConv [50], are widely used. DH3D [51] employs stacked FlexConv layers and SENet [52], integrating multi-level spatial context and channel feature correlations to form local feature descriptors.

To improve robustness to transformations like rotation and translation, Lin *et al*. [53] adopted EPN [54] or E2PN [55], building SE(3)-equivariant networks that learn global feature descriptors with SE(3)-invariant properties from 3D point clouds. SPConv, the core of EPN, is based on SE(3)-equivariant group convolutions from KPConv, while E2PN is a lightweight version of EPN. KPPR [56] simplifies KPConv to construct a local feature extraction network with a ResNet-like structure to boost performance.

2.3.3 Graph convolution-based methods

Common graph convolution methods in DL-LPR models include EdgeConv [57] and its variants. In models like LPD-Net [39] and SeqLPD [40], graph convolutional networks are used to aggregate point cloud information in both feature and Cartesian spaces. The residual graph convolution module in DAGC [38] builds on EdgeConv, aggregating features from multi-level neighboring points. PPT-Net [58] incorporates a graph embedding layer before each Pyramid Point Transformer (PPT) module, with EdgeConv at its core, further exploiting geometric structures.

HiBi-GCN [36] uses a hierarchical bidirectional graph convolutional network to learn features from sparse 3D point clouds, improving place representation. However, as the number of channels and neighbors increases, the memory consumption of EdgeConv escalates, and recalculating the k-nearest neighbor (k-NN) graph in feature space increases computational costs. To address this, Hui *et al*. [59] introduced ProxyConv in EPC-Net, which constructs the k-NN graph in the spatial domain, reducing computational costs by keeping the graph static and replacing k-NN points with proxy points.

2.3.4 Transformer-based methods

Transformers, originally developed for natural language processing, have revolutionized computer vision tasks due to their self-attention mechanism [48]. Recent studies have introduced Transformer models into the DL-LPR domain, advancing research in this area.

PPT-Net [58] employs Pyramid Point Transformers to capture spatial relationships between local features of point clouds at different resolutions. To address the limitations of traditional point-wise self-attention, the authors introduced a grouped self-attention module, enhancing the Transformer module in their model. HiTPR [61] employs a hierarchical architecture with both short-range and long-range Transformer modules, improving the learning of correlations between adjacent points and capturing global context dependencies. Ye *et al*. [60] proposed a lightweight point-level Transformer in FPET-Net, efficiently extracting local features from feature points.

2.3.5 Summarize

Methods based on raw 3D point clouds have evolved from the relatively simple PointNet architecture to more complex models. This evolution is driven by the need to improve local feature encoding, enhance robustness to viewpoint variations, and increase network efficiency. Recent research increasingly integrates attention mechanisms and Transformer architectures to refine key components, such as point convolution and graph convolution.

2.4. Methods based on 2D projections

These methods apply projection transformations to raw 3D point clouds, converting them into 2D representations. Depending on the projection technique, 2D-projection-based DL-LPR methods can be categorized into spherical, bird's eye view (BEV), cylindrical, sinogram, and hybrid projection methods. Further sub-categorization is possible based on specific implementation details. Representative methods for each category are summarized in Table 3.

2.4.1 Spherical projection-based methods

For a specific 3D point in a point cloud frame, its spherical projection formula is computed as:

$$\begin{cases} r = \sqrt{x^2 + y^2 + z^2}, \\ \varphi = \arctan(\frac{x}{y}), \\ \theta = \arcsin(\frac{z}{r}), \end{cases} \tag{1}$$

where r represents the distance of the 3D point from the LiDAR center, and θ and φ represent the azimuth and elevation angles in the spherical coordinate system, respectively.

Spherical projection captures more geometric structural information and offers inherent advantages in direction equivalence. As a 3D transformation, it is typically converted into 2D representations, such as range image (RI) or panoramic image (PI), in DL-LPR research. This has led to methods based on RI and PI.

2.4.1.1 Methods based on range imgae

Range images, compared to raw 3D point clouds, offer a more compact structure that preserves local geometric relationships and simplifies feature extraction by eliminating the need for KD-tree construction [62]. Range images also bypass complex voxelization operations, thus obviating the need for 3D sparse convolutions (SP-Conv).

Two primary methods for generating range images are used in DL-LPR: one computes row indices based on the LiDAR's laser ID (Projection By Laser ID, $PBID$), and the other uses the vertical field of view angle (Projection By Elevation Angle, $PBEA$) [62]. These range images are denoted as RI_{PBID} and RI_{PBEA}, respectively. The width and height of RI are denoted as ω and h, with the corresponding column and row indices as u and v.

Ideally, the central column of the range image aligns with the forward direction of the mobile robot. The column index is calculated as:

$$u = \left\lfloor \frac{1}{2}(1 + \frac{\varphi}{\pi}) \cdot \omega \right\rfloor, \tag{2}$$

where $\lfloor \cdot \rfloor$ is the floor operator.

For RI_{PBID}, the row index is computed as:

$$v = l, l \in 1, 2, \ldots N, \tag{3}$$

where l is the ID number of the laser beam, and N is the total number of laser beams.

For RI_{PBEA}, the row index is:

$$v = \left\lfloor \frac{\theta_{up} - \theta}{\theta_{up} - \theta_{down}} \cdot h \right\rfloor, \tag{4}$$

where θ_{up} and θ_{down} are the maximum and minimum elevation angles of the laser beams, respectively.

(1) Methods based on RI_{PBID}

Early methods like LocNet [19][63][64] use a siamese architecture with 2D convolutional neural network (CNN) feature extraction. Later models, such as SMD-Net [65], incorporate additional inputs like normal vectors, intensity, and elevation data to address point cloud sparsity.

Table 3 Representative methods based on 2D projection.

Category	Method	Time	Local Feature Extraction Network	Local Feature Aggregation Network	Loss Function	Ground Truth	Source Code	Dataset
Methods Based on Spherical Projection	LocNet [19,63,64]	2018	Siamese Architecture+Multi-layer 2D Convolution	-	Contrastive Loss	Distance	LocNet	KITTI
	OREOS [66]	2019	Three-Branch Architecture+Multi-layer Convolution	-	Triplet Loss	Distance	-	KITTI NCLT
	OverlapNet [33]	2020	Siamese Architecture+Multi-layer 2D Convolution	-	Sigmoid Loss+ Binary Cross-Entropy Loss	Overlap	OverlapNet	KITTI Ford Campus
	SeqSphereVLAD [71]	2020	SphereVLAD	NetVLAD	Lazy-Quadruplet Loss	Distance	-	KITTI
	SphereVLAD++ [73]	2022	SphereVLAD+ Self-Attention	Attention+ NetVLAD	Lazy-Quadruplet Loss	Distance	-	KITTI360
	DeLightLCD [67]	2022	Siamese Architecture+Multi-layer Depthwise Separable Convolution	-	Binary Cross-Entropy Loss	Distance	-	KITTI Ford Campus
	AttDLNet [68]	2022	DarkNet53 [69]+ Multi-layer Self-Attention	Max Pooling	Cosine Loss	Distance	AttDLNet	KITTI
	OT [35]	2022	Multi-layer Convolution+ Transformer	NetVLAD	Lazy Triplet Loss	Overlap	OT	KITTI Ford Campus Haomo
	SeqOT [70]	2022	Multi-layer Convolution+Single-frame/ Multi-frame-Transformer	GeM	Triplet Loss	Overlap	SeqOT	KITTI NCLT Haomo MulRan
Methods Based on Bird's Eye View	Yin [78]	2018	GAN	-	Adversarial Feature Inference Loss	Distance	-	KITTI NCLT
	DiSCO [79]	2023	UNet+FFT	-	Lazy-Quadruplet Loss	Distance	-	Oxford NCLT MulRan
	BEVPlace [77]	2023	Grouped Convolution	NetVLAD	Lazy Triplet Loss	Distance	BEVPlace	KITTI
Methods Based on Cylindrical Projection	Cao [81]	2022	Multi-Branch Architecture+ Multi-layer Convolution	-	Softmax Cross-Entropy Loss	Distance	-	Oxford

Table 3 *Cont.*

Category	Method	Time	Local Feature Extraction Network	Local Feature Aggregation Network	Loss Function	Ground Truth	Source Code	Dataset
Methods Based on Sinogram Projection	DeepRING [84]	2022	Siamese Architecture+Cycle Convolution-based UNet	-	Cross-Entropy Loss	Distance	-	NCLT MulRan
Category	Method	Time	Local Feature Extraction Network	Local Feature Aggregation Network	Loss Function	Ground Truth	Source Code	Dataset
Methods Based on Hybrid Projection	FusionVLAD [85]	2021	TVE+SVE+ VGG16	NetVLAD	Lazy Triplet Loss	Distance	-	KITTI NCLT
	CVTNet [75]	2023	Multi-layer Convolution+Transformer	NetVLAD	Lazy Triplet Loss	Overlap	CVTNet	KITTI NCLT Haomo

(2) Methods based on RI_{PBEA}

These methods can be further categorized into single-frame and sequence-based approaches. Single-frame methods generally offer higher computational efficiency, while sequence-based methods are more robust to viewpoint variations. Feature extraction networks in these models often employ multi-branch (e.g., siamese or tri-branch) or single-branch architectures, utilizing 2D convolutions, attention mechanisms, or Transformers.

(a) Multi-branch Architecture + Multi-layer CNN

Models like OREOS [66], OverlapNet [33–34], and DeLightLCD [67] use traditional 2D convolutions in their feature extraction networks. DeLightLCD, however, uses depth-wise separable convolutions to reduce parameter sizes and mitigate issues like gradient vanishing and overfitting.

(b) Single-branch + Attention/Transformer Enhancement

AttDLNet [68] uses an improved DarkNet53 [69] combined with multiple self-attention modules to capture long-range contextual dependencies. OT [35] and SeqOT [70] enhance OverlapNetLeg [33–34] with yaw rotation invariance and Transformer-based spatiotemporal feature extraction.

2.4.1.2 Methods based on panoramic image

Yin *et al*. [72] demonstrated that spherical convolutions in the spherical harmonic domain can extract local features from panoramic images invariant to directional changes. By aggregating these features, global descriptors invariant to viewpoint and local translations can be obtained. The seqSphereVLAD model [71–72] pioneers this approach, while SphereVLAD++ [73] introduces attention mechanisms to capture long-range dependencies and improve signal-to-noise ratios in global descriptors.

2.4.2 BEV projection-based methods

BEV projection preserves the rigid structure of the environment in the xy plane while disregarding the z-axis distribution. This projection captures structural information of the environment, with dynamic targets, buildings, and roads forming edges that exhibit good repeatability and stability as the robot moves [74].

In DL-LPR, BEV images are typically generated using metrics such as maximum perception distance [75], maximum height [76], or point cloud density [74,77]. For instance, for a point $p(x, y, z)$

in the point cloud P, if the maximum perception distance is used as the projection criterion, the 2D BEV image coordinates (u,v) can be calculated by [75]:

$$\begin{cases} u = \frac{1}{2}[1 - \frac{\arctan(y,x)}{\pi}] \cdot \omega, \\ v = \frac{r}{f} \cdot h, \end{cases} \tag{5}$$

where $r = \sqrt{x^2 + y^2}$ and f represent the maximum perception distance, respectively. There is considerable structural variation in the local feature extraction networks of such methods.

Yin *et al.* [78] incorporated adversarial and unsupervised learning into feature extraction, improving robustness to viewpoint changes and enabling real-time operation on mobile platforms. DiSCO [79] uses a 2D BEV map generated from Scan Context [22], employing a shared multi-branch U-Net for feature extraction. The model uses Fourier transforms on the feature tensor to exploit translation and rotational invariance in the frequency domain. BEVPlace [77] generates 2D BEV images based on point density, using group convolutions for rotation-invariant feature extraction.

2.4.3 Cylindrical projection-based methods

Cylindrical projections offer a compact representation of 3D point clouds, preserving significant geometric information. The method proposed by [80] addresses sparsity in single-frame point clouds and improves robustness to occlusion and viewpoint changes by accumulating sequence data during projection.

For a LiDAR point $p(x,y,z)$ in Cartesian coordinates, z is first normalized to the range $(0,\pi)$, and its corresponding cylindrical coordinate in the cylindrical space is obtained by:

$$\begin{cases} r = \sqrt{x^2 + y^2}, \\ \theta = \arctan(\frac{x}{y}) + \pi, \quad \theta \in [0, 2\pi) \\ h = \arctan(\frac{z}{l}) + \frac{\pi}{2}, \quad h \in (0, \pi) \end{cases} \tag{6}$$

Where $\arctan(x/y)$ represents the azimuth angle from the origin to point p, with a range of $(-\pi, \pi)$, h is the normalized height, and l is the normalization factor. To ensure viewpoint invariance, the origin is set at the centroid of the point cloud, and the absolute height of the 3D point is used to further mitigate the impact of viewpoint variations. The cylindrical space is then divided into $V = M \cdot N$ voxels, where M and N are the number of divisions for θ and h , respectively. For a point $p_k = (r_k, \theta_k, h_k)$ in the cylindrical space, its representation in the voxel unit $v_{i,j}$ is:

$$\begin{cases} i = \left\lfloor \theta_k \cdot \frac{M}{2\pi} \right\rfloor, \quad \theta_k \in [0, 2\pi) \\ j = \left\lfloor h_k \cdot \frac{N}{\pi} \right\rfloor, \quad h_k \in (0, \pi) \end{cases} \tag{7}$$

To improve computational efficiency, the cylindrical space is divided into voxels, with each voxel treated as a pixel in the cylindrical projection image. The pixel value is computed based on the point within the voxel and its geometric relationship with adjacent points:

$$\begin{cases} 255(1-\dfrac{\alpha}{2\pi}), \\ \alpha_k = \left\langle \overrightarrow{P_{m-1,n-1}, P_{m,n}}, \overrightarrow{P_{m,n}, P_{m+1,n+1}} \right\rangle, \\ P_{m,n} \neq (0,0,0) \\ 0, \qquad\qquad P_{m,n} = (0,0,0) \end{cases} \tag{8}$$

where α is the angle between vectors $\overline{P_{m-1,n-1}, P_{m,n}}$ and $\overline{P_{m,n}, P_{m+1,n+1}}$.

Cao *et al.* [81] developed a lightweight dual-head place classification network using 2D convolutions, pooling layers, and fully connected layers. The model employs large convolution kernels and average pooling layers to mitigate seasonal changes and focus on regional background features. The dual-head design addresses the limited depth of the convolutional network, and incremental learning distinguishes between different scenes for robust topological localization.

2.4.4 Sinogram projection-based methods

To address sparse localization, Lu *et al.* [82] developed the RING (Radon Sinogram) descriptor, a compact, unified representation that is invariant to both orientation and translation. This descriptor is generated using the Radon transform. For each point cloud frame, ground points are first removed to focus on the relevant features. Then, based on the scan context [22], the point cloud is projected into a 2D BEV representation, denoted as a 2D function $f(x, y)$. The Radon transform is applied to $f(x, y)$ to obtain the RING representation of the point cloud, denoted as $R_f(\theta, \tau)$, with its calculation formula given by:

$$\begin{aligned} R_f(\theta,\tau) &= \int_{L: x\cos\theta + y\sin\theta} f(x,y)dxdy \\ &= \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f(x,y)\delta(\tau - x\cos\theta - y\sin\theta)dxdy, \end{aligned} \tag{9}$$

where L represents the integral line parameterized by $x\cos\theta + y\sin\theta = \tau$, $\theta \in [0, 2\pi)$ is the angle between L and the y axis, and $\tau \in (-\infty, \infty)$ is the perpendicular distance from the origin to L.

Xu *et al.* [83] extended RING to RING++, improving robustness for global localization, while Lu *et al.* [84] applied RING in DL-LPR tasks with the DeepRING model, which includes a sine wave feature extraction module and frequency-domain feature aggregation.

2.4.5 Hybrid projection-based methods

To address sparsity, occlusion, and viewpoint variations, Yin *et al.* [85] introduced FusionVLAD, which extracts translation-invariant features from BEV projection and rotation-invariant features from spherical projection. These features are fused using VGG16 [86], significantly improving performance. Ma *et al.* [75] proposed CVTNet to explore and integrate internal and inter-relational information from range images and BEV images. The model uses Intra-Transformer and Inter-Transformer to uncover deep associations within and between data from different viewpoints.

2.4.6 Summarize

2D projection data retain certain features from the original 3D point cloud, such as local geometric structures and yaw rotation invariance. Researchers design specialized local feature extraction networks focusing on rotation invariance, developing DL-LPR models robust to viewpoint changes. These models typically use single-branch or multi-branch architectures. Single-branch models output global feature descriptors, while multi-branch models output similarity scores and may integrate additional tasks like pose estimation. Network architectures have evolved from simple convolutional stacks to advanced attention/Transformer-based designs. Inputs have expanded from single-channel 2D projections to multi-channel and sequential data, enriching information but demanding higher network performance and efficiency.

2.5. Methods based on 3D voxels

The original 3D point cloud often contains fine-grained local details essential for tasks such as segmentation and detection. However, these details may be irrelevant or considered noise for LPR tasks, complicating scene understanding for DL-LPR models. To address this, sparse 3D voxelization has been implemented to reduce unnecessary local details and data size, while preserving the overall structural information of the scene [87].

DL-LPR methods based on 3D voxelization can be categorized into several types depending on the voxelization technique used. These include Cartesian voxelization, spherical voxelization, 3D NDT, and dynamic voxelization, among others. Relevant classical models are summarized in Table 4.

Table 4. Representative methods based on 3D voxels.

Category	Method	Time	Local Feature Extraction Network	Local Feature Aggregation Network	Loss Function	Ground Truth	Source Code	Dataset
Methods Based on Cartesian Voxelization	MinkLoc3D [91]	2020	Feature Pyramid Architecture+3D SP-Conv	GeM	Triplet Loss	Distance	MinkLoc3D	Oxford In-house
	TransLoc3D [93]	2021	3D SP-Conv +Attention	-	Triplet Loss	Distance	TransLoc3D	
	MinkLoc3D-v2 [92]	2022	Feature Pyramid Architecture+3D SP-Conv	GeM	Smoothed AP Loss	Distance	MinkLoc3Dv2	
	SVT-Net [87]	2022	3D SP-Conv +Transformer	GeM	Triplet Loss	Distance	SVT-Net	
Methods Based on Cartesian Voxelization	LCDNet [94]	2022	Three-Branch Architecture+ Feature Pyramid+ 3D SP-Conv	-	Triplet Loss	Distance	LCDNet	KITTI KITTI360
	LoGG3D-Net [96]	2022	Four-Branch Architecture+ UNet+ 3D Point-Voxel Convolution	Second-Order Pooling	Quadruplet Loss	Distance	LoGG3D-Net	KITTI MulRan
Methods Based on Spherical Voxelization	SpoxelNet [99]	2020	Dual-Branch Architecture+ 3D Convolution	NetVLAD	Lazy-Quadruplet Loss	Distance	-	NAVER LABS
	MinkLoc3D-SI [100]	2021	Feature Pyramid Architecture+3D SP-Conv	GeM	Triplet Loss	Distance	MinkLoc3D-SI	Oxford KITTI

Table 4. *Cont.*

Category	Method	Time	Local Feature Extraction Network	Local Feature Aggregation Network	Loss Function	Ground Truth	Source Code	Dataset
Methods Based on 3D NDT	NDT-Transformer [101]	2021	Transformer	NetVLAD	Lazy-Quadruplet Loss	Distance	NDT-Transformer	Oxford
Methods Based on Dynamic Voxelization	GeoLCR [103]	2023	3D SP-Conv + Geometric Transformer	-	Mean Squared Error Loss	Distance	-	KITTI
Others	Kong [105]	2023	Siamese Architecture+Transformer	NetVLAD	-	Distance	-	Oxford In-house KITTI

2.5.1 Cartesian voxelization-based methods

Cartesian voxelization involves partitioning a 3D point cloud into voxel grids within Cartesian coordinates, simplifying the processing of the original point cloud. Siva *et al*. [88] applied representation learning to voxelized data, transforming the LPR problem into a regularized optimization task. Subsequent research has leveraged neural network models to extract meaningful features from voxelized data, effectively capturing location information. Key components of these models include 3D SP-Conv [89], 3D sparse point-voxel convolution [90], and attention/Transformer mechanisms.

2.5.1.1 Methods based on 3D SP-Conv

SP-Conv [89], implemented using the MinkowskiEngine, is designed for efficient processing of sparse data in high-dimensional spaces [87]. Several models, including MinkLoc3D [91], MinkLoc3Dv2 [92], TransLoc3D [93], and SVT-Net [87], are based on SP-Conv. MinkLoc3D and MinkLoc3Dv2 employ a Feature Pyramid Network (FPN) architecture, with MinkLoc3Dv2 improving performance by adding additional convolutional layers, transposed convolutions, and attention modules to enhance network depth and width.

TransLoc3D incorporates attention/Transformer modules to enable the model to process semantic objects of varying sizes using adaptive receptive fields, while SVT-Net utilizes Transformers to capture long-range contextual information.

LCDNet [94] adopts a three-branch architecture, with feature extraction based primarily on PV-RCNN [95]. Depending on the task, LCDNet retains 3D voxel convolution and voxel set abstraction modules from PV-RCNN, which are integrated with four feature pyramid modules constructed using 3D SP-Conv layers. This design effectively combines the fine-grained feature extraction strengths of PointNet-like architectures with the high-level feature extraction advantages of voxel-based methods.

2.5.1.2 Methods based on 3D point-voxel convolution

The LoGG3D-Net [96] model features a four-branch architecture, with its core local feature extraction network built upon 3D point-voxel convolution using SparseConv U-Net. It first maps input points to a high-dimensional feature space using SparseConv U-Net, then applies a local consistency loss to maximize feature similarity for overlapping point clouds. Next, second-order pooling and differentiable eigenvalue power normalization aggregate local features into global scene descriptors.

2.5.1.3 Methods based on Attention/Transformer

MinkLoc3Dv2 [92] improves performance by incorporating the Efficient Channel Attention (ECA) module [97] after certain 3D SP-Conv layers in the FPN. TransLoc3D [93] uses ECA in its self-adaptive receptive field module (ARFM) to aggregate global information while filtering out irrelevant noise features. To further improve contextual information aggregation and reduce the model's parameter count, an External Attention Transformer module is added after the ARFM. Simply stacking 3D SP-Conv layers may overlook long-range contextual information, so SVT-Net [87] introduces two types of Transformers: atom-based sparse voxel Transformers and cluster-based sparse voxel Transformers. These Transformers are designed to capture short-range local features and long-range contextual features within 3D voxels, respectively.

2.5.2 Spherical voxelization-based methods

Spherical voxelization divides the point cloud into voxels within spherical space. Unlike Cartesian voxelization, where the voxel size is fixed, spherical voxelization adjusts the voxel size based on radial distance from the origin, allowing for more accurate structural information representation. This approach reflects the sparsity of laser projections, which typically vary with distance. Spherical voxelization is particularly effective for encoding 3D point clouds into a compact and efficient voxel representation.

For a 3D point cloud frame $P_C(X,Y,Z)$, which contains M points represented by Cartesian coordinates $p_C(x_i,y_i,z_i)$, where $0 \le i \le M$, the corresponding 3D spherical coordinates $P_S(R,\Phi,\Theta)$ can be obtained as shown in Equation (1), and it contains M points represented by spherical coordinates $p_S(\rho,\phi,\theta)$. After spherical voxelization, the voxel representation obtained follows the calculation formula for points contained in each spherical voxel unit $V_{i,j,k}$ follows:

$$\begin{aligned} &V_{i,j,k} = \{P_S(R,\Phi,\Theta) \mid \\ &\Delta\rho * i \le \rho \le \Delta\rho * (i+1), \\ &\Delta\phi * j \le \phi \le \Delta\phi * (j+1), \\ &\Delta\theta * k \le \theta \le \Delta\theta * (k+1)\}, \end{aligned} \tag{10}$$

where, ρ, ϕ, θ represent radial distance, azimuth angle, and elevation angle, respectively, while $\Delta\rho$, $\Delta\phi$, $\Delta\theta$ represent the voxelization resolution along the three axes, i, j, k denote the index of the voxel unit along the three axes, respectively.

SpoxelNet [99] employs a two-branch network design for local feature extraction, with branches operating independently without weight-sharing. Each branch uses 3D convolutional layers, with the fine feature extraction branch capturing detailed structural relationships and the coarse feature extraction branch capturing broader features. A deconvolutional layer ensures consistent output dimensions, enabling efficient feature extraction at different granularities.

MinkLoc3D-SI [100] builds on MinkLoc3D [91], retaining the FPN design and 3D SP-Convs but using spherical voxelization to address uneven density distribution. It incorporates intensity information to enhance robustness to sparsity and viewpoint variation, improving LPR performance.

2.5.3 3D NDT-based methods

3D NDT [11] is widely used in 3D LiDAR SLAM for probabilistic representation of point clouds, such as in point cloud registration. In NDT representation, each point is modeled as a Gaussian distribution, generated by voxelizing the point cloud and calculating the Gaussian distribution for each voxel.

The NDT-Transformer model [101] uses the NDT representation of 3D point clouds as input, preserving geometric information while reducing data size. It employs a standard Transformer [47] to extract geometric and contextual features, aggregated into a global descriptor using the NetVLAD layer.

2.5.4 Dynamic voxelization-based methods

Dynamic voxelization, proposed by Zhou *et al*. [102], allocates storage space for each voxel unit based on the number of points, addressing challenges such as high memory consumption and computational cost in traditional voxelization methods. Figure 7 compares non-dynamic and dynamic voxelization.

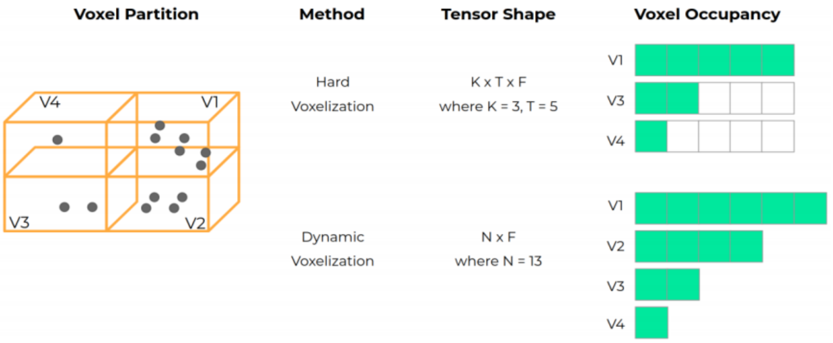


Figure 7. Schematic diagram of the difference between hard voxelization and dynamic voxelization [102].

The GeoLCR model [103] preprocesses 3D point clouds using dynamic voxelization and employs KPConv [50] to hierarchically extract point and voxel features. These features are passed to an overlap estimator using the Geometric Transformer [104] to estimate overlap scores between voxel frames. Loop closure detection is performed by combining overlap scores with pose information from the registration module.

2.5.5 Others

Kong *et al*. [105] proposed the Interest Point-Driven LPR method, inspired by human scene recognition. It uses LeGO-LOAM [106] to extract interest points, projects them onto grid cells, and encodes features using EdgeConv and PointNet. A grid-based U-Transformer and a twin Transformer-NetVLAD LPR module explore local topological relationships and global interactions, generating robust global feature descriptors.

2.5.6 Summary

Voxelization of 3D point clouds reduces data volume and regularizes the data structure, making it easier to process with neural network models. Various voxelization methods have advantages and limitations that affect DL-LPR model performance. Current feature extraction networks in DL-LPR models based on 3D voxels typically include architectures such as feature pyramids and U-Net, with recent advancements incorporating attention mechanisms or Transformers.

2.6. Methods based on semantic

The original 3D point cloud primarily contains low-level geometric structure information, while high-level semantic data obtained through semantic segmentation and other methods is more robust to environmental changes. This high-level information provides stronger constraints for LPR tasks, enhancing the overall performance of LPR systems [25,107]. Chen *et al.* [33–34] demonstrated in OverlapNet that incorporating semantic category information into the model input improves the accuracy of place recognition and loop closure detection.

This section focuses on semantic-based DL-LPR methods that process semantic information within the place feature encoding module, rather than methods used to acquire semantic information. While references [25,108–116] include neural network modules, these modules are primarily designed for semantic information extraction, not processing. The handling of semantic information in these studies relies on traditional techniques, such as clustering, histogram matching, and semantic topological graph matching, and will not be further discussed here.

The network structure of semantic-based DL-LPR models is closely tied to how semantic information is organized. Models that do not construct semantic graphs typically rely on conventional convolutions (e.g., 2D, 3D, or sparse 3D convolutions), while those that organize semantic information as graphs typically employ graph convolutions as their core building block. Some relevant classical models are listed in Table 5.

Table 5. Representative methods based on semantic.

Category	Method	Time	Local Feature Extraction Network	Local Feature Aggregation Network	Loss Function	Ground Truth	Source Code	Dataset
Methods Based on Conventional Convolution	SegMap [117]	2018	Multi-layer Convolution	-	Contrastive Loss	Distance	SegMap	KITTI
	Wietrzykowski [120]	2021	Multi-layer Convolution	-	-	Distance	-	KITTI MulRan
	PSE-Match [107]	2021	Three-Branch Architecture+Spherical Convolution	NetVLAD	Lazy Triplet Loss	Distance	-	KITTI NCLT
	Locus [121]	2021	SegMap-CNN	Second-Order Pooling	-	Distance	Locus	KITTI
	RINet [122]	2022	Siamese Architecture+Multi-layer Convolution	-	Soft Binary Cross-Entropy Loss	Distance	RINet	KITTI KITTI360 NCLT
	PADLoc [123]	2023	Three-Branch Architecture+Feature Pyramid+3D Voxel Convolution	-	Triplet Loss	Distance	PADLoc	KITTI Ford campus In-house Semantic-KITTI
Methods Based on Graph Convolution	SG-PR [124]	2020	Graph Convolution	Fully Connected Layer	Binary Cross-Entropy Loss	Distance	SG-PR	KITTI
	SC-LPR [126]	2022	GRU-EdgeConv++	-	Binary Cross-Entropy Loss	Distance	SC-LPR	KITTI

2.6.1 Conventional convolution-based methods

Dubé *et al*. [18] introduced SegMatch, the first method to leverage semantic segments from 3D LiDAR point clouds for LPR. Later, Dubé *et al.* [117–119] proposed SegMap, a semantic-based DL-LPR method using multi-layer 3D convolutions to extract feature descriptors from semantic segments. Wietrzykowski *et al*. [120] extended SegMap by incorporating both semantic segment data and point cloud intensity, enhancing the feature descriptors. Vidanapathirana *et al*. [121] introduced the Locus model, which fully exploits spatiotemporal relationships between semantic segments. This model first encodes semantic segments using SegMap-CNN [118], then applies spatial and temporal pooling, followed by second-order pooling to generate fixed-length global descriptors.

To improve descriptor robustness to viewpoint changes, Li *et al.* [122] proposed RINet, a twin network with rotation-invariant properties.

Building on LCDNet [94], Arce *et al.* [123] proposed PADLoC, a Transformer-based framework for loop closure detection and point cloud registration. PADLoC redefines the matching problem as both a semantic label classification task and an instance label graph connectivity assignment. The feature extraction module follows LCDNet, but in the matching module, the Transformer processes geometric and panoramic semantic labels for better internal structure utilization.

Yin *et al.* [107] introduced PSE-Match, a viewpoint-invariant model based on semantic analysis. The model performs spherical projection of static semantic objects (e.g., roads, buildings, and static targets) obtained through semantic segmentation. It uses spherical convolutions to encode and aggregate features of each semantic object in parallel, generating rotation-invariant feature descriptors. Divergence learning metrics are employed to further enhance the invariance of the descriptors to translation and viewpoint changes.

2.6.2 Graph convolution-based methods

Inspired by human perception, Kong *et al.* [124] proposed a DL-LPR method utilizing semantic graph representation and graph matching. Semantic information is extracted using the RangeNet++ model [125], and the point cloud is transformed into a concise graph representation, capturing key semantic details and topological relationships. The feature extraction network consists of the Node Embedding module (using EdgeConv [57] to extract spatial and semantic node features) and the Graph Embedding module (using attention to weigh node embeddings).

Dai *et al.* [126] proposed SC-LPR, a model that integrates spatiotemporal context and semantic information from sequential point clouds. It uses RangeNet++ for semantic labeling, discarding less useful categories such as roads and pedestrians. The DBSCAN algorithm [127] clusters the point cloud to extract semantic instance segments, creating the semantic graph. A GRU-EdgeConv++ network extracts features from the graph and aggregates spatiotemporal information. Similarity between point cloud frames is evaluated using the Cosine Tensor Network and Neural Tensor Network combination.

2.6.3 Summary

Semantic data, as high-level information, may sacrifice some low-level geometric details when used in isolation, and its quality depends on the effectiveness of the extraction method. To address this, semantic-based DL-LPR methods are evolving in two key directions: At the input level, researchers focus on improving semantic extraction techniques for stability and accuracy, while exploring ways to integrate semantic information with low-level geometric details to preserve scene information. At the feature extraction level, networks are increasingly adopting fusion strategies that combine components such as graph convolutions, attention mechanisms, and Transformers to enhance the capture of local features.

3. Common datasets and evaluation metrics

3.1. Common datasets

A series of open-source datasets in mobile robotics have significantly advanced research in DL-LPR. This section reviews some of the most commonly used public datasets, summarizing their relevant attributes in Table 6.

The Ford Campus dataset [128], collected between November and December 2009 at the Ford Research Campus and downtown Dearborn, Michigan, includes several loops of varying sizes, making it suitable for testing SLAM and place recognition algorithms. While data was collected using different types of LiDAR, most DL-LPR studies have utilized point clouds from the Velodyne HDL-64E sensor to assess the generalization performance of networks [33–35,67,123].

The KITTI dataset [129] comprises 22 sequences of point clouds, though DL-LPR research typically focuses on sequences 00-10, which provide ground truth poses for model evaluation. Among these, sequences 00, 02, 05, 06, 07, and 08 feature trajectory overlaps (*i.e.*, loops), with sequence 08 containing a reverse loop. A common usage strategy is to train models on sequences 03-10, validate on sequence 02, and evaluate on sequence 00 [35].

The NCLT dataset [130], collected at the University of Michigan's North Campus, includes both indoor and outdoor scenes, with point clouds captured using the Velodyne HDL-32. Data was collected over 15 months, with new data approximately every two weeks. Like the Oxford RobotCar dataset [131], NCLT features significant variations in seasons, weather, lighting, viewpoints, and scene appearance, and contains many dynamic objects, placing high demands on DL-LPR models.

The Oxford RobotCar dataset [131], created by the University of Oxford's Mobile Robotics Group, was collected over 1000 kilometers of driving for autonomous driving research. Data was gathered from May 2014 to December 2015 by driving an experimental car through central Oxford twice a week. As a result, the dataset includes variations in seasons, weather, lighting, and viewpoints. The LiDAR used for point cloud collection is 2D, and in PointNetVLAD, 2D point clouds are aggregated into local 3D maps, with redundant ground points removed. The submaps are downsampled to 4096 points using a voxel grid filter, and their coordinates are transformed to a specific range. Each submap's center is labeled with UTM coordinates for model training and evaluation. Subsequent DL-LPR methods using this dataset generally follow PointNetVLAD's preprocessing steps.

The In-House dataset [21], collected by Mikaela *et al*. using the Velodyne-64 LiDAR, was designed to validate the generalization ability of PointNetVLAD and improve network performance. It includes three main scenes: University Sector (U.S.), Residential Area (R.A.), and Business District (B.D.), with preprocessing following the same approach as PointNetVLAD's handling of the Oxford RobotCar dataset. The In-House dataset has since become a widely used benchmark for evaluating DL-LPR methods.

The SemanticKITTI dataset [132], derived from the KITTI dataset, provides point-level semantic annotations for the 22 KITTI sequences, including full 360-degree field-of-view labeling. This dataset is particularly useful for semantic-based DL-LPR methods, providing rich semantic labels to enhance model recognition capabilities.

The MulRan dataset [133] focuses on place recognition using distance sensors, including millimeter-wave radar and LiDAR. Point clouds were captured with an Ouster 64-line LiDAR sensor in four distinct environments: DCC, KAIST, Riverside, and Sejong City. The dataset is designed to test the robustness of range-based place recognition methods, particularly with challenges such as structural diversity, dynamic objects, reverse loops, repeated scenes, and changes in scene appearance

Table 6. Datasets commonly used in DL-LPR and their instructions.

Dataset	Year	LiDAR Type	Scene Type	Ground Truth of Place	Variation Factors					Related Literature
					Season	Weather	Day/ Night	Scene Appearance	Dynamic Objects	
Ford Campus [128]	2011	3D Velodyne-HDL-64E	Campus	6DoF Pose		√		√	√	B: [33-35, 67] D: [123]
KITTI [129]	2012	3D Velodyne HDL-64E	Urban Area	6DoF Pose					√	A: [43, 60] B: [33-35, 63-68, 70-73, 75, 78, 85] C: [94, 96, 100] D: [110-113, 121-124, 126]
NCLT [130]	2016	3D Velodyne HDL-32	Campus	6DoF Pose	√	√	√	√	√	B: [65, 70, 75, 78, 80, 85] C: [88] D: [122]
Oxford Robotcar [131]	2017	2D SICK LMS-151	Urban Area	UTM Coordinates	√	√	√	√	√	A: [21, 36-40, 45, 48, 51, 53, 56, 58, 59, 61, 105] B: [80, 81] C: [87, 91-93, 100, 101]
In House [21]	2018	3D Velodyne HDL-64	Campus/ Residential/ Commercial Area	UTM Coordinates		√			√	A: [21, 36-39, 45, 48, 53, 56, 59, 61, 105] C: [87, 91-93]
Semantic-KITTI [132]	2019	3D Velodyne HDL-64E	Urban Area	6DoF Pose					√	A: [105] D: [112, 113, 123, 124]
MulRan [133]	2020	3D Ouster-OS1-64	Urban Area	6DoF Pose				√	√	A: [103, 140] B: [70, 79] C: [96] D: [120]
KITTI-360 [134]	2022	3D Velodyne HDL-64E	Suburb	6DoF Pose					√	B: [73] C: [94] D: [122]
Haomo [35]	2022	3D HESAI-PandarXT-32	Urban Area	6DoF Pose		√		√	√	D: [35, 70, 75]
Wild-Place [135]	2023	3D Velodyne VLP-16	Unstructured Natural Scene	6DoF Pose		√		√	√	C: [92, 93, 96]

Note: A represents methods based on original 3D point clouds; B represents methods based on 2D projection; C represents methods based on 3D voxelization; D represents methods based on semantics.

The KITTI-360 dataset [134], collected in the Karlsruhe area, introduces richer input modalities, comprehensive semantic instance annotations, and more accurate localization data, making it a valuable supplement to the original KITTI dataset. For DL-LPR research, it includes 11 sequences, six of which contain loops: 0000, 0002, 0004-0006, and 0009. These sequences feature more loops and reverse loops than the KITTI dataset, presenting additional challenges. The rich semantic annotations in KITTI-360, similar to those in SemanticKITTI, provide ground truth semantic labels, enhancing its utility for semantic-based DL-LPR methods.

The Haomo dataset [35] was captured using the HESAI PandarXT-32 LiDAR sensor, spanning approximately two months and consisting of five sequences. Sequences 1-1 and 1-2 share the same trajectory but with opposite directions, while Sequence 1-3, collected 20 days later, shares the same movement direction as Sequence 1-1. This dataset is useful for testing the robustness of DL-LPR methods against extreme viewpoint changes (e.g., reverse direction) and variations in scene appearance.

The Wild-Places dataset [135], created by Knights *et al.*, was specifically designed for LPR research in unstructured environments. Collected over 14 months using a Velodyne 16-line LiDAR sensor in natural outdoor settings, it includes 67,000 frames of undistorted LiDAR point cloud submaps, divided into 8 sequences. The dataset features abundant intra-sequence and inter-sequence loops, making it suitable for testing DL-LPR models in complex, unstructured environments.

3.2. Performance evaluation metrics

The performance evaluation of DL-LPR models primarily focuses on three aspects: place recognition performance, model generalization ability, and computational efficiency, as illustrated in Figure 8. The analysis is conducted from these three perspectives, with a comparative assessment of the performance of various DL-LPR methods.

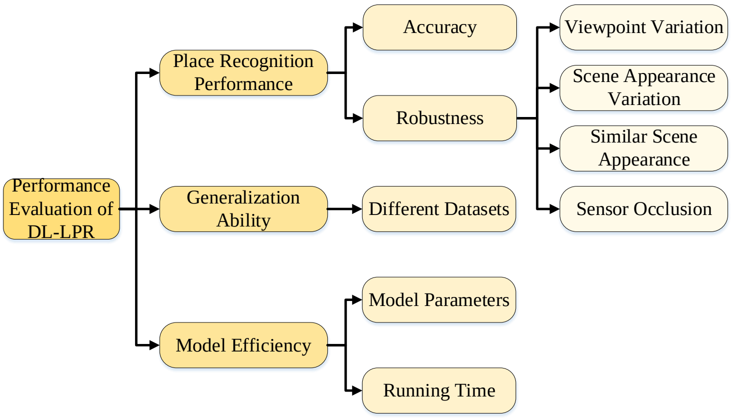


Figure 8. Different aspects of performance evaluation of existing DL-LPR methods.

3.2.1 Place recognition performance

Place recognition performance is evaluated from two primary aspects: recognition accuracy and algorithm robustness.

(1) Place recognition accuracy

Place recognition accuracy is typically assessed using several quantitative and qualitative metrics, including Precision (P), Recall (R), Area Under the Precision-Recall Curve (AUC), Maximum F1 score, Average recall at top 1% (AR@1%), Average recall at top 1 (AR@1), and the Precision-Recall (PR) curve.

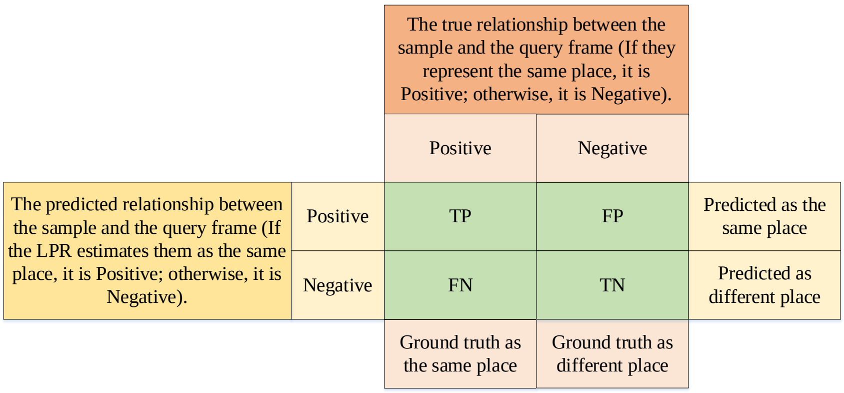
Table 7. The AR@1% and AR@1 of some algorithms on the Oxford and In-House datasets.

Methods		Oxford RobotCar[131]		In-House[21]					
				U.S.		R.A.		B.D.	
		AR@1%	AR@1	AR@1%	AR@1	AR@1%	AR@1	AR@1%	AR@1
Methods Based on Raw 3D Point Clouds	PointNetVLAD [21]	80.09	63.33	90.10	86.07	93.07	82.66	86.49	80.11
	PCAN [37]	86.40	70.72	94.07	83.69	92.27	82.26	87.00	80.31
	LPD-Net [39]	94.92	86.28	96.00	-	90.46	-	89.14	-
	seqLPD [40]	95.81	87.15	-	-	-	-	-	-
	DAGC [38]	87.78	71.39	94.29	86.34	93.36	82.78	88.51	81.29
	SOE-Net [45]	96.43	**89.28**	97.67	**91.75**	95.90	**90.19**	92.59	**88.96**
	PPT-Net [58]	98.40	-	**99.70**	-	**99.50**	-	**95.30**	-
	EPC-Net [59]	94.74	86.23	96.52	-	88.58	-	84.92	-
	HiTPR [61]	93.71	86.63	90.21	80.86	87.16	78.16	79.79	74.26
	KPPR [56]	97.08	-	98.01	-	95.10	-	92.09	-
	E2PN-GeM [53]	93.24	84.79	95.29	88.08	90.46	83.67	87.68	83.29
	HiBi-Net [36]	-	87.46	-	87.81	-	85.76	-	83.03
	RI-STV [105]	**98.50**	-	97.30	-	93.00	-	91.70	-
Methods Based on 3D Voxel	MinkLoc3D [91]	98.50	94.80	99.70	97.20	99.30	96.70	96.70	94.00
	NDT-Transformer [101]	97.65	93.80	-	-	-	-	-	-
	TransLoc3D [93]	98.50	95.00	**99.80**	97.50	**99.70**	97.30	97.40	94.80
	MinkLoc3Dv2 [92]	**99.10**	**96.90**	99.70	**99.00**	99.40	**98.30**	**99.10**	**97.60**
	SVT-Net [87]	97.80	93.70	96.50	90.10	92.70	84.30	90.70	85.50

The F1 score represents the harmonic mean of P and R, treating both as equally important for overall classification performance. To address the limitations of AUC in evaluating VPR algorithms, Ferrarini *et al.* [136] introduced the Extended Precision (EP) metric, which was later applied by Li *et al.* [112] in the LPR field. The formulas for calculating P, R, F1, and EP are provided in Equation (11):

$$\begin{cases} P = \dfrac{TP}{TP+FP}, \\ R = \dfrac{TP}{TP+FN}, \\ F1 = \dfrac{2PR}{P+R}, \\ EP = \dfrac{1}{2}(P_{R_0} + R_{P_{100}}), \end{cases} \tag{11}$$

where TP, FP, and FN are defined as illustrated in Figure 9. P_{R_0} represents precision at the minimum recall point, and $R_{P_{100}}$ represents the maximum recall at 100% precision. In Figure 9, the term "same place” refers to situations where the retrieved or matched place corresponds to the query place. Two definitions of “same place” are used: distance-based (e.g., PointNetVLAD [21], where places are considered the same if their geographic distance is less than 25 meters) and overlap-based (e.g., OverlapNet [33–34] and OT [35], where places are deemed identical if their overlap ratio exceeds 0.3). Table 7 presents the AR@1% and AR@1 metrics for various DL-LPR methods on the Oxford RobotCar and In-House datasets.



		The true relationship between the sample and the query frame (If they represent the same place, it is Positive; otherwise, it is Negative).		
		Positive	Negative	
The predicted relationship between the sample and the query frame (If the LPR estimates them as the same place, it is Positive; otherwise, it is Negative).	Positive	TP	FP	Predicted as the same place
	Negative	FN	TN	Predicted as different place
		Ground truth as the same place	Ground truth as different place	

Figure 9. The meaning of some relevant variables.

(2) Robustness of models

The robustness of DL-LPR methods is primarily challenged by factors such as viewpoint changes, scene appearance variations, similar scene appearances, and occlusion, as discussed in Section 2.2. Table 8 presents experiments evaluating the robustness of various DL-LPR methods to occlusion and viewpoint changes using sequence 00 from the KITTI dataset. In these experiments, occlusion is simulated by randomly deleting points within a specific range, while viewpoint changes are simulated by randomly rotating the point clouds [60]. As shown in Table 8, both occlusion and viewpoint changes result in a decrease in the maximum F1 score for several methods. However, the extent of the decrease varies, highlighting differences in their robustness to these challenges.

Table 8 The maximum F1 scores of some methods before and after occlusion and viewpoint changes in sequence 00 of KITTI dataset [60].

		PointNetVLAD [21]	LPD-Net [39]	FEPT-Net [60]
Occlusion	Before	0.866	0.814	0.971
	After	0.884	0.791	0.955
	Difference	0.018	−0.023	−0.016
Viewpoint Variation	Before	0.866	0.814	0.971
	After	0.839	0.765	0.940
	Difference	−0.027	−0.049	−0.031

Note: “Before” refers to the maximum F1 score before the occurrence of occlusion or viewpoint changes; “After” refers to the maximum F1 score after the occurrence of occlusion or viewpoint changes; the “Difference” is calculated by subtracting the maximum F1 score before the change from the maximum F1 score after the change.

3.2.2 Generalization Performance

Generalization ability is a crucial metric for evaluating neural network models. Since most DL-LPR datasets are collected from structured environments, it is important to note that significant variations exist within these environments. For example, campus environments typically feature more vegetation, urban street scenes involve dynamic pedestrians and vehicles, and highway environments exhibit repetitive geometric patterns.

Table 9. Generalization performance (F1 score) of some algorithms on different data sets.

	PointNetVLAD [21]	OverlapNet [33, 34]	OT [35]
KITTI	0.846	0.865	0.877
Ford Campus	0.830	0.843	0.856

The generalization ability of DL-LPR models is typically assessed by evaluating place recognition accuracy across different datasets. The standard evaluation procedure involves training and validating the model on sequences from a specific structured scene dataset, testing it on sequences from the same dataset, and then testing it on sequences from different datasets. This allows for the comparison of place recognition accuracy across varying environments. Table 9 presents experiments on the generalization performance of several methods using sequence 00 from both the KITTI and Ford Campus datasets. In these experiments, the models were trained on sequences 03-10 of the KITTI dataset, validated on sequence 02 of the same dataset, and tested on the Ford Campus dataset. This experiment demonstrates the generalization ability of DL-LPR methods across different datasets and environments. The results, presented as maximum F1 scores in Table 9, clearly show that different DL-LPR methods exhibit varying degrees of generalization ability across datasets and scene types.

3.2.3 Algorithmic efficiency

The ultimate goal of DL-LPR research is to deploy models on mobile robots for real-time place recognition in dynamic environments. However, mobile robots often have limited computing resources, making algorithm efficiency crucial for successful deployment.

Algorithm efficiency is influenced by two primary factors: model complexity and runtime. Model complexity is directly related to the number of parameters. Overly complex architectures can result in excessive parameters, making deployment difficult and significantly extending inference times.

The algorithm runtime consists of two components: data preprocessing time and model inference time. For DL-LPR methods based on 2D projections, 3D voxels, and semantics, preprocessing of raw 3D point clouds is required. This step may involve complex mathematical operations, leading to longer processing times. Model inference time is influenced by both model complexity and hardware computational power.

To improve algorithm efficiency, it is essential to focus on lightweight network design and reducing data preprocessing time. These factors are key to enhancing real-time performance. Table 10 compares the number of model parameters and algorithm runtimes for several DL-LPR methods, using an Intel Core i7-6950X CPU and an NVIDIA GeForce GTX 1080 Ti GPU with 12GB of VRAM [60].

Table 10. Model parameters and running time of some algorithms [60].

Methods	Model Parameters（M）	Running Time（ms）
PCAN [37]	20.4	39.7
PointNetVLAD [21]	19.8	34.4
LPD-Net [39]	19.8	96.94
EPC-Net [59]	4.7	32.82
FPET-Net [60]	1.77	7.6

4 Analysis of challenges and outlook for research trends

The place recognition task for mobile robots is increasingly characterized by long-term, large-scale, and highly dynamic environments [1], which introduce several new challenges:

(1) **Temporal dimension**: Operations span across different times of day, seasons, and weather conditions (e.g., rain, snow, fog, dust), leading to significant changes in scene appearance.

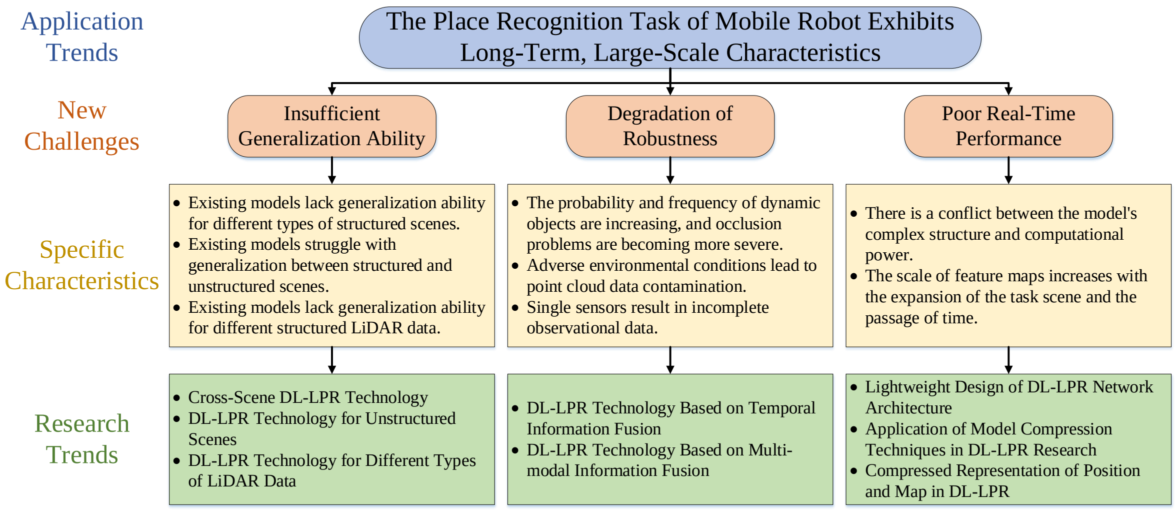

Application Trends
The Place Recognition Task of Mobile Robot Exhibits Long-Term, Large-Scale Characteristics
New Challenges
Insufficient Generalization Ability
Degradation of Robustness
Poor Real-Time Performance
Specific Characteristics
• Existing models lack generalization ability for different types of structured scenes.
• Existing models struggle with generalization between structured and unstructured scenes.
• Existing models lack generalization ability for different structured LiDAR data.
• The probability and frequency of dynamic objects are increasing, and occlusion problems are becoming more severe.
• Adverse environmental conditions lead to point cloud data contamination.
• Single sensors result in incomplete observational data.
• There is a conflict between the model's complex structure and computational power.
• The scale of feature maps increases with the expansion of the task scene and the passage of time.
Research Trends
• Cross-Scene DL-LPR Technology
• DL-LPR Technology for Unstructured Scenes
• DL-LPR Technology for Different Types of LiDAR Data
• DL-LPR Technology Based on Temporal Information Fusion
• DL-LPR Technology Based on Multi-modal Information Fusion
• Lightweight Design of DL-LPR Network Architecture
• Application of Model Compression Techniques in DL-LPR Research
• Compressed Representation of Position and Map in DL-LPR

Figure 10. Challenges and future development trends of DL-LPR research.

(2) **Spatial dimension**: The scale of operational environments is expanding, often involving cross-scenario operations, increasing the size of scene maps and search spaces.

(3) **Dynamic targets**: As operational time and range increase, so does the frequency of encountering dynamic targets, exacerbating occlusion problems.

These emerging challenges emphasize the growing demands on DL-LPR research, particularly regarding model generalization, robustness, and real-time performance. They also point to key future research directions, as illustrated in Figure 10.

4.1 Challenge analysis

While many DL-LPR methods perform well on single, limited-scale datasets, new challenges arise in long-term, large-scale, and high-dynamic tasks. This section analyzes these challenges in relation to model generalization ability, robustness, and real-time performance.

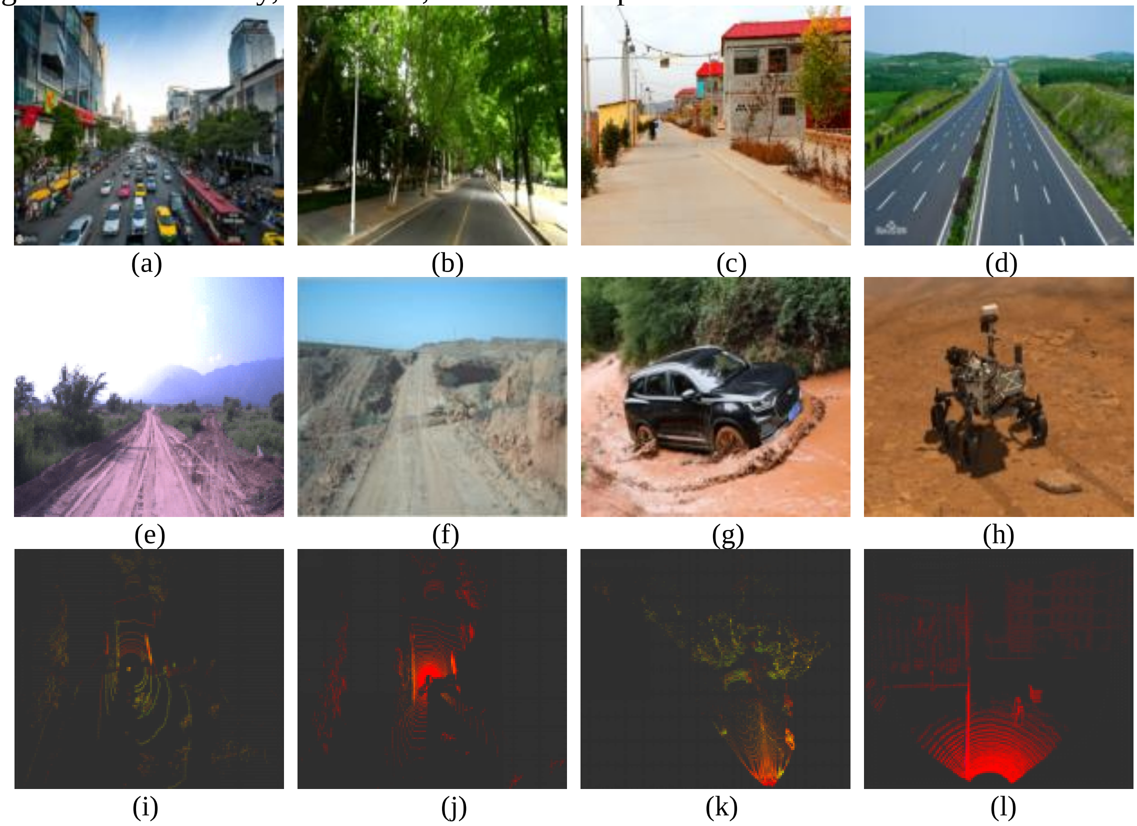


Figure 11. Diversity of scenes and Lidar data (The first row are examples of different types of structured scenes, the second row are examples of different types of unstructured scenes, and the third row are examples of different types of LiDAR data).

4.1.1 Insufficient generalization capacity

For long-term, large-scale, and high-dynamic LPR tasks, current DL-LPR models exhibit insufficient generalization capability in three main areas:

(1) **Generalization across different types of structured scenes**: Existing models show limited ability to generalize across diverse structured scenes. While generalization ability has been partially explored (see Section 3.2.2), a deeper analysis is needed for long-term, large-scale, and dynamic environments. With urbanization accelerating, autonomous driving and mobile robotics in urban areas are gaining increasing attention. Large public datasets have greatly advanced DL-LPR in structured scenes, but the generalization ability of current models remains inadequate. This is primarily due to the diversity of structured scenes, as shown in Figure 11, which includes urban neighborhoods, campuses, suburban areas, and highways. These scenes vary in terms of dynamic targets, road congestion, vegetation, and recurring scene types. A model trained on a single dataset may struggle to generalize to other environments. Therefore, developing models that can adapt to various structured scenes and enhancing their learning capacity are critical steps in addressing this challenge.

(2) **Generalization between structured and unstructured scenes**: Models also struggle with generalization between structured and unstructured scenes. The differences between these scene types are substantial. Figure 11 shows unstructured scenes such as off-road areas, deserts, and extraterrestrial terrains, which are more chaotic and less structured than urban scenes. Unstructured environments feature simpler elements, uneven terrain, and sparse structural features, making it difficult for DL-LPR models to adapt. Additionally, limited research and the scarcity of public datasets for unstructured environments hinder progress. For example, Knights *et al.* [135] released the first dataset for LPR in unstructured scenes, but it has not been widely adopted, limiting further development in this area.

(3) **Generalization across different types of LiDAR data**: Models face challenges in generalizing across different types of LiDAR data. As the use of diverse LiDAR systems grows, point clouds generated by systems with different line counts, installation positions, and working principles exhibit significant variations in data density and distribution. Figure 11 highlights these differences, with point clouds from Velodyne, Ouster, and Livox LiDAR systems showing distinct characteristics. These differences make it difficult to train DL-LPR models on one type of LiDAR data and apply them to others.

4.1.2 Robustness degradation

The robustness of current DL-LPR models degrades in long-term, large-scale, and high-dynamic environments due to the following factors:

(1) **Occlusion caused by dynamic targets**: Occlusion becomes more problematic in dynamic environments where the presence of moving objects obstructs the LiDAR sensor's field of view. Occlusion blocks some laser beams, resulting in incomplete observations and data distribution, which can degrade model performance. The random and unpredictable nature of dynamic targets exacerbates this issue, increasing the demands on the model's robustness.

(2) **Environmental contamination of point clouds**: Adverse environmental conditions—such as rain, snow, fog, and dust—can contaminate point clouds, impairing model robustness. These conditions scatter or absorb LiDAR laser beams, reducing the energy and accuracy of distance measurements. Dust introduces random noise, further degrading the quality of the point cloud data.

(3) **Limitations of single-sensor systems**: LiDAR sensors provide detailed geometric information but lack the ability to capture additional scene details like color or the motion of dynamic objects. Relying solely on LiDAR data can lead to incomplete representations of the environment, limiting model robustness. Incorporating additional sensors (e.g., cameras or radar) can provide complementary information, such as color and object dynamics, improving robustness in complex, changing environments.

4.1.3 Lack of real-time performance

In long-term, large-scale, and dynamic DL-LPR tasks, real-time performance is significantly affected by the following factors:

(1) **Model complexity and computational power**: There is a trade-off between model complexity and available computational resources. More complex neural network architectures can capture more intricate features, improving the model's ability to handle long-term, large-scale, and dynamic environments. However, mobile robots typically have limited computational power, which makes

deploying complex models challenging. Even if a model is deployed, higher complexity increases the number of parameters, resulting in slower inference and making it difficult to meet real-time performance requirements.

(2) **Feature map scale and search space**: As the operational scenario expands and the task duration increases, the feature map scale grows, requiring more storage resources and increasing the search space. This makes place recognition and matching more time-consuming. Existing DL-LPR methods struggle to handle the larger feature maps and expanded search spaces required in long-term, large-scale, and dynamic environments, limiting their real-time performance.

4.2 Outlook for research trends

In the context of long-term, large-scale, and highly dynamic tasks, this paper outlines the following research trends and future directions for DL-LPR based on the challenges identified in existing models.

4.2.1 Generalization capacity enhancement

(1) **DL-LPR technology for cross-scene**: To address the challenge of limited generalization across different types of structured scenes and between structured and unstructured environments, research in cross-scene place recognition aims to bridge these gaps. Efforts in this domain aim to enhance the generalization capability of DL-LPR models. For instance, Yu *et al.* [138] proposed a method that leverages multi-modal information fusion, which improves the model's ability to express place features by exploiting the full range of available data. Additionally, Knights *et al*. [139] explored incremental learning techniques to enhance the model's adaptability and generalization, particularly for dynamic and previously unseen scenes.

(2) **DL-LPR technology for unstructured scenes**: While most DL-LPR research has focused on structured environments, the growing importance of unstructured environments, such as deserts, swamps, and extraterrestrial terrains, requires dedicated research. This is particularly relevant with the rise of space exploration and the need for autonomous systems to operate in harsh, remote environments. Advances in this area will improve DL-LPR performance in unstructured environments and contribute to generalization across both structured and unstructured environments. A pioneering effort in this area was made by Knights *et al*. [135], who contributed foundational work on DL-LPR for unstructured environments, addressing challenges at the data level.

(3) **DL-LPR technology for different types of LiDAR data**: In place recognition tasks, LiDAR data used to build feature databases is typically fixed for a given environment. However, during feature retrieval, query frames may be captured using different types of LiDAR systems. Various LiDAR systems — differing in beam counts, fields of view, and working principles — produce distinct distributions of observation data, even when capturing the same places. If these data distribution differences can be modeled, it may be possible to adapt or transform query data to match the style of the feature database, enhancing the model's generalization across diverse LiDAR systems. Promising approaches include domain adaptation and Generative Adversarial Networks (GANs). For example, Qiao *et al.* [140] used domain adaptation to address the LPR problem by training on simulated data and testing with real-world data, while Yin *et al*. [141–142] employed GANs for style transfer between millimeter-wave radar and LiDAR data, providing valuable insights into cross-sensor data adaptation.

4.2.2 Robustness improvement

(1) **Temporal information fusion for DL-LPR**: To mitigate occlusion caused by sensor placement and dynamic targets, temporal information fusion strategies can enhance model robustness. Temporal information, consisting of observations from various perspectives over time, can reduce the effects of occlusion on data distribution and address the sparsity of point cloud features in individual frames. Studies [70–72] have demonstrated the effectiveness of such fusion strategies in improving robustness.

(2) **Multi-modal information fusion for DL-LPR**: To address challenges such as environmental contamination and incomplete sensor data, multi-modal fusion strategies can enhance model robustness. Multi-modal fusion can take two forms: (1) combining different modalities from a single sensor, such as depth, intensity, normal vectors, and semantics associated with 3D point clouds [33–34,68,75,85,143], and (2) integrating data from heterogeneous sensors, such as combining LiDAR, camera, and millimeter-wave radar data [138,144–146]. However, issues like cross-modal place recognition and the relationships between modalities remain key areas for further exploration.

4.2.3 Real-time performance improvement

(1) **Lightweight design of DL-LPR models**: To address the trade-off between model complexity and available computational resources, lightweight network architectures can be developed to enhance real-time performance. Several studies, such as those by [59,87], have made progress in this area. However, more research is needed to design lightweight architectures that maintain place recognition accuracy for long-term, large-scale, and dynamic tasks.

(2) **Application of model compression techniques in DL-LPR research**: To balance model performance with real-time capabilities, model compression techniques can complement lightweight network architectures, allowing large models that offer superior place recognition performance to be deployed within the computational constraints of mobile robots. Although model compression has rarely been explored in DL-LPR, its successful application in fields like natural language processing [147] and object detection [148] suggests it could be a promising direction for DL-LPR research.

(3) **Compressed representation of place and Map**: To address the growing size of feature maps as task scenarios and durations expand, compressed representations of places and feature maps can reduce storage demands and improve real-time performance. Wiesmann *et al.* [149] have made initial strides in this area by designing network models for compressed feature representation of point clouds, demonstrating that place feature maps can be constructed with reduced memory requirements while still meeting performance and real-time constraints. This research direction shows significant potential for further development.

5 Conclusion

This paper presents a comprehensive review of the current state of research in DL-LPR technology, focusing on fundamental concepts, method classifications, key components, evaluation metrics, challenges, and emerging research trends. A “coarse-to-fine” classification framework is adopted, categorizing existing methods from both the perspectives of input data structure and model network architecture. The review not only covers techniques for generating various types of structural data but also provides an in-depth analysis of the network architectures of corresponding DL-LPR models, offering valuable insights for future research.

Looking ahead, as DL-LPR technology becomes more widely adopted, there will be growing demands for improvements in generalization, robustness, and real-time performance. The continued evolution of deep learning and related technologies will provide new tools for model design. Moreover, the exploration of diverse data structures will offer multi-dimensional insights into scene characteristics. Thus, the integrated optimization of model architectures and input data structures is expected to be a key focus in the future development of DL-LPR technology.

Acknowledgments

This work was funded by internal projects of the Defense Innovation Institute.

Authors' contribution

Conceptualization, Weizhong Jiang and Shubin Si; methodology, Hanzhang Xue; software, Weizhong Jiang; validation, Yiming Nie, Zhipeng Xiao and Qi Zhu; formal analysis, Liang Xiao; investigation, Weizhong Jiang and Qi Zhu; resources, Weizhong Jiang and Zhipeng Xiao; data curation, Weizhong Jiang and Qi Zhu; writing—original draft preparation, Weizhong Jiang; writing—review and editing, Shubin Si, Qi Zhu and Liang Xiao; visualization, Weizhong Jiang; supervision, Liang Xiao; project administration, Yiming Nie; funding acquisition, Yiming Nie and Liang Xiao. All authors have read and agreed to the published version of the manuscript.

Conflicts of interests

The authors declare no conflict of interest.

Reference

[1] Yin P, Zhao S, Cisneros I, Abuduweili A, Huang G, *et al.* General place recognition survey: towards the real-world autonomy age. *arXiv* 2022, arXiv:2209.04497.

[2] Shi P, Zhang Y, Li J. LiDAR-based place recognition for autonomous driving: a survey. *arXiv* 2023, arXiv:2306.10561.

[3] Lowry S, Sünderhauf N, Newman P, Leonard JJ, Cox D, *et al.* Visual place recognition: a survey. *IEEE Transactions on Robotics* 2016, 32(1):1–19.

[4] Zeng Z, Zhang J, Wang X, Chen Y, Zhu C. Place recognition: an overview of vision perspective. *Applied Sciences* 2018, 8(11):2257.

[5] Zhang X, Wang L, Su Y. Visual place recognition: a survey from deep learning perspective. *Pattern Recogn.* 2021, 113:107760.

[6] Schubert S, Neubert P. What makes visual place recognition easy or hard. *arXiv* 2021, arXiv:2106.12671.

[7] Garg S, Fischer T, Milford M. Where is your place, visual place recognition. *arXiv* 2021, arXiv:2103.06443.

[8] Masone C, Caputo B. A survey on deep visual place recognition. *IEEE Access* 2021, 9:19516–19547.

[9] Barros T, Pereira R, Garrote L, Premebida C, Nunes UJ. Place recognition survey: an update on deep learning approaches. *arXiv* 2021, arXiv:2106.10458.

[10] Magnusson M, Andreasson H, Nuchter A, Lilienthal AJ. Appearance-based loop detection from

3D laser data using the normal distributions transform. In *2009 IEEE International Conference on Robotics and Automation*, 12–17 May 2009, pp. 23–28.

[11] Biber P, Strasser W. The normal distributions transform: a new approach to laser scan matching. In *Proceedings 2003 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2003) (Cat. No.03CH37453)*, 27–31 October 2003, pp. 2743–2748.

[12] Deng J, Dong W, Socher R, Li LJ, Kai L, et al. ImageNet: a large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, 20–25 June 2009, pp. 248–255.

[13] Si N, Zhang W, Qu D, Luo X, Chang H, Niu T. Representation visualization of convolutional neural networks: A survey. *Acta Automatica Sinica* 2022, 48(8): 1890–1920.

[14] Röhling T, Mack J, Schulz D. A fast histogram-based similarity measure for detecting loop closures in 3-D LIDAR data. In *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 28 September–2 October 2015, pp. 736–741.

[15] Lin J, Zhang F. A fast, complete, point cloud based loop closure for LiDAR odometry and mapping. *arXiv* 2019, arXiv:2106.10458.

[16] Bosse M, Zlot R. Place recognition using keypoint voting in large 3D lidar datasets. In *2013 IEEE International Conference on Robotics and Automation*, 6–10 May 2013, pp. 2677–2684.

[17] Zlot R, Bosse M. Place recognition using keypoint similarities in 2D lidar maps. In *Experimental Robotics*, Berlin, Heidelberg, 2009, pp. 363–372.

[18] Dubé R, Dugas D, Stumm E, Nieto J, Siegwart R, *et al.* SegMatch: segment based place recognition in 3D point clouds. In *2017 IEEE International Conference on Robotics and Automation (ICRA)*, 29 May–3 June 2017, pp. 5266–5272.

[19] Yin H, Ding X, Tang L, Wang Y, Xiong R. Efficient 3D LIDAR based loop closing using deep neural network. In *2017 IEEE International Conference on Robotics and Biomimetics (ROBIO)*, 5–8 December 2017, pp. 481–486.

[20] Charles RQ, Su H, Kaichun M, Guibas LJ. PointNet: deep learning on point sets for 3D classification and segmentation. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 21–26 July 2017, pp. 77–85.

[21] Uy MA, Lee GH. PointNetVLAD: Deep point cloud based retrieval for large-scale place recognition. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 18–23 June 2018, pp. 4470–4479.

[22] Kim G, Kim A. Scan context: egocentric spatial descriptor for place recognition within 3D point cloud map. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 1–5 October 2018, pp. 4802–4809.

[23] Wang H, Wang C, Xie L. Intensity scan context: coding intensity and geometry relations for loop closure detection. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, 31 May-31 August 2020, pp. 2095–2101.

[24] Wang Y, Sun Z, Xu CZ, Sarma SE, Yang J, *et al.* LiDAR iris for loop-closure detection. In *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 24 October–24 January 2021, pp. 5769–5775.

[25] Liao M, Zhang Y, Zhang J, Liang L, Coleman S, *et al.* Semantic topological descriptor for loop closure detection within 3d point clouds in outdoor environment. In *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 23–27 October 2022, pp. 2856–2863.

[26] Zhang G, Zhang T, Zhao S, Hou L. Binary image fingerprint: stable structure identifier for 3D LiDAR place recognition. *IEEE Robotics and Automation Letters* 2023, 8(9):5648–5655.

[27] Yin H, Xu X, Lu S, Chen X, Xiong R, *et al.* A survey on global LiDAR localization: challenges, advances and open problems. *International Journal of Computer Vision* 2024, 132(8):3139–3171.

[28] Hao W, Zhang W, Liang W, Xiao Z, Jin H. Scene recognition for 3D point clouds： a review. *Optics and Precision Engineering* 2022.

[29] Chen X, Ma H, Wan J, Li B, Xia T. Multi-view 3D Object detection network for autonomous

driving. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 21–26 July 2017, pp. 6526–6534.

[30] Maturana D, Scherer S. VoxNet: A 3D Convolutional Neural Network for real-time object recognition. In *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 28 September–2 October 2015, pp. 922–928.

[31] Dai A, Nießner M. 3DMV: Joint 3D-Multi-view prediction for 3d semantic scene segmentation. In *Computer Vision — ECCV 2018*, Cham, 2018, pp. 458–474.

[32] O'Keefe J, Dostrovsky J. The hippocampus as a spatial map. Preliminary evidence from unit activity in the freely-moving rat. *Brain Res.* 1971, 34(1):171–175.

[33] Chen X, Läbe T, Milioto A, Röhling T, Vysotska O, *et al*. OverlapNet: loop closing for LiDAR-based SLAM. *arXiv* 2021, arXiv:2105.11344.

[34] Chen X, Läbe T, Milioto A, Röhling T, Behley J, *et al*. OverlapNet: a siamese network for computing LiDAR scan similarity with applications to loop closing and localization. *Autonomous Robots* 2022, 46(1):61–81.

[35] Ma J, Zhang J, Xu J, Ai R, Gu W, *et al.* OverlapTransformer: An efficient and yaw-angle-invariant transformer network for LiDAR-Based place recognition. *IEEE Robotics and Automation Letters* 2022, 7(3):6958–6965.

[36] Shu DW, Kwon J. Hierarchical bidirected graph convolutions for large-scale 3-d point cloud place recognition. *IEEE Transactions on Neural Networks and Learning Systems* 2024, 35(7):9651–9662.

[37] Zhang W, Xiao C. PCAN: 3D attention map learning using contextual information for point cloud based retrieval. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 15–20 June 2019, pp. 12428–12437.

[38] Sun Q, Liu H, He J, Fan Z, Du X. DAGC: employing dual attention and graph convolution for point cloud based place recognition. *Proceedings of the 2020 International Conference on Multimedia Retrieval* 2020, 224–232.

[39] Liu Z, Zhou S, Suo C, Yin P, Chen W, *et al.* LPD-Net: 3D point cloud learning for large-scale place recognition and environment analysis. In *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, 27 October–2 November 2019, pp. 2831–2840.

[40] Liu Z, Suo C, Zhou S, Xu F, Wei H, *et al*. SeqLPD: sequence matching enhanced loop-closure detection based on large-scale point cloud description for self-driving vehicles. In *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 3–8 November 2019, pp. 1218–1223.

[41] Poiesi F, Boscaini D. Distinctive 3D local deep descriptors. In *2020 25th International Conference on Pattern Recognition (ICPR)*, 10–15 Jan. 2021, pp. 5720–5727.

[42] Ao S, Hu Q, Yang B, Markham A, Guo Y. SpinNet: learning a general surface descriptor for 3D point cloud registration. In *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 20–25 June 2021, pp. 11748–11757.

[43] Zhou Y, Wang Y, Poiesi F, Qin Q, Wan Y. Loop closure detection using local 3D deep descriptors. *IEEE Robotics and Automation Letters* 2022, 7(3):6335–6342.

[44] Lowe DG. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision* 2004, 60(2):91–110.

[45] Xia Y, Xu Y, Li S, Wang R, Du J, *et al.* SOE-Net: a self-attention and orientation encoding network for point cloud based place recognition. In *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 20–25 June 2021, pp. 11343–11352.

[46] Jiang M, Wu Y, Zhao T, Zhao Z, Lu C. Pointsift: a sift-like network module for 3d point cloud semantic segmentation. *arXiv* 2018, arXiv:1807.00652.

[47] Vaswani A, Shazeer N, Parmar N, Uszkoreit J, Jones L, *et al*. Attention is all you need. *Advances in neural information processing systems* 2017, 30.

[48] Fan Z, Song Z, Zhang W, Liu H, He J, et al. RPR-Net: a point cloud-based rotation-aware large scale place recognition network. In *Computer Vision — ECCV 2022 Workshops*, Cham, 2023, pp.

709–725.

[49] Groh F, Wieschollek P, Lensch HPA. Flex-Convolution. In *Computer Vision — ACCV 2018*, Cham, 2019, pp. 105–122.

[50] Thomas H, Qi CR, Deschaud J, Marcotegui B, Goulette F, *et al*. KPConv: flexible and deformable convolution for point clouds. In *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, 27–2 November 2019, pp. 6410–6419.

[51] Du J, Wang R, Cremers D. DH3D: Deep hierarchical 3D descriptors for robust large-scale 6DoF relocalization. In *Computer Vision – ECCV 2020*, Cham, 2020, pp. 744–762.

[52] Hu J, Shen L, Sun G. Squeeze-and-excitation networks. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 18–23 June 2018, pp. 7132–7141.

[53] Lin CE, Song J, Zhang R, Zhu M, Ghaffari M. SE(3)-Equivariant Point Cloud-Based Place Recognition. *Proceedings of The 6th Conference on Robot Learning* 2023, 2051520–2051530.

[54] Chen H, Liu S, Chen W, Li H, Hill R. Equivariant point network for 3D point cloud analysis. In *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 20–25 June 2021, pp. 14509–14518.

[55] Zhu M, Ghaffari M, Clark WA, Peng H. E2PN: efficient SE(3)-equivariant point network. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 17–24 June 2023, pp. 1223–1232.

[56] Wiesmann L, Nunes L, Behley J, Stachniss C. KPPR: exploiting momentum contrast for point cloud-based place recognition. *IEEE Robotics and Automation Letters* 2023, 8(2):592–599.

[57] Wang Y, Sun Y, Liu Z, Sarma SE, Bronstein MM, *et al.* Dynamic graph CNN for learning on point clouds. *ACM Trans. Graph.* 2019, 38(5): 146.

[58] Hui L, Yang H, Cheng M, Xie J, Yang J. Pyramid point cloud transformer for large-scale place recognition. In *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, 10–17 October 2021, pp. 6078–6087.

[59] Hui L, Cheng M, Xie J, Yang J, Cheng MM. Efficient 3D point cloud feature learning for large-scale place recognition. *IEEE Transactions on Image Processing* 2022, 31:1258–1270.

[60] Ye T, Yan X, Wang S, Li Y, Zhou F. An efficient 3-D point cloud place recognition approach based on feature point extraction and transformer. *IEEE Transactions on Instrumentation and Measurement* 2022, 71:1–9.

[61] Hou Z, Yan Y, Xu C, Kong H. HiTPR: hierarchical transformer for place recognition in point cloud. *2022 International Conference on Robotics and Automation (ICRA)* 2022, 2612–2618.

[62] Wu T, Fu H, Liu B, Xue H, Ren R, *et al.* Detailed analysis on generating the range image for lidar point cloud processing. *Electronics* 2021, 10(11).

[63] Yin H, Tang L, Ding X, Wang Y, Xiong R. LocNet: global localization in 3d point clouds for mobile vehicles. In *2018 IEEE Intelligent Vehicles Symposium (IV)*, 26–30 June 2018, pp. 728–733.

[64] Yin H, Wang Y, Ding X, Tang L, Huang S, *et al.* 3D LiDAR-Based global localization using siamese neural network. *IEEE Transactions on Intelligent Transportation Systems* 2020, 21(4):1380–1392.

[65] Kong D, Li X, Hu Y, Xu Q, Wang A, *et al.* Learning a novel LiDAR submap-based observation model for global positioning in long-term changing environments. *IEEE Transactions on Industrial Electronics* 2023, 70(3):3147–3157.

[66] Schaupp L, Bürki M, Dubé R, Siegwart R, Cadena C. OREOS: oriented recognition of 3d point clouds in outdoor scenarios. In *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 3–8 November 2019, pp. 3255–3261.

[67] Xiang H, Zhu X, Shi W, Fan W, Chen P, *et al.* DeLightLCD: A deep and lightweight network for loop closure detection in LiDAR SLAM. *IEEE Sensors Journal* 2022, 22(21):20761–20772.

[68] Barros T, Garrote L, Pereira R, Premebida C, Nunes UJ. AttDLNet: attention-based deep network for 3D LiDAR place recognition. In *ROBOT2022: Fifth Iberian Robotics Conference*, Cham, 2023, pp. 309–320.

[69] Redmon J, Farhadi A. Yolov3: An incremental improvement. *arXiv* 2018, arXiv:1804.02767.

[70] Ma J, Chen X, Xu J, Xiong G. SeqOT: A spatial — temporal transformer network for place recognition using sequential LiDAR data. *IEEE Transactions on Industrial Electronics* 2023, 70(8):8225–8234.

[71] Yin P, Wang F, Egorov A, Hou J, Zhang J, *et al.* SeqSphereVLAD: sequence matching enhanced orientation-invariant place recognition. In *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 24 October–24 January 2021, pp. 5024–5029.

[72] Yin P, Wang F, Egorov A, Hou J, Jia Z, *et al.* Fast sequence-matching enhanced viewpoint-invariant 3-D place recognition. *IEEE Transactions on Industrial Electronics* 2022, 69(2):2127–2135.

[73] Zhao S, Yin P, Yi G, Scherer S. SphereVLAD++: attention-based and signal-enhanced viewpoint invariant descriptor. *IEEE Robotics and Automation Letters* 2023, 8(1):256–263.

[74] Luo L, Cao SY, Han B, Shen HL, Li J. BVMatch: lidar-based place recognition using bird's-eye view images. *IEEE Robotics and Automation Letters* 2021, 6(3):6076–6083.

[75] Ma J, Xiong G, Xu J, Chen X. CVTNet: A cross-view transformer network for lidar-based place recognition in autonomous driving environments. *IEEE Transactions on Industrial Informatics* 2024, 20(3):4039–4048.

[76] Kim G, Park B, Kim A. 1-Day learning, 1-Year localization: long-term LiDAR localization using scan context image. *IEEE Robotics and Automation Letters* 2019, 4(2):1948–1955.

[77] Luo L, Zheng S, Li Y, Fan Y, Yu B, *et al.* BEVPlace: Learning LiDAR-based Place Recognition using Bird's Eye View Images. In *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, 1–6 October 2023, pp. 8666–8675.

[78] Yin P, Xu L, Liu Z, Li L, Salman H, *et al.* Stabilize an unsupervised feature learning for LiDAR-based place recognition. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 1–5 October 2018, pp. 1162–1167.

[79] Xu X, Yin H, Chen Z, Li Y, Wang Y, *et al.* DiSCO: differentiable scan context with orientation. *IEEE Robotics and Automation Letters* 2021, 6(2):2791–2798.

[80] Cao F, Yan F, Wang S, Zhuang Y, Wang W. Season-invariant and viewpoint-tolerant LiDAR place recognition in gps-denied environments. *IEEE Transactions on Industrial Electronics* 2021, 68(1):563–574.

[81] Cao F, Wu H, Wu C. An end-to-end localizer for long-term topological localization in large-scale changing environments. *IEEE Transactions on Industrial Electronics* 2023, 70(5):5140–5149.

[82] Lu S, Xu X, Yin H, Chen Z, Xiong R, *et al.* One RING to rule them all: radon sinogram for place recognition, orientation and translation estimation. In *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 23–27 October 2022, pp. 2778–2785.

[83] Xu X, Lu S, Wu J, Lu H, Zhu Q, *et al.* RING++: Roto-translation invariant gram for global localization on a sparse scan map. *IEEE Transactions on Robotics* 2023, 39(6):4616–4635.

[84] Lu S, Xu X, Tang L, Xiong R, Wang Y. DeepRING: learning roto-translation invariant representation for LiDAR based place recognition. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, 29 May–2 June 2023, pp. 1904–1911.

[85] Yin P, Xu L, Zhang J, Choset H. FusionVLAD: A multi-view deep fusion networks for viewpoint-free 3D place recognition. *IEEE Robotics and Automation Letters* 2021, 6(2):2304–2310.

[86] Simonyan K, Zisserman A. Very deep convolutional networks for large-scale image recognition. *ArXiv* 2015, arXiv:1409.1556.

[87] Fan Z, Song Z, Liu H, Lu Z, He J, *et al.* SVT-Net: Super light-weight sparse voxel transformer for large scale place recognition. *Proceedings of the AAAI Conference on Artificial Intelligence* 2022, 36(1):551–560.

[88] Siva S, Nahman Z, Zhang H. Voxel-based representation learning for place recognition based on 3D point clouds. In *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 24 October–24 January 2021, pp. 8351–8357.

[89] Choy C, Gwak J, Savarese S. 4D spatio-temporal convnets: minkowski convolutional neural networks. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*,

15–20 June 2019, pp. 3070–3079.

[90] Tang H, Liu Z, Zhao S, Lin Y, Lin J, *et al*. Searching efficient 3D architectures with sparse point-voxel convolution. *Computer Vision — ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXVIII* 2020, 685–702.

[91] Komorowski J. MinkLoc3D: Point cloud based large-scale place recognition. In *2021 IEEE Winter Conference on Applications of Computer Vision (WACV)*, 3–8 January 2021, pp. 1789–1798.

[92] Komorowski J. Improving point cloud based place recognition with ranking-based loss and large batch training. In *2022 26th International Conference on Pattern Recognition (ICPR)*, 21–25 August 2022, pp. 3699–3705.

[93] Xu T, Guo Y, Li Z, Yu G, Lai Y, *et al*. TransLoc3D: Point cloud based large-scale place recognition using adaptive receptive fields. *arXiv* 2022, arXiv:2105.11605

[94] Cattaneo D, Vaghi M, Valada A. LCDNet: deep loop closure detection and point cloud registration for LiDAR SLAM. *IEEE Transactions on Robotics* 2022, 38(4):2074–2093.

[95] Shi S, Guo C, Jiang L, Wang Z, Shi J, *et al*. PV-RCNN: point-voxel feature set abstraction for 3D object detection. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 13–19 June 2020, pp. 10526–10535.

[96] Vidanapathirana K, Ramezani M, Moghadam P, Sridharan S, Fookes C. LoGG3D-Net: locally guided global descriptor learning for 3D place recognition. In *2022 International Conference on Robotics and Automation (ICRA)*, 23–27 May 2022, pp. 2215–2221.

[97] Wang Q, Wu B, Zhu P, Li P, Zuo W, *et al*. ECA-Net: efficient channel attention for deep convolutional neural networks. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 13–19 June 2020, pp. 11531–11539.

[98] Guo MH, Liu ZN, Mu TJ, Hu SM. Beyond self-attention: external attention using two linear layers for visual tasks. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 2023, 45(5):5436–5447.

[99] Chang MY, Yeon S, Ryu S, Lee D. SpoxelNet: spherical voxel-based deep place recognition for 3D point clouds of crowded indoor spaces. In *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 24 October–24 January 2021, pp. 8564–8570.

[100] Żywanowski K, Banaszczyk A, Nowicki MR, Komorowski J. MinkLoc3D-SI: 3D LiDAR place recognition with sparse convolutions, spherical coordinates, and intensity. *IEEE Robotics and Automation Letters* 2022, 7(2):1079–1086.

[101] Zhou Z, Zhao C, Adolfsson D, Su S, Gao Y, *et al*. NDT-transformer: large-scale 3D point cloud localisation using the normal distribution transform representation. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, 30 May–5 June 2021, pp. 5654–5660.

[102] Zhou Y, Sun P, Zhang Y, Anguelov D, Gao J, *et al*. End-to-end multi-view fusion for 3d object detection in lidar point clouds. In *Conference on Robot Learning. PMLR*, 2020: 923–932.

[103] Liang J, Son S, Lin M, Manocha D. GeoLCR: attention-based geometric loop closure and registration. *arXiv* 2023, arXiv:2302.13509.

[104] Qin Z, Yu H, Wang C, Guo Y, Peng Y, *et al*. GeoTransformer: fast and robust point cloud registration with geometric transformer. *IEEE Trans. Pattern Anal. Mach. Intell.* 2023, 45(8):9806–9821.

[105] Kong D, Li X, Hu W, Hu J, Hu Y, *et al*. Explicit points-of-interest driven siamese transformer for 3D LiDAR place recognition in outdoor challenging environments. *IEEE Transactions on Industrial Informatics* 2023, 19(10):10564–10577.

[106] Shan T, Englot B. LeGO-LOAM: lightweight and ground-optimized lidar odometry and mapping on variable terrain. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 1–5 October 2018, pp. 4758–4765.

[107] Yin P, Xu L, Feng Z, Egorov A, Li B. PSE-Match: A viewpoint-free place recognition method with parallel semantic embedding. *IEEE Transactions on Intelligent Transportation Systems* 2022, 23(8):11249–11260.

[108] Zaganidis A, Magnusson M, Duckett T, Cielniak G. Semantic-assisted 3D normal distributions transform for scan registration in environments with limited structure. In *2017 IEEE/RSJ*

International Conference on Intelligent Robots and Systems (IROS), 24–28 September 2017, pp. 4064–4069.

[109] Zaganidis A, Sun L, Duckett T, Cielniak G. Integrating deep semantic segmentation into 3-D point cloud registration. *IEEE Robotics and Automation Letters* 2018, 3(4):2942–2949.

[110] Zaganidis A, Zerntev A, Duckett T, Cielniak G. Semantically assisted loop closure in SLAM using NDT histograms. In *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 3–8 November 2019, pp. 4562–4568.

[111] Zhu Y, Ma Y, Chen L, Liu C, Ye M, *et al*. GOSMatch: graph-of-semantics matching for detecting loop closures in 3D LiDAR data. In *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 24 October–24 January 2021, pp. 5151–5157.

[112] Li L, Kong X, Zhao X, Huang T, Li W, *et al.* SSC: semantic scan context for large-scale place recognition. In *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 27 September–1 October 2021, pp. 2092–2099.

[113] Pramatarov G, Martini DD, Gadd M, Newman P. BoxGraph: semantic place recognition and pose estimation from 3D LiDAR. In *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 23–27 October 2022, pp. 7004–7011.

[114] Li L, Kong X, Zhao X, Huang T, Liu Y. Semantic scan context: a novel semantic-based loop-closure method for LiDAR SLAM. *Autonomous Robots* 2022, 46(4):535–551.

[115] Li L, Kong X, Zhao X, Li W, Wen F, *et al.* SA-LOAM: semantic-aided LiDAR SLAM with loop closure. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, 30 May–5 June 2021, pp. 7627–7634.

[116] Jin S, Wu Z, Zhao C, Zhang J, Peng G, *et al*. SectionKey: 3-D semantic point cloud descriptor for place recognition. In *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 23–27 October 2022, pp. 9905–9910.

[117] Cramariuc A, Dubé R, Sommer H, Siegwart R, Gilitschenski I. Learning 3D segment descriptors for place recognition. *arXiv* 2018, arXiv:1804.09270.

[118] Dubé R, Cramariuc A, Dugas D, Sommer H, Dymczyk M, *et al*. SegMap: segment-based mapping and localization using data-driven descriptors. *Int. J. Rob. Res.* 2020, 39(2–3):339–355.

[119] Dubé R, Cramariuc A, Dugas D, Nieto JI, Siegwart R, *et al*. SegMap: 3D Segment Mapping using Data-Driven Descriptors. *CoRR* 2018, abs:1804.09557.

[120] Wietrzykowski J, Skrzypczyński P. On the descriptive power of LiDAR intensity images for segment-based loop closing in 3-D SLAM. In *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 27 September–1 October 2021, pp. 79–85.

[121] Vidanapathirana K, Moghadam P, Harwood B, Zhao M, Sridharan S, *et al*. Locus: LiDAR-based place recognition using spatiotemporal higher-order pooling. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, 30 May–5 June 2021, pp. 5075–5081.

[122] Li L, Kong X, Zhao X, Huang T, Li W, *et al*. RINet: Efficient 3D Lidar-Based place recognition using rotation invariant neural network. *IEEE Robotics and Automation Letters* 2022, 7(2):4321–4328.

[123] Arce J, Vödisch N, Cattaneo D, Burgard W, Valada A. PADLoC: LiDAR-based deep loop closure detection and registration using panoptic attention. *IEEE Robotics and Automation Letters* 2023, 8(3):1319–1326.

[124] Kong X, Yang X, Zhai G, Zhao X, Zeng X, *et al*. Semantic graph based place recognition for 3D point clouds. In *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 24 October–24 January 2021, pp. 8216–8223.

[125] Milioto A, Vizzo I, Behley J, Stachniss C. RangeNet ++: Fast and accurate LiDAR semantic segmentation. In *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 3–8 November 2019, pp. 4213–4220.

[126] Dai D, Wang J, Chen Z, Bao P. SC-LPR: Spatiotemporal context based LiDAR place recognition. *Pattern Recognition Letters* 2022, 156:160–166.

[127] Khan K, Rehman SU, Aziz K, Fong S, Sarasvady S. DBSCAN: Past, present and future. In *The*

Fifth International Conference on the Applications of Digital Information and Web Technologies (ICADIWT 2014), 17–19 February 2014, pp. 232–238.

[128] Pandey G, Mcbride JR, Eustice RM. Ford campus vision and lidar data set. *Int. J. Rob. Res.* 2011, 30(13):1543–1552.

[129] Geiger A, Lenz P, Urtasun R. Are we ready for autonomous driving? The KITTI vision benchmark suite. In *2012 IEEE Conference on Computer Vision and Pattern Recognition*, 16–21 June 2012, pp. 3354–3361.

[130] Carlevaris-Bianco N, Ushani AK, Eustice RM. University of michigan north campus long-term vision and lidar dataset. *The International Journal of Robotics Research* 2015, 35(9):1023–1035.

[131] Maddern W, Pascoe G, Linegar C, Newman P. 1 year, 1000 km: The oxford robotcar dataset. *Int. J. Rob. Res.* 2017, 36(1):3–15.

[132] Behley J, Garbade M, Milioto A, Quenzel J, Behnke S, *et al*. SemanticKITTI: A dataset for semantic scene understanding of LiDAR sequences. In *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, 27 October–2 November 2019, pp. 9296–9306.

[133] Kim G, Park YS, Cho Y, Jeong J, Kim A. MulRan: multimodal range dataset for urban place recognition. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, 31 May–31 August 2020, pp. 6246–6253.

[134] Liao Y, Xie J, Geiger A. KITTI-360: A novel dataset and benchmarks for urban scene understanding in 2D and 3D. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 2023, 45(3):3292–3310.

[135] Knights J, Vidanapathirana K, Ramezani M, Sridharan S, Fookes C, *et al.* Wild-places: a large-scale dataset for lidar place recognition in unstructured natural environments. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, 29 May–2 June 2023, pp. 11322–11328.

[136] Ferrarini B, Waheed M, Waheed S, Ehsan S, Milford MJ, *et al*. Exploring performance bounds of visual place recognition using extended precision. *IEEE Robotics and Automation Letters* 2020, 5(2):1688–1695.

[137] Li Q, Yu X, Queralta JP, Westerlund T. Multi-modal lidar dataset for benchmarking general-purpose localization and mapping algorithms. In *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 23–27 October 2022, pp. 3837–3844.

[138] Yu X, Zhou B, Chang Z, Qian K, Fang F. MMDF: multi-modal deep feature based place recognition of mobile robots with applications on cross-scene navigation. *IEEE Robotics and Automation Letters* 2022, 7(3):6742–6749.

[139] Knights J, Moghadam P, Ramezani M, Sridharan S, Fookes C. InCloud: incremental learning for point cloud place recognition. In *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 23–27 October 2022, pp. 8559–8566.

[140] Qiao Z, Hu H, Shi W, Chen S, Liu Z, *et al.* A registration-aided domain adaptation network for 3D point cloud based place recognition. In *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 27 September–1 October 2021, pp. 1317–1322.

[141] Yin H, Wang Y, Tang L, Xiong R. Radar-on-lidar: metric radar localization on prior lidar maps. In *2020 IEEE International Conference on Real-time Computing and Robotics (RCAR)*, 28–29 September 2020, pp. 1–7.

[142] Yin H, Wang Y, Wu J, Xiong R. Radar style transfer for metric robot localisation on lidar maps. *CAAI Transactions on Intelligence Technology* 2023, 8(1):139–148.

[143] Fu C, Li L, Peng L, et al. OverlapNetVLAD: A Coarse-to-Fine Framework for LiDAR-based Place Recognition. *arXiv* 2023, arXiv:2303.06881.

[144] Komorowski J, Wysoczańska M, Trzcinski T. MinkLoc++: Lidar and monocular image fusion for place recognition. In *2021 International Joint Conference on Neural Networks (IJCNN)*, 18–22 July 2021, pp. 1–8.

[145] Yin H, Xu X, Wang Y, Xiong R. Radar-to-lidar: heterogeneous place recognition via joint

learning. *Frontiers in Robotics and AI* 2021, 8: 661199.

[146] Lai H, Yin P, Scherer S. AdaFusion: Visual-LiDAR fusion with adaptive weights for place recognition. *IEEE Robotics and Automation Letters* 2022, 7(4):12038–12045.

[147] Xu R, Luo F, Wang C, Chang B, Huang J, *et al*. From dense to sparse: contrastive pruning for better pre-trained language model compression. *Proceedings of the AAAI Conference on Artificial Intelligence* 2022, 36(10):11547–11555.

[148] Cho H, Choi J, Baek G, Hwang W. itKD: Interchange transfer-based knowledge distillation for 3D object detection. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 17–24 June 2023, pp. 13540–13549.

[149] Wiesmann L, Marcuzzi R, Stachniss C, et al. Retriever: Point cloud retrieval in compressed 3D maps. 2022 International Conference on Robotics and Automation (ICRA). IEEE, 2022: 10925–10932

[150] Arandjelović R, Gronat P, Torii A, Pajdla T, Sivic J. NetVLAD: CNN architecture for weakly supervised place recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 2018, 40(6):1437–1451.

[151] Tsintotas KA, Bampis L, Gasteratos A. The revisiting problem in simultaneous localization and mapping: a survey on visual loop closure detection. *IEEE Transactions on Intelligent Transportation Systems* 2022, 23(11):19929–19953.