GNN-PR: 3D Point Cloud Place Recognition Based on Graph Neural Network

lwl2021 Liu[1], Jiajun Fei[1], and Ziyu Zhu[2]

[1]Tsinghua University
[2]Department of Computer Science, Tsinghua University

September 19, 2022

Abstract

Place recognition technology is very important for autono-mous driving. To realize the large-scale recognition task of 3D point clouds, we propose a large-scale 3D point cloud place recognition framework based on graph neural networks, which combines local and global features. In extracting features, instance segmentation is performed on the large scene point clouds first, and then the GNN network trains each segmented instance to obtain local attribute features. We construct a graph model with each object as a node and the relationship between them as edges, then obtain the global topological structure features of the scene. In calculating similar scores, we calculate the similarity vector of the global and local feature through a similarity network and cosine similarity, respectively. Finally, we fuse the similarity vectors and calculate the final similarity score. This paper uses the SemanticKitti and nuScenes datasets to verify the proposed method. Compared with the state-of-the-art deep learning-based place recognition method, the proposed method achieves the best results in the SemanticKitti and nuScenes datasets.

GNN-PR: 3D Point Cloud Place Recognition Based on Graph Neural Network

Wenlei Liu, Jiajun Fei and Ziyu Zhu

Place recognition technology is very important for autonomous driving. To realize the large-scale recognition task of 3D point clouds, we propose a large-scale 3D point cloud place recognition method based on graph neural networks, which combines local and global features. In extracting features, instance segmentation is performed on the large scene point clouds first, and then the GNN trains each segmented instance to obtain local attribute features. We construct a graph model with each object as a node and the relationship between them as edges, then get the global topological structure features of the scene. In calculating similar scores, we calculate the similarity vector of the global and local feature through a similarity network and cosine similarity, respectively. Finally, we fuse the similarity vectors and calculate the final similarity score. This paper uses the SemanticKitti and nuScenes datasets to verify the proposed method. Compared with the state-of-the-art deep learning-based place recognition method, the proposed method achieves the best results in the SemanticKitti and nuScenes datasets.

Key words: place recognition, graph neural network, similarity features, feature fusion

Introduction

With the development of technology, the application of autonomous driving in our life is more extensive, such as unmanned postal delivery, truck transportation, and geological surveying. High-precision positioning is the main cornerstone of autonomous driving, which provides reliable and accurate positioning of self-driving vehicles. The place recognition technology can eliminate accumulated errors in autonomous driving, which is essential for this task. Usually, the pose information of two frames of data is used to calculate the Euclidean distance, and the distance determines whether there is recognition, so the place recognition problem is transformed into a binary classification problem. Image-based methods are currently one of the most effective recognition methods. However, due to changes in sunlight, weather, seasons, viewing angle, and structure, images of a vehicle's surroundings can vary widely, affecting recognition accuracy and even resulting in recognition failure Barros et al. (2021b). Because the geometric information of 3D point clouds is less sensitive to lighting, seasonal, and structural changes, using 3D point clouds for place recognition has better robustness.

Nowadays, there is much more literature on deep learning-based place relocation methods. PointNetVLAD Uy and Lee (2018), which combines PointNet Qi et al. (2017) and NetVLAD Arandjelovic et al. (2016), is the first network model for place recognition. PointNet extracts the features of each point cloud and NetVLAD clusters and classifies point cloud features, to realize end-to-end training and inference. However, it lacks local features and the ability to describe point

Wenlei Liu is a postdoctoral fellow at the Institute of Artificial Intelligence, Department of Computer Science, Tsinghua University. Jiajun Fei and Ziyu Zhu is a doctoral student at the Department of Artificial Intelligence, Department of Computer Science, Tsinghua University.

AI Magazine. 00(000): 1–12; doi:10.1002/aaai.0001
The Authors. Association for the Advancement of Artificial Intelligence.

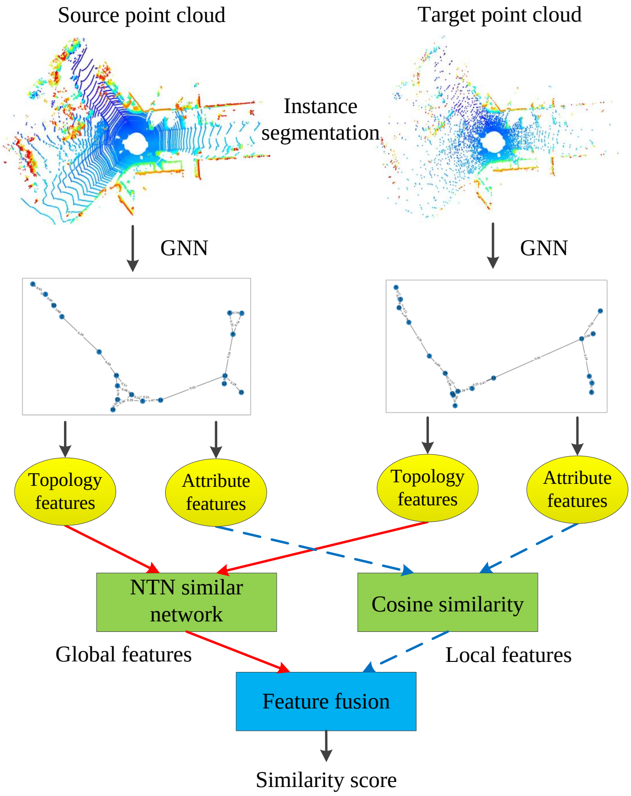


Figure 1. The architecture of the GNN-PR. We utilize a method based on GNN of local and global features fusion for place recognition.

features. LPD-net Liu et al. (2019) uses a combination of adaptive local features and graph-based neighborhood aggregation, which uses DGCNN Wang et al. (2019) to dynamically learn the feature and geometric space, and finally uses NetVLAD for feature aggregation to obtain a global descriptor. Although this method improves the discrimination and generalization ability, the number of model parameters is relatively large. SG-PR Kong et al. (2020) proposes a strategy based on semantic graph recognition that uses the semantic and topological information of the point clouds. Inspired by the above methods, this paper combines global and local features to achieve recognition.

To solve the problem of precise recognition in large scenes, we use the powerful representation learning capabilities of GNN Veličković et al. (2017) Wang, Cui, and Zhu (2016) to learn the abstract features of point cloud data and realize efficient recognition. Graph model includes two types of characteristics: attribute features and structural features. Attribute features describe the inherent attributes of the graph, which are the local features of the object in the scene. Structural features describe the nature of the association with each other, which are the relationship between objects in the scene, is the global features. This paper fuses the local and global features to relocate in large scenes. The architecture of the GNN-PR shows in Figure 1. The source and target point cloud data are first segmented, and the GNN model then trains each instance segmentation object to obtain the attribute features. The graph model takes each instance segmentation object as a node and their relationship as edges to get the graph structural features. NTN Bai et al. (2019) similarity networks calculate the similarity vector of global features, and the similarity vector of local features is calculated by cosine similarity. Finally, the final recognition score is calculated by the fusion method. The best results of place recognition are achieved in the SemanticKitti and nuScenes datasets. The main contributions of this paper are the following:

- We propose an innovative 3D point cloud large-scale place recognition method based on GNN.
- In calculating similarity scores, we fuse the similarity vectors calculated by local and global features, improving the matching accuracy.
- The global features fuse topology features, geometric features, and histogram features to enhance the global expression ability of features.
- Experiments on the Semankitti and nuScenes datasets achieve the best results.

Related Work

Graph Model

Graphs represent the structural relationship between different objects and are widely used in autonomous driving. PointGNN Shi and Rajkumar (2020) encodes the point cloud into a neighboring graph with a fixed radius, GCN predicts the type and shape of the object that each vertex belongs to in the graph, and detects multiple targets. Struc2vec Ribeiro, Saverese, and Figueiredo (2017) is one method of learning the potential representation of the nodes' structural features, which independently assesses the similarity between nodes and the consistency of edges. The structural similarity between nodes only depends on their degree, while the hierarchical similarity depends on the whole network. SAGPool Lee, Lee, and Kang (2019) is a graph pooling method based on self-attention. It uses the self-attention mechanism to distinguish points that should be deleted or kept, achieving high efficiency. Wang, Ni, and Yang (2020) Li et al. (2021) use the GNN model to achieve bone-based action recognition.

Matching Method

Matching issues are crucial in recognition. 3DFeat-Net Yew and Lee (2018) is a pioneering work in 3D matching that learns 3D local feature detection and descriptors. However, this method directly knows the attention graph from the input points and only pays attention to the structural information of the local cluster. DH3D Du, Wang, and Cremers (2020) proposes a unified global position recognition and local 6DoF pose solution to overcome the above problems. The Siamese network learns 3D local descriptions from the original 3D points. It integrates FlexConv and Squeeze-and-Excitation (SE) to ensure that the local learned descriptors capture multi-level geometric information and channel correlation. StickyPillars Fischer et al. (2021) is a 3D feature matching method based on end-to-end training of graph neural network, which uses a multi-head attention mechanism and crosses attention to achieve context aggregation. ReAgent Bauer, Patten, and Vincze (2021) is a registration method based on reinforcement learning. A discrete registration strategy of imitation learning is adopted based on a stable expert strategy.

Place Recognition

The point-to-point recognition calculation is too computationally expensive. Now the global descriptor-based recognition method is the primary method. M2DP He, Wang, and Zhang (2016) projects 3D points onto a series of 2D planes with different viewpoints. By describing the projection of the spatial density distribution of a point on a plane, many density distributions of a single cloud can be obtained. Scan Context Kim and Kim (2018) proposes a global descriptor based on a non-histogram. First, a single 3D scan's point cloud is encoded into the context. The N_r (number of rings) dimensional vector encoded from the scanning context. Finally, the retrieved candidates are compared with the query scanning context. The candidate that meets the acceptance threshold and is closest to the query is considered a loop. EPC-Net Hui et al. (2021) is an efficient method for extracting global descriptors. It contains two subnetworks: proxy point convolutional neural network (PPCNN) and grouped VLAD network (G-VALD). The PPCNN is mainly extract multi-scale local geometric features, while the G-VLAD is mainly generate discriminative global descriptors from the acquired multi-scale local geometric features. AttDL-Net Barros et al. (2021a) and PCAN Zhang and Xiao (2019) apply the attention mechanism to recognition to improve the accuracy of position recognition.

Graph Neural Network for Place Recognition (GNN-PR)

This section introduces the 3D point cloud large-scale place recognition method based on graph neural networks. The main steps include data preprocessing, GNN model training, similarity feature and score calculation. The data preprocessing unit performs instance segmentation on point clouds and filters out certain noises and irrelevant objects for relocation. GNN model training mainly uses instance segmentation objects to obtain local attributes and global structural features. Similar features and score calculation fuse local and global similarity vectors and calculate similarity scores.

This paper adopts two-stage training. The first stage is mainly to train the GNN model. The features and super nodes obtained in this stage are the basis for the second stage of training. The second stage fuses global and local features to calculate similarity scores and then determines whether there is relocation.

Data Preprocessing

The primary purpose of data preprocessing is to perform instance segmentation on 3D point cloud data of large scenes. Firstly, the relatively mature point cloud semantic segmentation method obtains the semantic features. Then the clustering method is used to cluster each semantic in-formation to complete the instance segmentation of the entire scene.

In semantic segmentation, we reference the PolarSeg Zhang et al. (2020) method. First, polar coordinates are used to quantify point cloud data into grids, and then U-net structure is used to obtain point cloud semantic features and labels. This method has achieved good results in the semantic segmentation experiments of the SemanticKitti and nuScenes datasets, therefore, we use the semantic segmentation results for instance segmentation.

The semantic segmentation results contain point cloud data of the same kind but belonging to different objects, and instance segmentation needs to separate all objects. The instance segmentation method is: according to the number of instances included in each semantic segmentation, use the K-means++ method to cluster and obtain different instance objects with the same semantics.

The First Stage: GNN Model Training

Each object in the instance segmentation is used as the input of the GNN training model. Each instance cluster contains instance data, semantic labels, and instance labels. The graph model is automatically constructed according to each instance data. In building the GNN model, each point cloud is taken as a node and used to create edges within a certain distance threshold. We select points within 0.5m from the point cloud to construct the edge and use the reciprocal distance between the two points as the weight. In other words, the closer the space is, the higher the weight is, the farther the distance is, and the lower the weight is. The constructed graph model is then used as the input to the training network.

The threshold selection method of the construction edge is usually selected according to the specific scene. A significant threshold can be chosen for a large open scene, and a small one can be selected for a dense scene. For example, in the calculation of GNN features, the distance between points in the same object is relatively close, and the threshold of the edge is selected to be small.

The structure of the GNN training model is shown in Figure 2. We use the graph collapse network Ying et al. (2018) structure to combine the graph collapse process and GNN with learning layer-level tasks to obtain GNN attribute features. We denote the graph is represented by $G = (A, F)$, where A is the adjacency matrix, F is the feature matrix of each node, G contains k subgraphs $\{G_k\}_{k=1}^{K}$, and N_k represents the number of nodes included in the subgraph G_k. G_1, G_2, G_3, and G_4 represent 4 subgraphs in Figure 2.

The network training process involves first learning the features of each node through one GNN model and then learning the probability distribution of each cluster for each node through another GNN model. This process is expressed as follows:

$$Z^l = f_{conv}\left(A^l, H^l\right) \tag{1}$$

$$S^l = Softmax\left(f_{pool}\left(A^l, H^l\right)\right) \tag{2}$$

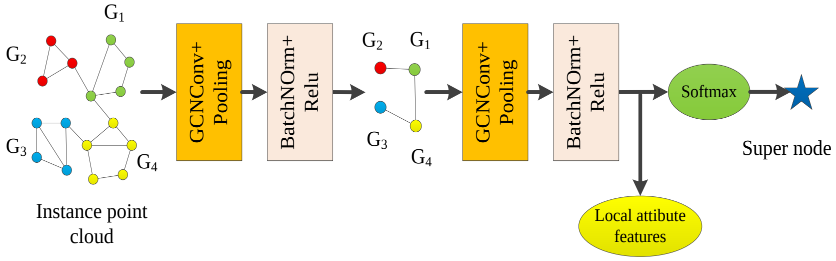


Figure 2. The method of obtaining instance attribute features is based on GNN.

Where Z is the new node embedding, H is the original node embedding, H_0 is F, the adjacency matrix is $A^l \in R^{n^l \times n^l}$, the cluster allocation matrix is $S^l \in R^{n^l \times n^{l+1}}$, n^l represents the number of nodes in layer l, n^{l+1} represents the number of nodes in layer $l+1$, the value of S^l represents the probability that the node is allocated to any cluster, and the supernode in the next layer is the fully connected structure of all nodes in the previous layer. f_{conv} and f_{pool} are two independent GNN with the same input but different parameters. f_{conv} denotes the calculation of new node embedding vectors, and f_{pool} represents the calculation of allocation parameters.

According to the new node embedding Z and cluster as-signment matrix S, the graph model can be collapsed:

$$H^{l+1} = S^{l^T} Z^l \tag{3}$$

$$A^{l+1} = S^{l^T} Z^l S^l \tag{4}$$

The above two formulas are the graph collapse models, which implement the recursive transformation of $\left(A^l, Z^l\right) \rightarrow \left(A^{l+1}, Z^{l+1}\right)$. Eq.(3) is the fusion operation on the information within the cluster, and Eq.(4) is the calculation of the adjacency matrix between clusters. Figure 2 shows the collapsing and fusing point cloud data into a new supernode. The last layer of the cluster allocation matrix is usually 1×1, since the graph model of the entire object needs to be collapsed into a supernode. In other words, a point represents the attribute features of an object.

The loss function mainly includes two components in the training process: the cross-entropy loss function and the connection prediction loss.

$$L_{loss} = L_{cross} + L_{link} \tag{5}$$

Where L_{cross} is the cross-entropy loss function of target prediction and $L_{link} = \|A_i, S_l S_l^T\|$ is the connection prediction loss. Because adjacent points usually are gathered together, the error between the adjacency matrix calculated by the cluster allocation matrix and the adjacency matrix of the next layer should be minimized as much as possible.

The Second Stage: Similarity Features and Score Calculation

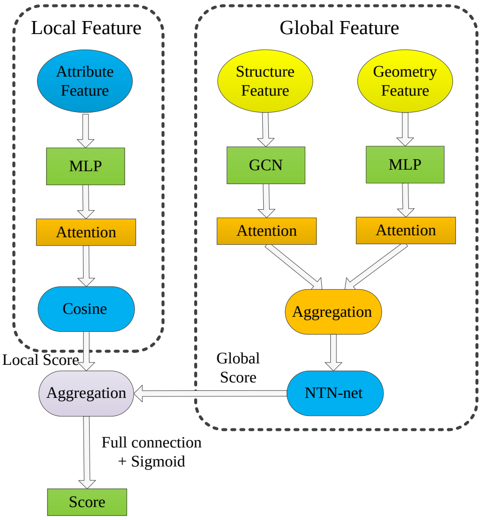


Figure 3. Similar feature computation network structure. We first fuse structural and geometric features to obtain global features and then merge local features.

The network structure of the similar feature calculation shows in Figure 3. In a similar network feature calculation, three features are mainly involved: GNN attribute features, GNN structural features, and geometric features. The method of fusing local and global features, on the other hand, calculates similarity scores.

Topology features. Through the graph collapse method, a supernode represents an instance segmentation object, which describes the attribute features of the object. At the same time, the center of each object, which is calculated by using the point clouds contained in the instance segmentation cluster, is the 3D position information of the supernode. The relationship between objects in the large scene represents the structural features. Constructing structural features involves taking each supernode as the node and the connection relationship between them as the edge, setting the threshold of building edge to 10m, and finally obtaining the topology information of the entire scene.

Geometric features. Geometric features make full use of the spatial geometric information of supernodes. First, normalize all the supernodes in the scene, and then use the (x, y, z) coordinate difference of the super nodes to construct a covariance matrix $M_{m\times 3}$, where m is the number of supernodes, and then find the eigenvalues and eigenvectors of the covariance matrix, and the resulting eigenvalues are used to construct the scene geometry for each frame Liu et al. (2019). The three eigenvalues calculated by the symmetric positive definite matrix respectively satisfy: $\lambda_1^i \geq \lambda_2^i \geq \lambda_3^i$.

A 10-dimensional vector represents the structural feature of these supernodes, including F_{3D} features: curvature change: $C_i = \lambda_3^i / \sum_{j=1}^{3} \lambda_j^i$, total variance: $O_i = \sqrt[3]{\prod_{j=1}^{3} \lambda_j^i} / \sum_{j=1}^{3} \lambda_j^i$, linearity: $\left(\lambda_1^i - \lambda_2^i\right)/\lambda_1^i$, eigenvalue entropy: $A_i = -\sum_{j=1}^{3}\left(\lambda_j^i \ln \lambda_j^i\right)$, feature density: $D_i = 3N/4\prod_{j=1}^{3} \lambda_j^i$, N is the number of supernodes. F_{2D} features: scattering: $S_{i,2D} = \lambda_{2D,1}^i + \lambda_{2D,2}^i$, linearity: $L_{i,2D} = \lambda_{2D,2}^i / \lambda_{2D,1}^i$. F_V feature: the vertical component of the direction vector. F_Z features: maximum height difference and maximum variance. After the 10-dimensional feature passes through the MLP and Attention network, obtaining the global geometric feature.

Global feature fusion method. As shown in Figure 3, the global feature is a fusion of geometric features and GNN structural features. The topology features are obtained by the GNN structural features, including two parts: 1. The degree information of each node; 2. The clustering coefficient of each node, or the number of edges between nodes in a cluster, is divided by the number of possible edges between them.

The geometric features and topology information are fused together to obtain global features, and then the global similarity score is calculated by NTN-net. The NTN network is shown in Figure 4, which is the interaction between graphs and includes calculating graph interactive feature and histogram feature. We assume that the feature vectors are u_1 and u_2. Therefore, the histogram features are:

$$F_{hist} = hist\left(\sigma\left(u_1 u_2^T\right)\right) \tag{6}$$

Where $hist()$ is a normalized histogram function and returns statistical feature information, σ is a nonlinear activation function.

The interactive feature information of the graph is:

$$F_{inter} = \sigma\left(u_1^T W_1 u_2 + W_2 \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} + b\right) \tag{7}$$

Where W_1, W_2 are the learned parameters, and b is the deviation vector.

F_{hist} and F_{inter} are then fused and passed through a fully connected layer to obtain a global matching similarity score.

Similarity score calculation. The similarity score calculation is mainly obtained by fusing the global and local similarity scores. Attention mechanisms can strengthen important information and obtain iconic features. The global similarity score obtains by using a similar network to calculate the fused global features. In contrast, the local feature similarity score f_{score} obtains by using the cosine similarity to calculate the similarity of the two attribute features.

Because the relocation problem can be transformed into a binary classification problem, the cross-entropy loss function is

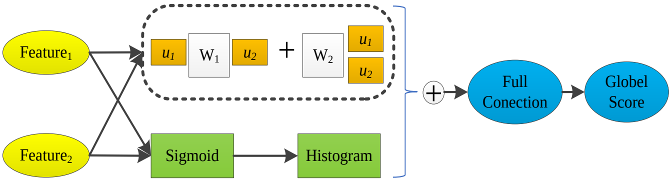


Figure 4. NTN network structure.

used, and its calculation formula is as follows:

$$L_{score} = -\frac{1}{N}\sum_{i=1}^{N}[f^i_{label}\cdot\log(f^i_{score}) + (1-f^i_{label})\cdot\log(1-f^i_{score}) \tag{8}$$

where, f^i_{label} is the ith pair of recognition test label data, f^i_{score} is the similarity score, if it is recognition, it is 1, otherwise it is 0.

Experiment

Experimental setup

This paper uses the SemanticKitti and nuScenes dataset to verify the proposed method and compares it with the most advanced methods. The SemanticKitti dataset is a large-scale dataset based on the 64-line LIDAR of automobiles, containing 22 scenes and annotating 19 types of objects. We selected 12 main types used for experiments. The nuScenes dataset is a large-scale dataset based on automotive 32-line LiDAR, annotating 23 types of objects. We select 10 scenes and 9 main classes for experiments. These datasets are widely used in automatic driving algorithm verification, and their sensor layout and collected data are also consistent with the real automatic driving scene.

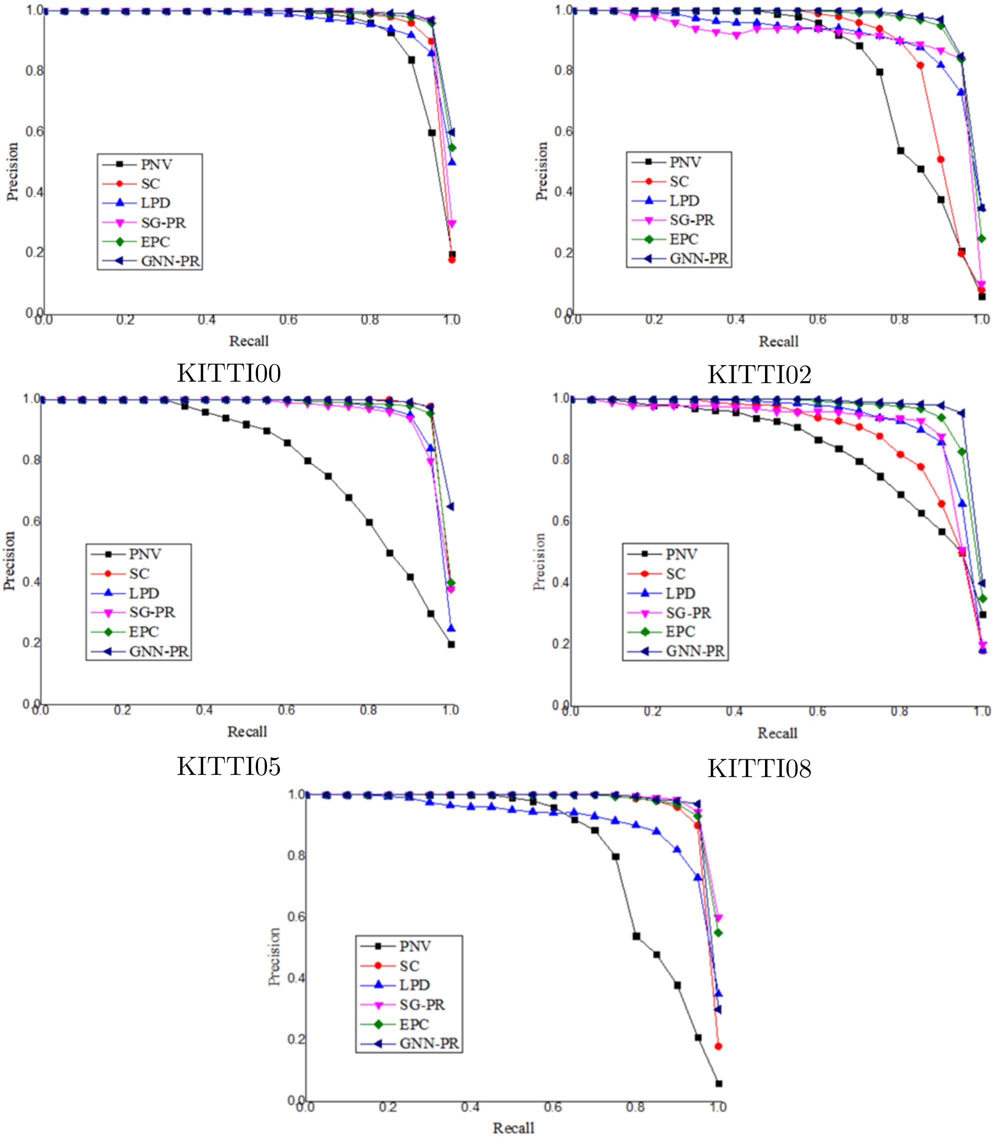
Evaluation method: in the experiment, we use the pose information of two frames to calculate the Euclidean distance, which is used to judge whether there is a closed loop. The recognition problem transforms into a binary classification problem, and its loss function is the cross-entropy loss function of binary classification. Generally, the precision-recall (PR) curve measures the experimental results of the binary classification problem. F_1 measures the advantages and disadvantages of different PR curves. F_1 is the harmonic average of precision and recall, and its expression is: $F_1 = 2\times P\times R/\left(P+R\right)$. To sum up, this paper will use the PR curve and F_1 to measure the effect of recognition. In the experiment, when the distance between the two frames is less than 3m, it is a positive example; when the distance between the two frames is greater than 20m, it is a negative example. We ignore the frames between 3m and 20m to increase the feature difference between positive and negative examples. The conditions for judging the existence of a closed-loop are: the similarity probability of the two frames exceeds a certain threshold, and the difference in the number of frames between the two frames is greater than 100. This paper mainly uses the deep learning platform PyTorch to implement the proposed method.

This paper mainly conducts four experiments: first, compare the effectiveness of the proposed method with advanced methods; then conduct robustness experiments; then conduct fusion experiments to compare different fusion methods; finally, study the influence of the topological radius on the positioning accuracy.

Comparative Experiment

Quantitative calculation analysis. This paper compares results with the most advanced recognition methods based on deep learning to verify the effectiveness of the 3D point cloud large-scale place recognition method based on GNN. The comparison methods in this paper are PNV Uy and Lee (2018), SC Kim and Kim (2018), LPD Liu et al. (2019), SG-PRKong et al. (2020), and EPC Hui et al. (2021).

In this paper, the PR curve of the comparative experiment is shown in Figure 5, and the maximum F_1 score is shown in Table 1. The GNN-PR has a relatively good effect on all datasets by comparing the results. The experimental results are relatively stable, and the F_1 score for all datasets is the highest. In the



KITTI00

KITTI02

KITTI05

KITTI08

nuScenes

Figure 5. The Precision-Recall (PR) curves obtained from recognition comparison experiments on SemanticKitti and nuScenes datasets

KITTI08 dataset, the effect of other methods has significantly decreased due to the reverse closed loop, while GNN-PR still maintains relatively good performance. Mainly because of the following: 1. This paper uses the local and global features fusion method to capture richer information, better adaptability to environmental changes and stronger robustness; 2. This paper uses GNN features, which usually reflect the essential attributes of point cloud instance objects, which are only related to their point cloud data. Thus, the impact on the environment is relatively small.

Qualitative analysis. This experiment uses KITTI05 for qualitative experiments and the place recognition diagram in Figure 6. Here, the 821th frame and the 2570th frame are examples of descriptions. Although the vehicle passes through the intersection from different directions, it can still be recognized commonly and located accurately by the

Table 1. F_1 **max score on datasets**

Method	00	02	05	08	nuS
PNV	0.882	0.791	0.734	0.765	0.785
SC	0.937	0.858	0.955	0.811	0.935
LPD	0.892	0.923	0.924	0.907	0.921
SG-PR	0.969	0.891	0.905	0.900	0.965
EPC	0.958	0.930	0.962	0.936	0.963
GNN-PR	**0.975**	**0.951**	**0.978**	**0.943**	**0.971**

proposed method in this paper. Frame 821 first passes through the intersection, and after detouring ”8”, it passes through the intersection again. At this time, it forms a closed loop, eliminating the cumulative error. This paper uses GNN features to perform feature matching. The two sub-graphs at the top of Figure 6 are topological structure diagrams of two frames. Each node represents a different object in the scene, and the edges between nodes represent the connection relationship between objects. The greater the weight of the edge, the closer the relationship between them. Graph structure for matching can improve the relocation accuracy and the system’s robustness.

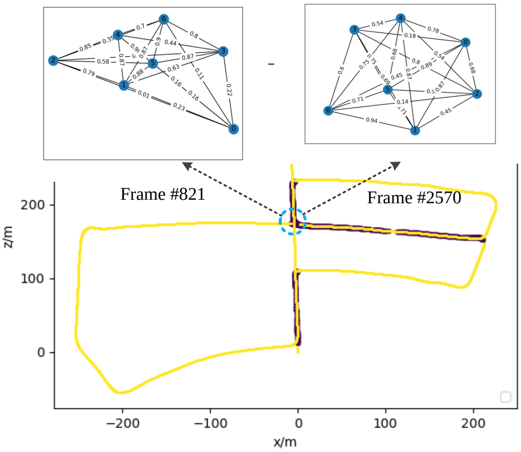


Figure 6. KITTI05 palce recognition diagram

Robustness Test

In this experiment, dynamic objects are removed, and static objects are used to build a graphic model, which facilitates the expression of a scene. However, some dynamic objects in the natural environment, such as pedestrians, bicycles, and motor vehicles, appear in the background. To achieve effective recognition, it is necessary to consider the impact of these dynamic points on the matching. In the robustness test experiment, some points delete randomly to verify the influence of dynamic objects occlusion on the recognition accuracy. The point cloud data obtained by LIDAR from different angles are different, affecting the relocation method’s stability. It must consider the impact of angle changes on the experimental results. In this experiment, the robustness of the proposed method to the angle transformation is verified by changing the angle randomly.

Table 2 shows the proposed method’s robustness to occlusion and angle changes. Because GNN-PR uses the instance segmentation object as the node, it has higher-level abstract features. It is thus insensitive to local feature and angle changes and has better stronger robustness. PNV and LPD are based on PointNet to extract features, and the ability to describe point features is insufficient, so it is easy to receive interference. SC adopts the method of scanning context and introduces repetitive calculation, which is robust to disturbances. SG-PR builds the model and recognition based on semantic features, robust against disturbances. EPC improves adaptability to the environment through efficient global features and has better robustness.

Research on Fusion Method

This paper studies the fusion method of GNN attribute, topological structure features and geometric features. The GNN attribute features, topological structure features, and geometric features are abbreviated as ATT, SRT, and GEO. (ATT+STR)+ GEO means that attribute features and topology features are fused first, then GEO features. (ATT+ GEO)+STR indicates that attribute features merge with GEO features first, then topology

Table 2. **Results of robustness experiment**

	00	02	05	08	nuS	Cmp
PNV	0.777	0.696	0.632	0.718	0.767	-9.2%
SC	0.916	0.847	0.925	0.721	0.929	-3.5%
LPD	0.831	0.869	0.871	0.859	0.885	-4.1%
SG-PR	0.951	0.856	0.874	0.849	0.962	-3.0%
EPC	0.950	0.921	0.961	0.932	0.959	-0.5%
GNN-PR	**0.972**	**0.947**	**0.975**	**0.938**	**0.969**	**-0.3%**

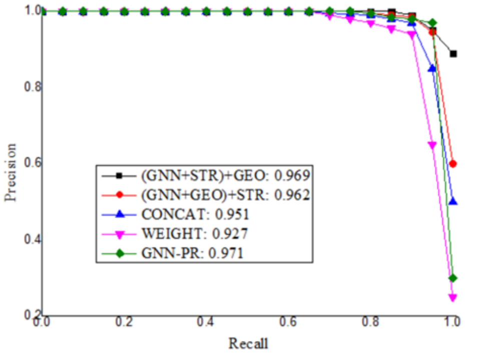


Figure 7. Experimental results of different fusion methods

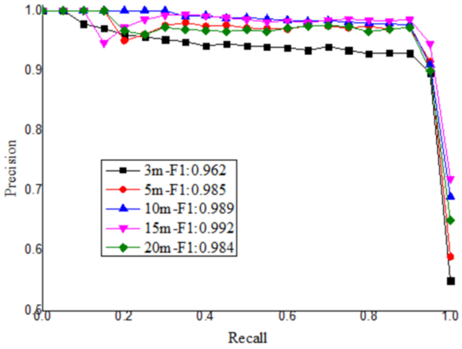


Figure 8. The influence of topology radius on recognition

features. The third method involves concatenating all the features together. The fourth method is the weighted sum of all features. We use nuScenes to study the fusion method.

The experimental results are shown in Figure 7. The fusion method improves the matching accuracy, makes the results more stable, and has strong robustness. Analysis of the nature of features, local features represent the intrinsic properties of each object, and global features include the macroscopic features of the entire system. The fusion of features with different properties can learn from each other and make full use of features. The fusion method has strong robustness to random disturbances.

We compared the operating efficiency of different fusion methods, because the basic components of different fusion methods are the same, so the number of parameters of fusion methods is similar, about 4M. In the experiments on the SemanticKitti and nuScenes datasets, the running time of the first stage is 10 $\sim$ 12ms, and the average running time of the second stage is about 5ms, which basically meets the real-time requirements of relocation in autonomous driving.

Topological Radius Influence

The topological radius plays a decisive role in the topological structure features. The larger the topological radius, the more associated objects. The more edges to construct the graph model, the stronger the object constraint relationship. This paper verifies the influence of topology radius on recognition accuracy through experiments. This experiment studies the recognition accuracy when the topological radius is 3m, 5m, 10m, 15m, and 20m. The experimental results are shown in Figure 8.

It can be seen from experiments that the larger the topology radius, the higher the recognition accuracy, but when the radius exceeds a threshold, the increase in accuracy will slow down. The main reasons are: when the radius is small, there are more isolated points, and the connection between objects becomes weak, so the effect on recognition will also decrease. When the radius is large, the number of objects associated with each object is relatively large, the coupling between objects is too strong, easy to receive interference and poor robustness.

Conclusion

This paper proposed an innovative method for 3D point cloud large-scale place recognition based on GNN. It fused the local and global features to calculate similarity scores and improve accuracy and robustness. The fusion of topological structure and geometric features improved the expression ability of global features. The experiment on SemanticKitti and nuScenes datasets proved the effectiveness of this method. However, the proposed method in this paper still needs improvement. For example, this method performed relatively poor when distinguishing similar scenes in different locations since the more abstract features express weaker local subtle features. The next step will be to take measures to improve the problems of this method.

References

Arandjelovic, R.; Gronat, P.; Torii, A.; Pajdla, T.; and Sivic, J. 2016. NetVLAD: CNN architecture for weakly supervised place recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 5297–5307.

Bai, Y.; Ding, H.; Bian, S.; Chen, T.; Sun, Y.; and Wang, W. 2019. Simgnn: A neural network approach to fast graph similarity computation. In *Proceedings of the Twelfth ACM International Conference on Web Search and Data Mining*, 384–392.

Barros, T.; Garrote, L.; Pereira, R.; Premebida, C.; and Nunes, U. J. 2021a. AttDLNet: Attention-based DL Network for 3D LiDAR Place Recognition. *arXiv preprint arXiv:2106.09637*.

Barros, T.; Pereira, R.; Garrote, L.; Premebida, C.; and Nunes, U. J. 2021b. Place recognition survey: An update on deep learning approaches. *arXiv preprint arXiv:2106.10458*.

Bauer, D.; Patten, T.; and Vincze, M. 2021. ReAgent: Point Cloud Registration using Imitation and Reinforcement Learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 14586–14594.

Du, J.; Wang, R.; and Cremers, D. 2020. Dh3d: Deep hierarchical 3d descriptors for robust large-scale 6dof relocalization. In *European Conference on Computer Vision*, 744–762. Springer.

Fischer, K.; Simon, M.; Olsner, F.; Milz, S.; Gross, H.-M.; and Mader, P. 2021. Stickypillars: Robust and efficient feature matching on point clouds using graph neural networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 313–323.

He, L.; Wang, X.; and Zhang, H. 2016. M2DP: A novel 3D point cloud descriptor and its application in loop closure detection. In *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 231–237. IEEE.

Hui, L.; Cheng, M.; Xie, J.; and Yang, J. 2021. Efficient 3d point cloud feature learning for large-scale place recognition. *arXiv preprint arXiv:2101.02374*.

Kim, G.; and Kim, A. 2018. 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)*, 4802–4809. IEEE.

Kong, X.; Yang, X.; Zhai, G.; Zhao, X.; Zeng, X.; Wang, M.; Liu, Y.; Li, W.; and Wen, F. 2020. Semantic graph based place recognition for 3d point clouds. In *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 8216–8223. IEEE.

Lee, J.; Lee, I.; and Kang, J. 2019. Self-attention graph pooling. In *International Conference on Machine Learning*, 3734–3743. PMLR.

Li, L.; Wang, M.; Ni, B.; Wang, H.; Yang, J.; and Zhang, W. 2021. 3D Human Action Representation Learning via Cross-View Consistency Pursuit. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 4741–4750.

Liu, Z.; Zhou, S.; Suo, C.; Yin, P.; Chen, W.; Wang, H.; Li, H.; and Liu, Y.-H. 2019. Lpd-net: 3d point cloud learning for large-scale place recognition and environment analysis. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2831–2840.

Qi, C. R.; Su, H.; Mo, K.; and Guibas, L. J. 2017. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 652–660.

Ribeiro, L. F.; Saverese, P. H.; and Figueiredo, D. R. 2017. struc2vec: Learning node representations from structural identity. In *Proceedings of the 23rd ACM SIGKDD international conference on knowledge discovery and data mining*, 385–394.

Shi, W.; and Rajkumar, R. 2020. Point-gnn: Graph neural network for 3d object detection in a point cloud. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 1711–1719.

Uy, M. A.; and Lee, G. H. 2018. Pointnetvlad: Deep point cloud based retrieval for large-scale place recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 4470–4479.

Veličković, P.; Cucurull, G.; Casanova, A.; Romero, A.; Lio, P.; and Bengio, Y. 2017. Graph attention networks. *arXiv preprint arXiv:1710.10903*.

Wang, D.; Cui, P.; and Zhu, W. 2016. Structural deep network embedding. In *Proceedings of the 22nd ACM SIGKDD international conference on*

Knowledge discovery and data mining, 1225–1234.

Wang, M.; Ni, B.; and Yang, X. 2020. Learning multi-view interactional skeleton graph for action recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.

Wang, Y.; Sun, Y.; Liu, Z.; Sarma, S. E.; Bronstein, M. M.; and Solomon, J. M. 2019. Dynamic graph cnn for learning on point clouds. *Acm Transactions On Graphics (tog)*, 38(5): 1–12.

Yew, Z. J.; and Lee, G. H. 2018. 3dfeat-net: Weakly supervised local 3d features for point cloud registration. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 607–623.

Ying, R.; You, J.; Morris, C.; Ren, X.; Hamilton, W. L.; and Leskovec, J. 2018. Hierarchical graph representation learning with differentiable pooling. *arXiv preprint arXiv:1806.08804*.

Zhang, W.; and Xiao, C. 2019. PCAN: 3D attention map learning using contextual information for point cloud based retrieval. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 12436–12445.

Zhang, Y.; Zhou, Z.; David, P.; Yue, X.; Xi, Z.; Gong, B.; and Foroosh, H. 2020. Polarnet: An improved grid representation for online lidar point clouds semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 9601–9610.