GCP : Graph Convolution with Pooling for Place recognition

Yunhun Nam[1*], Wongi Park[2*], Wonseok Oh[3*]
Department of {Mechanical[1], Software and Computer[2] Engineering}, University of Ajou,
Electrical and Computer Engineering, University of Michigan[3],
{ yh0326, psboys}@ajou.ac.kr, okong@umich.edu

Abstract

Despite extensive studies on 3D point cloud place recognition, there are still limitations due to the failure to extract sophisticated local features. In this paper, we introduce a novel network that incorporates a Graph Convolution and pooling ***(GCP-Net)****. We leverage 3D graph convolution to enhance detailed features. Furthermore, in traditional place recognition, the issue of not using pooling in the process of extracting local features is addressed by adopting 3D Graph Max-Pooling. Although graph methods have various advantages, the absence of pooling led to the adoption of PointNet and sparse convolution. This approach allowed the use of graph methods without the need to reduce the batch size. Through extensive experiments on retrieval tasks, we demonstrate that our method performs better than existing methods and even shows competitive results on four different datasets. Our code is available at: https://github.com/yunpal/city-challenge*

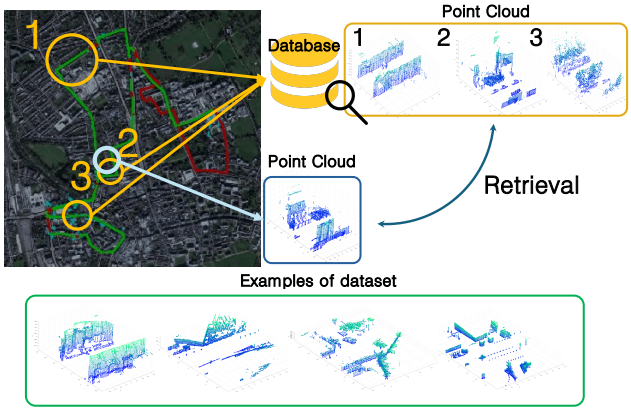


Figure 1. 3D point cloud place recognition employs descriptors from raw 3D data to identify locations. A trained network computes descriptors for query point clouds, facilitating localization by matching them with point clouds in a database. This method enables accurate recognition across diverse conditions.

1. Introduction

Place recognition is an essential part of the 3D vision and robotics communities and has been widely adapted to many fields such as simultaneous localization and mapping(SLAM)[2, 5, 19], Autonomous Driving(AD)[3, 8, 10, 17, 22, 27, 37], and augment reality[20, 21, 28, 39]. Place recognition is mainly categorized into two ways; image-based methods and point-cloud-based methods. Since image-based methods find it hard to capture local features, recent efforts have focused on point cloud[16, 30, 38] for place recognition, proposing algorithms that generate distinctive descriptors [34]. Initially, PointNet[30] has been utilized in the point cloud by extracting discriminate features. Many studies[30, 35, 43] leverage PointNet. PointNetVLAD[35] leverages PointNet[30] to extract local features and adopts NetVLAD[1] for generating descriptors. However, PointNetVLAD[35] makes it hard to generalize descriptions of the point cloud. To address the problem, LPD-Net [18] employs a graph-based module to extract local features of the point clouds focusing on the structural and spatial complexities of the data. However, it is limited to predicting the significance of local features; point contextual attention network (PCAN)[43] further refined the feature extraction through attention mechanism, enhancing point-wise feature representation. These algorithms need to be revised to fully capture the spatial understanding inherent in 3D data. Although the innovative application of [30, 35, 43], they overlook the inter-point structural connections. Conversely, LPD-Net's [18] efficacy is impeded due to its dependence on fully connected layers, constraining its capacity to utilize the intricacies of 3D data fully. Graph-convolution-based methods [12, 13, 32, 33] introduce graph-based convolutions to enhance local feature extraction by addressing the challenge of inter-point connections in 3D point cloud place recognition. These methods incorporate proxy points, edge convolution, and hierarchical graph convolutions to harness graph-based method-

*Equal contribution.

ologies for improved structural details and feature comprehension preservation. Nevertheless, these approaches use graph convolution to extract local features, and the substantial computational load required often reduces the batch size during the training process, leading to poorer descriptor quality. 2D Encoding based methods [4, 9, 23–25] address the challenge of substantial computational load. These studies aim to enhance the network's resilience against viewpoint alterations, facilitating its management of perspective variation. However, the inherent loss of information during the transition from 3D to 2D results in the acquisition of poor local features. Consequently, despite endeavors to extract adequate features, the loss of information severely constrains the network's performance. Sparse convolution-based methods [6, 14, 42] propose networks incorporating 3D sparse convolution to improve computational efficiency and address the issues associated with information loss in 2D encoding. With the increasing attention towards self-attention-based methodologies, it's evident that leveraging such approaches has become prevalent in place recognition, primarily aimed at elevating representation quality. SVT-Net[6] leverage transformers, thus making it possible to learn both short-range local features and long-range contextual features. On the other hand, Transloc3D, as presented in [42], employs 3D sparse convolution enhanced by Efficient Channel Attention and transformers. This integration effectively combines convolutional and transformer technologies to extract local features in place recognition tasks efficiently. These methods concentrate on learning contextual features and enhancing efficient attention. However, sparse convolution-based methods utilize voxels to reduce computational loads, yet this technique yields inferior local features compared to those obtained using raw points. Moreover, a limitation of Transformers is their significant need for extensive training data to improve network performance. Despite the clear advantage of Graph Convolution-based methods in extracting superior local features compared to other methods, these approaches require substantial computational loads. This limitation hinders the ability to stack multiple network layers and results in extended data processing times. Moreover, adjustments to the batch size are required during the training process. These limitations restrict the use of Graph Convolution-based methods in 3D Place Recognition, where large-scale datasets are used. To address this problem, we propose a Graph Convolution-based with pooling for 3D point cloud Place recognition**(GCP-Net)**. We leverage a 3D graph convolution network(3D GCN) and 3D graph Max-pooling introduced in [16] to enable the extraction of informative local features using graph-based methods. 3D Graph Convolution Network, which considers not only inter-point connections but also structural information. This network offers the advantage of considering structural information, unlike previous graph methods used in place recognition. In 3D point cloud place recognition, we adopt 3D Graph Max-pooling, a technique not previously used, to enable graph methods without reducing the batch size during the training process and to allow for deeper layer construction. Thus, we obtained discriminative descriptors representing each submap and performed highly in the retrieval task. Finally, our method performs better than existing methods and even shows competitive results on four different datasets. The following are the contributions of this paper.

- We proposed a novel Graph Convolution-based Network (GCP-Net) to overcome the issue of high computational loads associated with existing graph convolution-based methods.
- We identified problems from not using pooling in 3D point cloud place recognition and addressed them by adopting 3D Graph Max-pooling. This has laid the foundation for further advancements in 3D point cloud place recognition.
- We conducted experiments on four benchmark datasets[26, 34] and achieved higher performance than existing methods, even showing higher performance than existing methods and even showing competitive results on four different datasets.

2. Related Work

2.1. PointNet-based methods

PointNetVLAD [34] was the first end-to-end network proposed for 3D LiDAR place recognition, addressing the issue of point clouds being order-independent. It utilizes PointNet to extract local features and then employs NetVLAD to create a global descriptor, leveraging the fact that both PointNet and NetVLAD are indifferent to the input order. However, due to its use of PointNet [30], this method fails to consider the relationships between points when extracting local features. LPD-Net [18] aims to address the limitations of previous methods in capturing structural and spatial features in 3D LiDAR place recognition. It utilizes 10 hand-crafted local features and a graph-based approach to learning these features. Despite this innovative approach, LPD-Net faces challenges due to its reliance on multilayer perceptron. This reliance limits its ability to fully leverage the complexities of the data, preventing substantial performance improvements. PCAN[43] takes a similar approach to PointNetVLAD in obtaining local features but aims to craft a more efficient descriptor that can give weight to each point using a Point Contextual Attention Network. Despite this, both methods struggle to account for the structural relationships between points due to their reliance on PointNet.

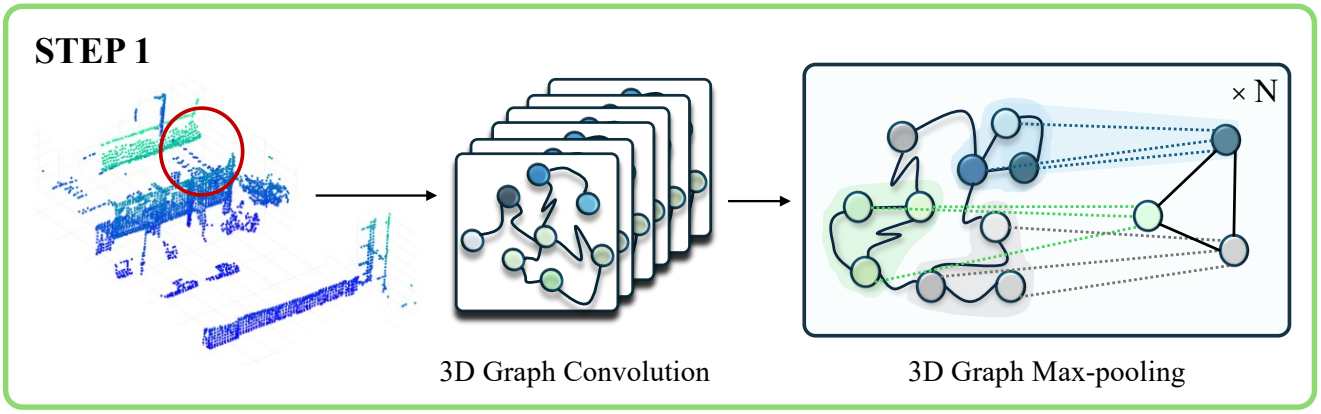


Figure 2. Overview of extracting local features: The 3D Graph Convolution Network and 3D Graph Max-pooling are aimed at capturing detailed local features. The 3D Graph Convolution extracts features through relationships with neighboring points in Cartesian coordinates, while the 3D Graph Max-pooling pools representative values of local features to reduce computational load.

2.2. 2D Encoding-based methods

The 2D Encoding method involves converting the 3D point cloud into a 2D image using various techniques, fed into the network as input. Examples of 2D images include range images [4, 9, 23–25], and digital elevation maps [9]. There is also CVT[25], which utilizes both Bird Eye View(BEV) and range images.

2.3. Sparse convolution-based methods

SVT-Net[6] applies transformers to the modified voxels, such as atom-based sparse voxels and cluster-based sparse voxels. Similarly, Transloc3D[42] employs 3D sparse convolution followed by Efficient Channel Attention and transformers to extract local features, combining advanced convolution techniques and transformer technology in 3D spatial recognition networks. LoGG3d-Net[36] employs local consistency loss and global loss during its training phase, diverging from the conventional method of using global loss. This approach of leveraging both local and global loss metrics has improved performance. MinkLoc3D[14], a simpler network, employs sparse convolution and Feature Pyramid Network for the extraction of local features, and utilizes Generalized Mean Pooling to obtain descriptors made significant strides in MinkLoc3D v2[15] by introducing a new loss function and dynamically adjusting the batch size during training to optimize the effectiveness of the loss function. CrossLoc3D employs a diffusion model to ensure that data from two distinct sources depicting the same scene are uniformly represented within the same embedding space. CASSPR[41] obtained local features based on points and voxels and then fused the two features using a cross-attention transformer.

2.4. Graph convolution-based methods

EPC-NET[13] leverages proxy points to replace multiple neighbors, enabling the extraction of local features through proxy convolution. It effectively employs Grouped-VLAD for parameter reduction and utilizes graph convolution techniques to adeptly capture local features, emphasizing its capability to preserve structural details. DAGC[33] utilized a dual attention module and ResGCN, which combines DGCNN[38] with residual connections. This network uses point-wise attention mechanisms and channel-wise attention mechanisms to understand the relationships between points and features, respectively, and then extracts local features. PPT-Net[12] utilizes edge convolution, as introduced in DGCCN, for structural learning during local feature extraction and a Pyramid Point Transformer Network to understand spatial relationships between local features. It combines feature maps of varying sizes using a VLAD module, with both models effectively capturing structural features through graph-based methods. Hierarchical Bidirected Graph Convolutions for Large-Scale 3-D Point Cloud Place Recognition [32] points out the limitations of traditional graph convolution methods, which use k-nearest neighbors to obtain features from adjacent points. It aims to effectively extract features through hierarchical bidirected graph convolutions.

3. Methods

3.1. Local Feature

The 3D Graph Convolution Network[16] depicted in Figure 3 is designed to capture local features with structural information. It considers n neighboring points in Cartesian coordinates to extract features, thereby enabling the acquisition of structural local features. Although all networks used in place recognition maintain point clouds during the extrac-

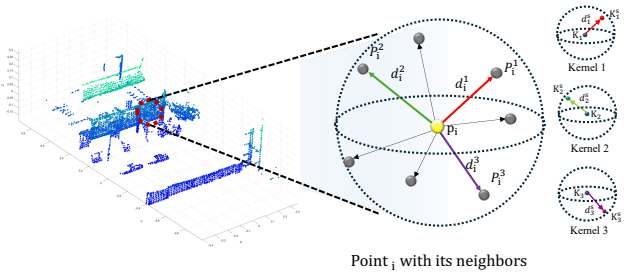


Figure 3. A point i within a point cloud, along with its 7 neighboring points and 3 kernels. Unlike other graph methods, this approach uses direction to obtain structural information.

tion of local features, which has limited the use of graph methods[12, 33], the max pooling proposed in [16] overcomes the limitations of conventional lidar place recognition by enabling the acquisition of detailed local features.

3D Graph Convolution Network, capable of extracting not only structural but also spatial local features from a point cloud, is defined by the following equation:

$$\text{3D Graph Conv}\left(R_i^n, K_m\right) = \left\langle f\left(p_i\right), w_i \right\rangle + \max_n \left\{ \left\langle f\left(p_i^n\right), w_i^s \right\rangle + \frac{\langle d_n^i, d_k^s \rangle}{\|d_n^i\| \|d_k^s\|} \right\} \tag{1}$$

The $\mathcal{N}$ points in point cloud is represented as $P = \{p_1, \ldots, p_N\} \in \mathbb{R}^3$, and the m neighboring points of the p_i are denoted as $P_m^i = \{p_1^i, p_2^i, \ldots, p_m^i\} \in \mathbb{R}^3$. The local features of a point p_i are represented by $f(p_i)$ and direction d_n^i represents the direction from p_i to p_n^i. Kernel's central weight, represented as $K_m = \{K_1, K_2, \ldots, K_m\}$. Each kernel K_i has n supports, each of which includes a weight w_i^s and a support direction d_k^s, which represents the direction from the kernel's center to its support. It considers the neighbors' points in Cartesian coordinates, which allows for obtaining local features that are invariant to shifts and scales. Additionally, using the dot product between each point and the kernel enables more accurate extraction of structural local features.

3D Graph Max-pooling [16] is similar to max pooling in 2D CNNs. Max pooling in a 2D CNN reduces the dimensions of the data by selecting the maximum value, thereby simplifying the input while preserving essential features. 3D Graph Max-pooling is shown in Algorithm 1. First, each point within a point cloud initially updates its feature value to the highest feature value among its n neighboring points through comparison. Subsequently, points from these $\mathcal{N}$ neighbors are pooled.

Algorithm 1 3D Graph Max-Pooling

Require: $Points, Features$
Ensure: $Pooled\_Points, Pooled\_Features$
$Pooled\_Points \leftarrow$ empty list
$Pooled\_Features \leftarrow$ empty list
for each point p in $Points$ **do**
  $Neighbors \leftarrow$ find_n_nearest_neighbors
  $Max\_Feature \leftarrow -\infty$
  for each $neighbor$ in $Neighbors$ **do**
    if $Features[neighbor] > Max\_Feature$ **then**
      $Max\_Feature \leftarrow Features[neighbor]$
    end if
  end for
  $Features[p] \leftarrow Max\_Feature$
end for
$Reduced\_Points \leftarrow$ random_sampling
for each point r in $Reduced\_Points$ **do**
  Append r to $Pooled\_Points$
  Append $Features[r]$ to $Pooled\_Features$
end for
return $Pooled\_Points, Pooled\_Features$

3.2. Descriptor

The NetVLAD[1] is a neural network designed for aggregating local features into a compact descriptor for place recognition tasks. It aggregates unordered local feature input to create descriptors through the equation as follows:

$$V(k) = \sum_{i=1}^{N} \frac{e^{w_k^T p_i + b_k}}{\sum_{k'} e^{w_{k'}^T p_i + b_{k'}}} (p_i - c_k), \tag{2}$$

where a point of point cloud $p_i \in \mathbb{R}^D$ represents the $\mathcal{D}$-dimensional features, $V \in \mathbb{R}^{L \times D}$ denotes the VLAD with $\mathcal{L}$ number of clusters, $V(k)$ denotes the k_{th} cluster and $c_k \in \mathbb{R}^D$ denotes a cluster to which the local features are assigned. The VLAD representation passes through a fully connected layer to generate a global descriptor $F \in \mathbb{R}^{1 \times D}$. It captures the essence of aggregating the differences between each point's local features and the cluster c_k, each weighted by the likelihood of assignment to cluster c_k. Essentially, it aims to distribute the local features' information based on the c_k criteria, thereby achieving an aggregation that considers all local features. NetVLAD[1] dynamically summarizes local features into a descriptor optimized for specific clusters, designated as c_k. This process involves evaluating the variations among local features relative to each cluster center, effectively capturing the distinctiveness of each feature set. By assigning weights based on the probability that a feature belongs to a particular cluster, NetVLAD ensures that local features efficiently contribute to the final descriptor.

3.3. Lazy Quadruplet Loss

Inspired by the evolution of loss functions, We adopt the Lazy Quadruplet Loss function introduced in [29, 34]. The following equation defines the Lazy Quadruplet Loss:

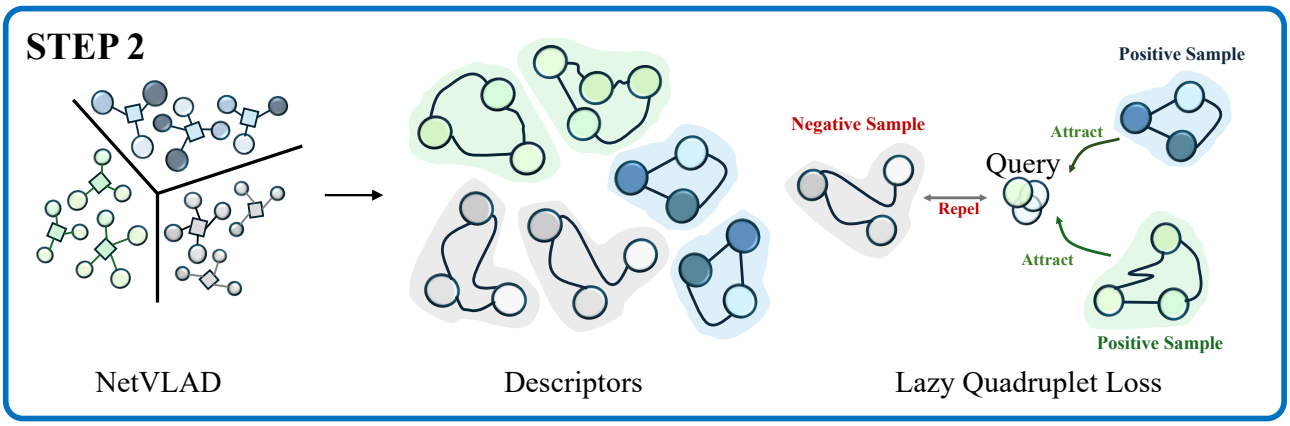


Figure 4. The overview of aggregating local features: Netvlad aggregates local features into descriptors and uses these descriptors along with a Lazy Quadruplet Loss to train the network. In this process, descriptors in a positive relationship with a query attract each other, while descriptors in a negative relationship with a query repel each other.

$$\begin{aligned}\mathcal{L}_{LQ} = &\max\left(\left[\alpha + Dist^2_{\text{pos}} - Dist^2_{\text{neg}}\right]_+\right) \\ &+ \max\left(\left[\beta + Dist^2_{\text{pos}} - Dist^2_{\text{neg}^*}\right]_+\right),\end{aligned} \tag{3}$$

where, α and β is margin and $[.]_+$ denotes the hinge loss. The max operator helps to select a hard positive sample and a hard negative sample from the query. $Dist_{\text{pos}}$ is the Euclidean distance between the query and one of the positive descriptors. $Dist_{\text{neg}}$ is the Euclidean distance between the query and negative descriptors. $Dist_{\text{neg}^*}$ is the Euclidean distance between negative descriptors and negative's negative descriptors. The loss function encourages the query and positive descriptors to be closer while expanding the query and negative descriptors. Additionally, it expands the negative and negative's negative descriptors.

4. Experiments

4.1. Implementation details

In our network, the margins for α and β used the Lazy Quadruplet Loss function are 0.5 and 0.2, respectively. In the 3D Graph Convolutional Network, the support number is 1, and the number of neighbors is 20. In NetVLAD, 64 clusters are used. During the training process, two batches are used. Each batch consists of 1 query, 2 positives, 18 negatives, and 1 additional negative. The training process is conducted over 20 epochs. After the initial 5 epochs, we use a strategy of selecting hard negatives for training. Our algorithm leverages components from PointNetVLAD and 3D GCN.

4.2. Dataset & Evaluation Metric

Four distinct datasets are used to verify our GCP-Net, which include the Oxford RobotCar dataset [26] along with three

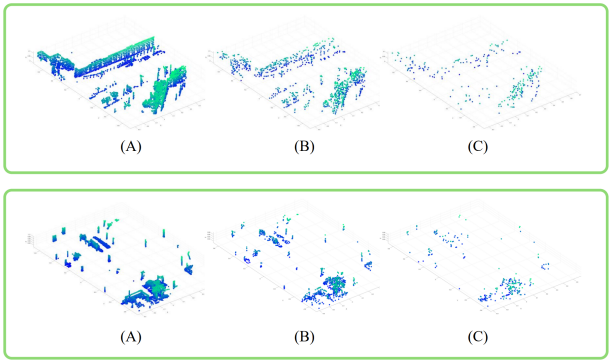


Figure 5. (A) consists of point clouds with 4096 points, (B) with 1024 points, and (C) with 256 points.

in-house datasets [34]. The Oxford RobotCar dataset contains observations gathered from a 10km route repeated 44 times using a SICK LMS-151 2D LiDAR system and records UTM coordinates. The in-house datasets were collected using a Velodyne-64 LiDAR system. Specifically, these include data from the university sector dataset, which spans a 10km circuit completed five times, the residential area dataset covering an 8km circuit also completed five times, and the business district dataset, which includes five laps around a 5km circuit. Each dataset also records UTM coordinates. The submaps of four datasets are preprocessed to remove the ground surface, and each submap is downsampled to contain 4096 points. They are centered around the origin of the UTM coordinate system, with the point cloud positioned within the range of [-1, 1] from the central origin. For the Oxford dataset, preprocessing is performed on all points within 20 meters of the vehicle's trajectory. For the in-house datasets, preprocessing is conducted on all points within a 25m x 25m bounding box.

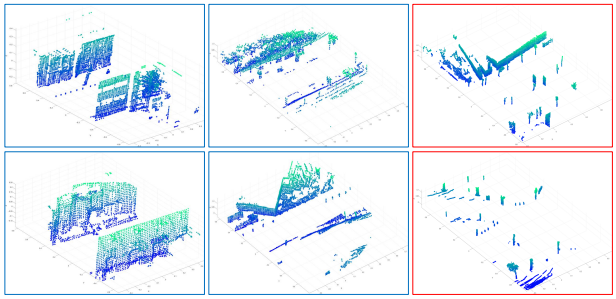
	Oxford		U.S.		R.A.		B.D.	
	AR@1	**AR**@1%	**AR**@1	**AR**@1%	**AR**@1	**AR**@1%	**AR**@1	AR@1%
PointNetVLAD [34]	62.8	80.3	63.2	72.6	56.1	60.3	57.2	65.3
PCAN [43]	69.1	83.8	62.4	79.1	56.9	71.2	58.1	66.8
LPD-Net [18]	86.3	94.9	87.0	96.0	83.1	90.5	82.5	89.1
EPC-Net [40]	86.2	94.7	-	96.5	-	88.6	-	84.9
SOE-Net [40]	89.4	96.4	82.5	93.2	82.9	91.5	83.3	88.5
MinkLoc3D [14]	93.0	97.9	86.7	95.0	80.4	91.2	81.5	88.5
HiTPR [11]	86.6	93.7	80.9	90.2	78.2	87.2	74.3	79.8
NDT-T [44]	93.8	97.7	-	-	-	-	-	-
PPT-Net [12]	93.5	98.1	90.1	97.5	84.1	93.3	84.6	90.0
SVT-Net [6]	93.7	97.8	90.1	96.5	84.3	92.7	85.5	90.7
TransLoc3D [42]	95.0	98.5	-	94.9	-	91.5	-	88.4
MinkLoc3Dv2 [15]	**96.3**	**98.9**	90.9	96.7	86.5	93.8	86.3	91.2
CrossLOC [7]	94.4	98.6	-	-	-	-	-	-
CASSPR [41]	95.6	98.5	**92.9**	97.9	89.5	94.8	87.9	92.1
GCP-Net(**ours**)	92.5	97.7	**92.9**	**98.3**	**89.7**	**95.0**	**89.0**	**93.2**

Table 1. Average recall (%) at top 1% (@1%) and top 1 (@1) for each model trained on the Oxford RobotCar dataset.

The submaps from four datasets are preprocessed to remove the ground surface, with each submap reduced to 4096 points with voxel grid filter[31]. These are centralized around the UTM coordinate system's origin, positioning the point cloud within a [-1, 1] range from the center. In the case of the Oxford dataset, all points within 20 meters of the vehicle's trajectory are preprocessed. For the in-house datasets, all points within 25m x 25m bounding box of the vehicle's trajectory are preprocessed. The GCP-Net is trained using 21,711 submaps from the Oxford dataset and was evaluated on 3,030 Oxford submaps not involved in training, as well as on 4,542 submaps from three in-house datasets representing a university sector, a residential area, and a business district, referred to as U.S., R.A., and B.D., respectively. For the training process, a submap is labeled as positive if it lies within 10 meters of the query submap and as negative if it is more than 50 meters away from the query submap. Place recognition evaluates its effectiveness through a retrieval task. In the task, a descriptor generated from a query map is used to search a database containing descriptors of known locations. The goal is to find the closest match based on the descriptors, with successful retrieval typically defined by finding a location descriptor within a specified geometric distance from the query. This method is used for evaluation, testing the network's capacity to accurately recognize and match places even with changes in viewpoint and environmental conditions. Success is achieved if at least one of the search results retrieved by a query submap falls within a geometric distance of 25 meters. Our evaluation metric uses Average Recall@1 and Average Recall@1%, as used in PointNetVLAD[34].

4.3. Evaluation Results

We conduct evaluations on four different datasets and benchmark our results against alternative approaches, as detailed in Table 1. Our GCP-Net marked improvements in performance on the U.S., R.A., and B.D. datasets, surpassing previous state-of-the-art achievements by significant margins. In particular, for the U.S. dataset, we achieved an increase of 1.3% in AR@1 and 0.4% in AR@1%. For R.A., we report improvements of 0.2% in AR@1 and 0.2% in AR@1%. Additionally, for B.D., we observe a notable improvement of 1.1% in AR@1 and 1.1% in AR@1%. Although our results on the Oxford dataset do not surpass state-of-the-art levels, these enhancements underscore the robustness of our network across various domains. We conduct evaluations on four different datasets and benchmark our results against alternative approaches, as detailed in Table. Our GCP-Net marked improvements in performance on the U.S., R.A., and B.D. datasets, surpassing previous state-of-the-art achievements by significant margins. In particular, for the U.S. dataset, we achieved an increase of 1.3% in Average Recall@1 and 0.4% in Average Recall@1%. In the R.A. dataset, we observed enhancements of 0.2% in both Average Recall@1 and Average Recall@1%. Moreover, the B.D. dataset showed substantial gains with an improvement of 1.1% in both Average Recall@1 and Average Recall@1%. While our performance on the Oxford dataset did not exceed that of the leading method, the overall enhancements highlight the effectiveness and adaptability of our network across different environmental contexts, confirming its robust performance across a variety of environments. These results not only demonstrate the capability of our approach to effectively navigate and recognize diverse



(A) (B) (C)

Figure 6. Examples of retrieval successes and failures when using our network. The top part is the query point clouds, and the bottom part is the point clouds of the retrieval. (A) and (B) show the nearest correct matches to the query. (C) shows an erroneous match to the query, indicating a mismatch within our retrieval process.

geographical areas but also establish a new benchmark for future research in place recognition. The robustness of our network is highlighted by its use of pooling while extracting local features, which distinguishes it from other networks by preserving high levels of performance despite variations in the domain. This stability is primarily due to the detailed extraction of local features, made possible through the use of pooling.

5. Ablation Study

5.1. Qualitative results

For a comprehensive understanding, we conducted a qualitative evaluation of our proposed GCP-Net. Figure 6 illustrates both success and failure cases in retrieval tasks. Despite utilizing sophisticated techniques for extracting local features, the pooling method used proved less effective in comparing different submaps accurately. Nonetheless, as shown in Table 1, our method outperforms other methods, confirming that our approach not only manages complex retrieval tasks effectively but also significantly advances over existing methods. These results underline the potential and limitations of our network's matching accuracy.

6. Limitations and future work

Graph-based methods are highly effective at extracting local features. However, the computational demands of graph-based methods limit their ability to process large amounts of data at once, which has restricted their network performance. Nonetheless, the adoption of 3D Graph max-pooling in lidar place recognition has helped overcome these limitations and has contributed to the field's advancement. However, 3D Graph Max-pooling randomly samples points from the point cloud, which presents a drawback as it fails to differentiate between significant and insignificant points.

7. Conclusion

In this paper, we show Graph Convolution with Pooling Network(GCP-Net) for place recognition integrates a 3D Graph Convolution Network to greatly improve the extraction of discriminative local features. Furthermore, by utilizing Graph Max-pooling for Graph Convolution, we reduced the size of the network. Our proposed GCP-Net is demonstrated through experimental results on four different datasets, showing that our method not only outperforms existing methods but also shows competitive results.

References

[1] Relja Arandjelovic, Petr Gronat, Akihiko Torii, Tomas Pajdla, and Josef Sivic. Netvlad: Cnn architecture for weakly supervised place recognition. In *CVPR*, 2016. 1, 4

[2] Tim Bailey and Hugh Durrant-Whyte. Simultaneous localization and mapping (slam): Part ii. *IEEE robotics & automation magazine*, 13(3):108–117, 2006. 1

[3] Li Chen, Penghao Wu, Kashyap Chitta, Bernhard Jaeger, Andreas Geiger, and Hongyang Li. End-to-end autonomous driving: Challenges and frontiers. *arXiv preprint arXiv:2306.16927*, 2023. 1

[4] Xieyuanli Chen, Thomas Läbe, Andres Milioto, Timo Röhling, Jens Behley, and Cyrill Stachniss. Overlapnet: A siamese network for computing lidar scan similarity with applications to loop closing and localization. *Autonomous Robots*, pages 1–21, 2022. 2, 3

[5] Hugh Durrant-Whyte and Tim Bailey. Simultaneous localization and mapping: part i. *IEEE robotics & automation magazine*, 13(2):99–110, 2006. 1

[6] Zhaoxin Fan, Zhenbo Song, Hongyan Liu, Zhiwu Lu, Jun He, and Xiaoyong Du. Svt-net: Super light-weight sparse voxel transformer for large scale place recognition. In *AAAI*, pages 551–560, 2022. 2, 3, 6

[7] Tianrui Guan, Aswath Muthuselvam, Montana Hoover, Xijun Wang, Jing Liang, Adarsh Jagan Sathyamoorthy, Damon Conover, and Dinesh Manocha. Crossloc3d: Aerial-ground cross-source 3d place recognition. In *ICCV*, pages 11335–11344, 2023. 6

[8] Christian Häne, Lionel Heng, Gim Hee Lee, Friedrich Fraundorfer, Paul Furgale, Torsten Sattler, and Marc Pollefeys. 3d visual perception for self-driving cars using a multi-camera system: Calibration, mapping, localization, and obstacle detection. *Image and Vision Computing*, 68:14–27, 2017. 1

[9] Sudarshan S. Harithas, Gurkirat Singh, Aneesh Chavan, Sarthak Sharma, Suraj Patni, Chetan Arora, and Madhava Krishna. Findernet: A data augmentation free canonicalization aided loop detection and closure technique for point clouds in 6-dof separation. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pages 8399–8408, 2024. 2, 3

[10] Gim Hee Lee, Friedrich Faundorfer, and Marc Pollefeys. Motion estimation for self-driving cars with a generalized camera. In *CVPR*, pages 2746–2753, 2013. 1

[11] Zhixing Hou, Yan Yan, Chengzhong Xu, and Hui Kong. Hitpr: Hierarchical transformer for place recognition in point cloud. In *2022 International Conference on Robotics and Automation (ICRA)*, pages 2612–2618. IEEE, 2022. 6

[12] Le Hui, Hang Yang, Mingmei Cheng, Jin Xie, and Jian Yang. Pyramid point cloud transformer for large-scale place recognition. In *ICCV*, pages 6098–6107, 2021. 1, 3, 4, 6

[13] Le Hui, Mingmei Cheng, Jin Xie, Jian Yang, and Ming-Ming Cheng. Efficient 3d point cloud feature learning for large-scale place recognition. *IEEE TIP*, 31:1258–1270, 2022. 1, 3

[14] Jacek Komorowski. Minkloc3d: Point cloud based large-scale place recognition. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pages 1790–1799, 2021. 2, 3, 6

[15] Jacek Komorowski. Improving point cloud based place recognition with ranking-based loss and large batch training. In *ICPR*, pages 3699–3705, 2022. 3, 6

[16] Zhi-Hao Lin, Sheng-Yu Huang, and Yu-Chiang Frank Wang. Convolution in the cloud: Learning deformable kernels in 3d graph convolution networks for point cloud analysis. In *CVPR*, 2020. 1, 2, 3, 4

[17] Liu Liu, Hongdong Li, Yuchao Dai, and Quan Pan. Robust and efficient relative pose with a multi-camera system for autonomous driving in highly dynamic environments. *IEEE Transactions on Intelligent Transportation Systems*, 19(8): 2432–2444, 2017. 1

[18] Zhe Liu, Shunbo Zhou, Chuanzhe Suo, Peng Yin, Wen Chen, Hesheng Wang, Haoang Li, and Yun-Hui Liu. Lpd-net: 3d point cloud learning for large-scale place recognition and environment analysis. In *ICCV*, 2019. 1, 2, 6

[19] Zhe Liu, Chuanzhe Suo, Yingtian Liu, Yueling Shen, Zhijian Qiao, Huanshu Wei, Shunbo Zhou, Haoang Li, Xinwu Liang, Hesheng Wang, et al. Deep learning-based localization and perception systems: Approaches for autonomous cargo transportation vehicles in large-scale, semiclosed environments. *IEEE Robotics & Automation Magazine*, 27(2): 139–150, 2020. 1

[20] Katie Luo, Guandao Yang, Wenqi Xian, Harald Haraldsson, Bharath Hariharan, and Serge Belongie. Stay positive: Non-negative image synthesis for augmented reality. In *CVPR*, pages 10050–10060, 2021. 1

[21] Katie Luo, Guandao Yang, Wenqi Xian, Harald Haraldsson, Bharath Hariharan, and Serge Belongie. Stay positive: Non-negative image synthesis for augmented reality. In *CVPR*, pages 10050–10060, 2021. 1

[22] Xinyu Luo, Jiaming Zhang, Kailun Yang, Alina Roitberg, Kunyu Peng, and Rainer Stiefelhagen. Towards robust semantic segmentation of accident scenes via multi-source mixed sampling and meta-learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, pages 4429–4439, 2022. 1

[23] Junyi Ma, Jun Zhang, Jintao Xu, Rui Ai, Weihao Gu, and Xieyuanli Chen. Overlaptransformer: An efficient and yaw-angle-invariant transformer network for lidar-based place recognition. *IEEE Robotics and Automation Letters*, 7(3): 6958–6965, 2022. 2, 3

[24] Junyi Ma, Xieyuanli Chen, Jingyi Xu, and Guangming Xiong. Seqot: A spatial–temporal transformer network for place recognition using sequential lidar data. *IEEE Transactions on Industrial Electronics*, 70(8):8225–8234, 2023.

[25] Junyi Ma, Guangming Xiong, Jingyi Xu, and Xieyuanli Chen. Cvtnet: A cross-view transformer network for lidar-based place recognition in autonomous driving environments. *IEEE Transactions on Industrial Informatics*, 20(3): 4039–4048, 2024. 2, 3

[26] Will Maddern, Geoffrey Pascoe, Chris Linegar, and Paul Newman. 1 year, 1000 km: The oxford robotcar dataset. *The International Journal of Robotics Research*, 36(1):3–15, 2017. 2, 5

[27] Milind Naphade, Shuo Wang, David C. Anastasiu, Zheng Tang, Ming-Ching Chang, Yue Yao, Liang Zheng, Mohammed Shaiqur Rahman, Meenakshi S. Arya, Anuj Sharma, Qi Feng, Vitaly Ablavsky, Stan Sclaroff, Pranamesh Chakraborty, Sanjita Prajapati, Alice Li, Shangru Li, Krishna Kunadharaju, Shenxin Jiang, and Rama Chellappa. The 7th ai city challenge. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, 2023. 1

[28] Liyuan Pan, Yuchao Dai, Miaomiao Liu, and Fatih Porikli. Simultaneous stereo video deblurring and scene flow estimation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4382–4391, 2017. 1

[29] Wongi Park, Inhyuk Park, Sungeun Kim, and Jongbin Ryu. Robust asymmetric loss for multi-label long-tailed learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2711–2720, 2023. 4

[30] Charles R. Qi, Hao Su, Kaichun Mo, and Leonidas J. Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *CVPR*, 2017. 1, 2

[31] Radu Bogdan Rusu and Steve Cousins. 3d is here: Point cloud library (pcl). In *2011 IEEE international conference on robotics and automation*, pages 1–4. IEEE, 2011. 6

[32] Dong Wook Shu and Junseok Kwon. Hierarchical bidirected graph convolutions for large-scale 3-d point cloud place recognition. *IEEE Transactions on Neural Networks and Learning Systems*, pages 1–12, 2023. 1, 3

[33] Qi Sun, Hongyan Liu, Jun He, Zhaoxin Fan, and Xiaoyong Du. Dagc: Employing dual attention and graph convolution for point cloud based place recognition. In *Proceedings of the 2020 International Conference on Multimedia Retrieval*, pages 224–232, 2020. 1, 3, 4

[34] Mikaela Angelina Uy and Gim Hee Lee. Pointnetvlad: Deep point cloud based retrieval for large-scale place recognition. In *CVPR*, 2018. 1, 2, 4, 5, 6

[35] Mikaela Angelina Uy and Gim Hee Lee. Pointnetvlad: Deep point cloud based retrieval for large-scale place recognition. In *CVPR*, pages 4470–4479, 2018. 1

[36] Kavisha Vidanapathirana, Milad Ramezani, Peyman Moghadam, Sridha Sridharan, and Clinton Fookes. Logg3d-net: Locally guided global descriptor learning for 3d place recognition. In *2022 International Conference on Robotics and Automation (ICRA)*, pages 2215–2221, 2022. 3

[37] Shuo Wang, David C. Anastasiu, Zheng Tang, Ming-Ching Chang, Yue Yao, Liang Zheng, Mohammed Shaiqur Rahman, Meenakshi S. Arya, Anuj Sharma, Pranamesh Chakraborty, Sanjita Prajapati, Quan Kong, Norimasa Kobori, Munkhjargal Gochoo, Munkh-Erdene Otgonbold, Ganzorig Batnasan, Fady Alnajjar, Ping-Yang Chen, Jun-Wei Hsieh, Xunlei Wu, Sameer Satish Pusegaonkar, Yizhou Wang, Sujit Biswas, and Rama Chellappa. The 8th AI City Challenge. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, 2024. 1

[38] Yue Wang, Yongbin Sun, Ziwei Liu, Sanjay E Sarma, Michael M Bronstein, and Justin M Solomon. Dynamic graph cnn for learning on point clouds. *ACM TOG*, 38(5): 1–12, 2019. 1, 3

[39] Jamie Watson, Mohamed Sayed, Zawar Qureshi, Gabriel J. Brostow, Sara Vicente, Oisin Mac Aodha, and Michael Firman. Virtual occlusions through implicit depth. In *CVPR*, pages 9053–9064, 2023. 1

[40] Yan Xia, Yusheng Xu, Shuang Li, Rui Wang, Juan Du, Daniel Cremers, and Uwe Stilla. Soe-net: A self-attention and orientation encoding network for point cloud based place recognition. In *CVPR*, pages 11348–11357, 2021. 6

[41] Yan Xia, Mariia Gladkova, Rui Wang, Qianyun Li, Uwe Stilla, João F Henriques, and Daniel Cremers. Casspr: Cross attention single scan place recognition. In *ICCV*, pages 8461–8472, 2023. 3, 6

[42] Tian-Xing Xu, Yuan-Chen Guo, Zhiqiang Li, Ge Yu, Yu-Kun Lai, and Song-Hai Zhang. Transloc3d: point cloud based large-scale place recognition using adaptive receptive fields. *Communications in Information and Systems*, 23(1):57–83, 2023. 2, 3, 6

[43] Wenxiao Zhang and Chunxia Xiao. Pcan: 3d attention map learning using contextual information for point cloud based retrieval. In *CVPR*, 2019. 1, 2, 6

[44] Zhicheng Zhou, Cheng Zhao, Daniel Adolfsson, Songzhi Su, Yang Gao, Tom Duckett, and Li Sun. Ndt-transformer: Large-scale 3d point cloud localisation using the normal distribution transform representation. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pages 5654–5660. IEEE, 2021. 6