SBD^2-Net : Domain-Invariant 3D Structural Block and Double Descriptor for Place Recognition

Anonymous CVPR submission

Paper ID 49

Abstract

*Despite various 3D point cloud place recognition studies, limitations persist in fully ensuring generalization across diverse environments. In this paper, we introduce a domain-invariant network that incorporates a 3D structural block and double descriptor **(SBD^2-Net)**. We leverage 3D graph convolution to enhance spatial structural features. Furthermore, we aggregate these features in two distinct ways to create the key and global features descriptors, each representing the same submap differently. These descriptors, combined with our proposed directional Euclidean loss, consider the previously overlooked but critical relationship between positive and negative samples, thereby enhancing place recognition capabilities across changing environments or dataset domains. Through extensive experiments on retrieval tasks, we demonstrate that our method shows higher performance than existing methods and even shows competitive results on four different datasets.*

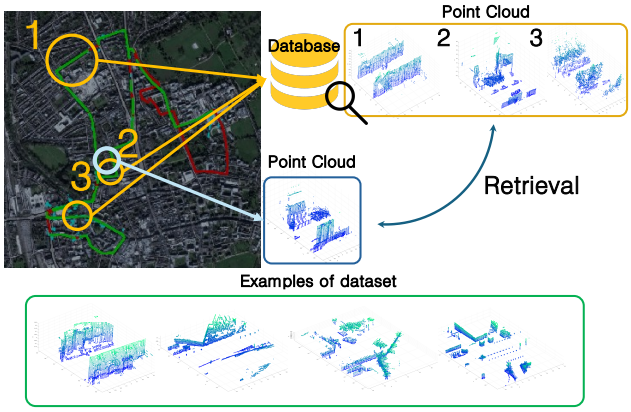


Figure 1. Point cloud-based place recognition employs global descriptors from raw 3D data to identify locations. A trained network computes descriptors for query point clouds, facilitating localization by matching them with point clouds in a database. This method enables accurate recognition across diverse 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, 6, 19], Autonomous Driving(AD)[4, 9, 11, 17, 22, 26, 34], and augment reality[20, 21, 27, 36]. Place recognition is mainly categorized into two ways: image-based methods and point-cloud-based methods. However image-based methods hard to capture local features, recent efforts have focused on point cloud[16, 28, 35] for place recognition, proposing algorithms that generate distinctive descriptors [31]. Initially, PointNet[28] has been utilized in place recognition by extracting discriminate features to learn the local features and generate global descriptors for retrieval. PointNetVLAD[32], which leverages PointNet[28] to extract point features and adopts NetVLAD[1] for generating global descriptors. However, PointNetVLAD[32] makes it hard to generalize global 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 predict significance of local features, Point contextual attention network (PCAN)[41] further refined the feature extraction through attention mechanism, enhancing point-wise feature representation. However, these algorithms encounter limitations in fully capturing the spatial understanding inherent in 3D data. Despite the innovative application of [1, 28, 32, 41], 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. To address these constraints, sparse convolution-based techniques [7, 14, 40] propose networks incorporating 3D sparse convolution to improve noise and computational efficiency robustness. With the increasing attention towards self-attention-based methodologies, it's evident that leveraging such approaches has become prevalent in place recog-

nition, primarily aimed at elevating representation quality. SVT-Net[7] 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 [40], 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, it is limited to learning the full contextual features of 3D spaces and fails to ensure generalization across diverse environments fully. Additionally, due to their low bias, Transformers demand substantial training data to bolster model generalization. Nonetheless, the challenge lies in the practical constraints of gathering ample real-world environment data, rendering them unsuitable for plane recognition applications. [5, 10, 23–25] address the challenge of balancing spatial and structural context. While 2D encoding-based methodologies aim to bolster the network's resilience against viewpoint alterations, facilitating its management of perspective variation, the inherent loss of spatial structural information during the transition from 3D to 2D undermines adaptability across domains. Consequently, despite endeavors to extract adequate features, the insufficiency of spatial structural information severely constrains the network's adaptiveness. However, the graph-convolution-based methods, including EPC-NET, DAGC, PPT-Net, and Hierarchical Bidirected Graph Convolutions [13, 29, 30], introduces graph-based convolutions to enhance local feature extraction in 3D point cloud place recognition. These methods incorporate proxy points, edge convolution, and hierarchical graph convolutions to harness graph-based methodologies for improved structural details and feature comprehension preservation. Nevertheless, while these approaches primarily focus on node and edge relationships, they often overlook the intricacies of structural relationships. Consequently, the importance of spatial structural features remains underestimated. Despite the improvement of voxel-based and point cloud-based methods in a way that complements each other, it still remains a dilemma that point cloud-based methods are vulnerable to sparse points part and noise, whereas voxel-based methods often fail to capture fine-grained localization and complex local features. With these limitations, the challenge in using point pooling during local feature extraction stems from each point's features having insufficient information. To address this problem, we propose a domain-invariant network that incorporates a 3D structural block and double descriptor **(SBD^2-Net)**. We leverage a 3D graph convolution network(GCN) and denoising point pooling to make it robust from the noise point. Also, a 3D structural block, which consists of GCN and pooling methods, enhances learning for structural local feature-based points. Furthermore, selecting randomly often suffers from considering noise points or avoiding extracting significant points. Thus, we improve feature selection and noise removal by dividing each point into three categories based on local density: core, boundary, and noise. Pooling is then conducted proportionally to the number of points in each of these three categories, allowing us to select important features and eliminate noise. Thus, we obtained domain-invariant features in the Submap and showed high performance in the retrieval task. Finally, our method shows higher performance than existing methods and even shows competitive results on four different datasets. The following are the primary contributions of this paper.

- We proposed a domain-invariant network that incorporates a 3D structural block and double descriptor (SBD^2-Net) by preserving point cloud data as domain-invariant information in various environments.
- We propose a novel directional Euclidean loss function for place recognition, focused on accounting for the relationship between positive and negative samples, using key and global feature descriptors for its application.
- We proposed the Denoising Pooling (DP) method, which improves feature selection and noise removal by dividing each point's neighborhood into three categories based on local density: core, boundary, and noise.

2. Related Work

2.1. Pointnet based methods

The introduction of PointNetVLAD [31] marked a pivotal shift towards end-to-end networks that efficiently handle the order-independence of point clouds, pairing PointNet's [28] local feature extraction with NetVLAD's [1] global descriptor creation. LPD-Net [18] expands upon this, introducing hand-crafted local features and a graph-based learning approach to better capture structural nuances. PCAN [41] attempts to refine the approach by incorporating a Point Contextual Attention Network, aiming to enhance descriptor efficiency through the weighting of point significance. Nonetheless, it still confronts the inherent limitations of PointNet 's[28] structural relationship analysis.

2.2. Graph conv-based methods

To overcome the limitations of PointNet in capturing structural local features, graph-based methods have emerged. EPC-NET [13] introduces proxy points and proxy convolution, combined with grouped VLAD network for efficient feature extraction and parameter reduction, though it avoids point pooling which might limit complexity reduction efforts. DAGC [30] leverages dual attention mechanisms and a Residual Graph Convolution Network to deepen understanding of point and feature relationships. PPT-Net [12] introduces edge convolution and a Pyramid Point Transformer

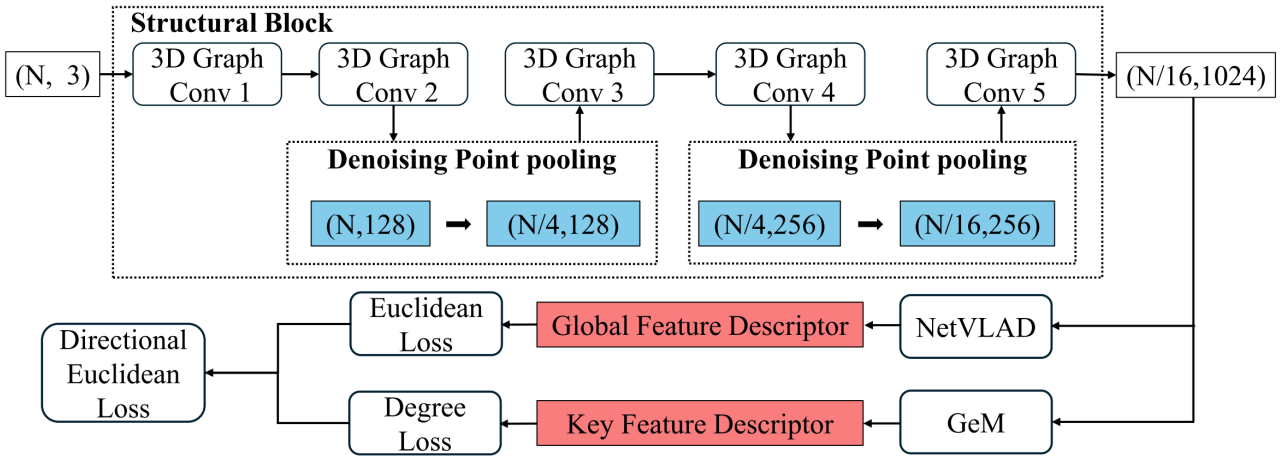


Figure 2. An overview of the proposed SBDD^2-Net. The structural block is composed of 3D graph convolution and denoising point pooling, aimed at capturing detailed spatial structural context. Local features are aggregated into global and key feature descriptors via NetVLAD and GeM, and directional Euclidean loss is for optimization.

Network for nuanced structural and spatial feature analysis. Meanwhile, approaches like Hierarchical Bidirected Graph Convolutions [29] aim to overcome the limitations of k-nearest neighbor reliance on traditional graph convolutions.

2.3. Sparse conv-based methods

Networks including SVT-Net [7] and Transloc3D [40] leverage 3D sparse convolution and transformers, including techniques like atom-based sparse voxels, cluster-based sparse voxels, Efficient Channel Attention, to enhance 3D spatial recognition. MinkLoc3D[14] and MinkLoc3D v2[15] focus on sparse convolution and feature pyramid networks for efficient feature extraction, employing dynamic batch size adjustments for optimal training. LoGG3D-Net [33] introduces local consistency and global loss for improved descriptors. Crossloc3D [8] 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 [39] enhances feature extraction by obtaining local features from points and voxels, followed by feature fusion through a cross-attention transformer.

3. Methods

3.1. 3D Structural Block

The structural block depicted in Figure 2 is designed to capture local features enriched with spatial structural contexts. We have refined the graph-kernel-based convolution part originally introduced by 3D-GCN [16], a concept initially validated on the ModelNet40 [37] dataset. This refinement has been tailored to embody a structural block

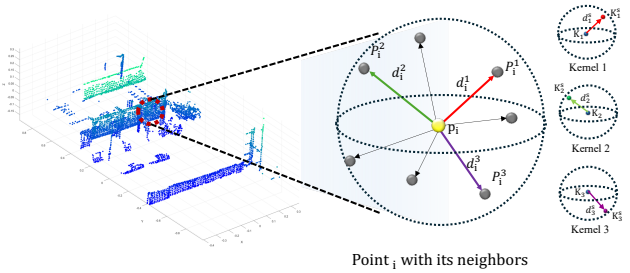


Figure 3. A point$_i$ within a point cloud, along with its 7 neighboring points, and 3 kernels. Unlike other graph methods, this approach uses direction to obtain structural information.

within our architecture, facilitating effective place recognition across extensive spatial datasets through a proposed Denoising Point Pooling (DP Pooling) method. DP Pooling method is a novel component specifically developed to improve the performance by addressing the shortcomings of existing methods when applied to sparse point clouds. By integrating DP Pooling, our structural block adeptly navigates these challenges, achieving a superior representation of spatial structural features.

3D graph convolution, capable of extracting spatial structural local features from a point cloud, is defined by the following equation.

$$\begin{aligned}\text{Structural Conv}\left(p_i, K_i\right) &= \left\langle f\left(p_i\right), w_{ic}\right\rangle \\ &+ \max_n \left\{ \left\langle f\left(p_{ni}\right), w_{is}\right\rangle + \frac{\left\langle dir_n, dir_{ks}\right\rangle}{\|dir_n\|\|dir_{ks}\|} \right\}\end{aligned} \tag{1}$$

The point cloud is represented in Cartesian coordinates as $P = \{p_1, \ldots, p_N\} \in \mathbb{R}^3$, and the neighborhood

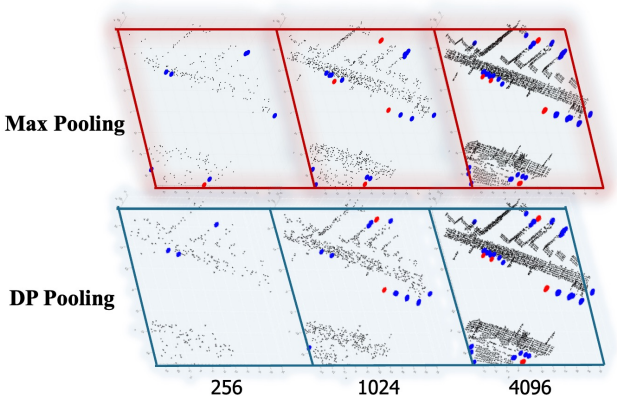


Figure 4. Black points represent the core section, blue points indicate the boundary section and red points correspond to the noise section. In the case of Max Pooling, points belonging to the noise section are not completely removed; however, in the case of DP Pooling, points in the noise section are entirely eliminated.

of p_i is denoted in Cartesian coordinates as $P_i = \{p_{1i}, p_{2i}, \ldots, p_{Ni}\} \in \mathbb{R}^3$. The features of a point p_s are represented by $f(p_s)$ and direction dir_n represents the direction from p_i to p_{ni}. Each kernel comprises a central element and a support element, represented as $\{K_1, K_2, \ldots, K_d\}$. Within a single kernel K_i, it is defined by a central weight w_{ic}, a support weight w_{is}, and a support direction dir_{ks}, where dir_{ks} represents the direction from the kernel's center to its support. This approach enables extracting features that more accurately capture spatial structural characteristics.

3.2. Denoising Point Pooling

Limitation of Maxpooling, In the previous method, maxpooling [16] is utilized. However, in this process, there existed a limitation in that the points for pooling were selected randomly. Since selecting randomly means selecting noise and significant points with equal probability, we adjusted this pooling method to suit sparse point clouds better.

Denoising Point Pooling We propose an appropriate point pooling approach, called Denoising Point Pooling, which selectively samples to remove noise effectively. The proposed DP pooling is shown in Algorithm 1. In Algorithm 1 normalized $D[p]$ is small, the point is in an area with a high density of surrounding points, and if $D[p]$ is large, the point is in an area with a low density of surrounding points. Points are assigned to different regions based on the normalized value, that is, the density. The regions are divided based on density: the cores are quite dense, while the boundaries are relatively sparse, and noise is significantly sparse. Accordingly, the points are distributed into three major sections: core section, boundary section, and noise section. Points are pooled based on each section's proportion, prioritizing removing outliers to pool the points with a focus on noise reduction. As shown in its pooling effect in Figure 4, the proposed method effectively removes noise, thus making better structural and spatial local features.

Algorithm 1 Denoising Point Pooling Algorithm

$N \leftarrow$ Total number of points
$S \leftarrow N/4$ (Number of points to sample)
Define the number of intervals θ
$D \leftarrow$ an array of the sum of distances for each point
for each point p in all points **do**
  $D_p \leftarrow 0$
  for each neighboring point q in the k nearest points of p **do**
    $D_p \leftarrow D_p + \text{distance}(p, q)$
  end for
  $D[p] \leftarrow D_p$
end for
Normalize D by using $\frac{D[p]-\min(D)}{\max(D)-\min(D)}$
$H \leftarrow$ a count histogram of points in each interval
for each point p in all points **do**
  $i \leftarrow$ the index of the interval for $D[p]$
  $H[i] \leftarrow H[i] + 1$
end for
$T \leftarrow$ an array of target number of samples per interval
for $i \leftarrow 0$ to 2 **do**
  $T[i] \leftarrow \lfloor \frac{H[i]}{N} \times S \rfloor$
  Select $T[i]$ samples from interval i
end for

3.3. Double Descriptor

Local features extracted through the structural block contain spatial information. Two methods for aggregating local features to create descriptors that represent the same submap exist. First, NetVLAD generates descriptors encapsulating the overall local features. Second, GeM generates descriptors reflecting important local features.

Global Feature Descriptor NetVLAD [1] aggregates local features through the equation as follows:

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

where $x_i \in \mathbb{R}^D$ represents the local features of a point that are subject to aggregation, $c_k \in \mathbb{R}^D$ denotes a k_{th} cluster to which the local features are assigned and $V \in \mathbb{R}^{L \times D}$ denotes the VLAD representation with L number of clusters. The VLAD representation passes through a fully connected layer to generate a global descriptor $F \in \mathbb{R}^D$.

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.

Key Feature Descriptor GeM aggregates local features as follows:

$$g^{(k)} = \left(\frac{1}{n} \left[f_1^{(k)} \right]_+^p + \left[f_2^{(k)} \right]_+^p + \left[f_3^{(k)} \right]_+^p + \ldots + \left[f_{n-1}^{(k)} \right]_+^p + \left[f_n^{(k)} \right]_+^p \right)^{\frac{1}{p}}, \tag{3}$$

where the hinge loss denoted by $[\ ]_+$, $f \in \mathbb{R}^{N \times D}$ denotes the local features of the points, the descriptor $g \in \mathbb{R}^D$ represents the aggregated local features, with k indicating the feature of the k^{th} dimension. The parameter p is a learnable parameter that influences the aggregation process.

This concept revolves around the synergy between hinge loss and the parameter p, enabling the GeM to aggregate significant features effectively.

3.4. Directional Euclidean Loss

In this study, we propose a new loss function that considers not only the distance between samples but also the direction between samples.

The lazy quadruplet Loss, which is called Euclidean loss introduced in [31], is defined by the following equation:

$$\begin{aligned} L_{Euclidean} = &\max\left([m_1 + ED_{\text{pos}} - ED_{\text{neg}}]_+ \right) \\ &+ \max\left([m_2 + ED_{\text{pos}} - ED_{\text{other\_neg}}]_+ \right), \end{aligned} \tag{4}$$

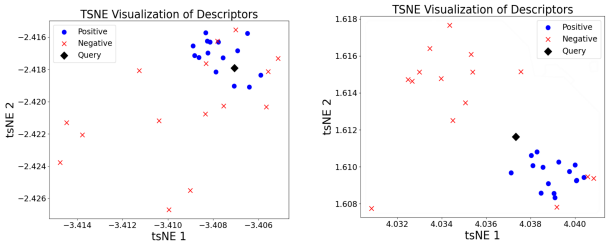
where hinge loss denoted by $[\ ]_+$. ED_{pos} is the distance between the query and positive descriptors, aiming to be minimized. Conversely, ED_{neg} represents the distance between the query and negative descriptors targeted for expansion. Additionally, ED_{other_neg} indicates the distance between the negative descriptor and the negative descriptors.

In contrast, degree loss L_{Degree} considers the angular relationships between descriptors.

$$\begin{aligned} L_{Degree} = &\max\left([m_1 + D_{\text{pos}} - D_{\text{neg}}]_+ \right) \\ &+ \max\left([m_2 + D_{\text{pos}} - D_{\text{other\_neg}}]_+ \right), \end{aligned} \tag{5}$$

which considers the relationship between positive and negative samples. D_{pos}, representing the angle formed between the query and positive descriptors, aims to minimize this angle to zero degrees. Conversely, D_{neg}, representing the angle formed between the query and negative descriptors, aims to maximize this angle to 180 degrees. Similarly, $D_{\text{other\_neg}}$ representing the angle formed between the negative and negative's negative descriptors aims to expand this angle to 180 degrees.

Those two losses are combined as a total loss $L_{total} = \lambda L_{Degree} + L_{Euclidean}$, where λ is a designed loss ratio parameter of L_{Degree}. By considering both Euclidean distance and angle within the embedding space, we address the previously overlooked aspect of the relationship between positive and negative samples, thereby creating a more discriminative descriptor.



(a) Euclidean Loss (b) Directional Euclidean Loss

Figure 5. In case (a), where the relationship between positive and negative samples is not considered, positives and negatives are positioned in similar directions. However, in case (b), where the relationship between the two is considered, positives and negatives are positioned opposite to each other, with the query as the reference point.

3.5. Training Method

As aforementioned, the process of creating descriptors are divided into key and global feature descriptors. Additionally, it is necessary to incorporate a degree loss into the loss function. The loss that simultaneously considers angle and distance through a single descriptor does not perform well Due to adjusting distance and angle, they are not equivalent. Therefore, to address this issue, different losses are applied to different descriptors.

By applying Euclidean loss to global feature descriptors, we ensure that the entire set of features moves properly in the embedding space. By applying degree loss to local feature descriptors, an important set of features is fine-tuned.

In retrieval tasks, a global feature descriptor is particularly effective due to its ability to encapsulate the entire set of features, thereby ensuring a generalized representation. However, what truly sets our approach apart is the enhanced learning of key features within this global descriptor. We have fine-tuned the learning process by leveraging angles to focus on these crucial features. This not only makes the descriptor general but also discriminative.

CVPR #49

	Oxford		U.S.		R.A.		B.D.	
	AR@1	**AR**@1%	**AR**@1	**AR**@1%	**AR**@1	**AR**@1%	**AR**@1	AR@1%
PointNetVLAD [31]	62.8	80.3	63.2	72.6	56.1	60.3	57.2	65.3
PCAN [41]	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 [38]	86.2	94.7	-	96.5	-	88.6	-	84.9
SOE-Net [38]	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
PPT-Net [12]	93.5	98.1	90.1	97.5	84.1	93.3	84.6	90.0
SVT-Net [7]	93.7	97.8	90.1	96.5	84.3	92.7	85.5	90.7
TransLoc3D [40]	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 [8]	94.4	98.6	-	-	-	-	-	-
CASSPR [39]	95.6	98.5	92.9	97.9	89.5	94.8	87.9	92.1
SBD^2-Net(**ours**)	93.3	97.9	**94.2**	**98.6**	**91.0**	**95.2**	**89.4**	**93.7**

Table 1. Average recall (%) at top 1% (@1%) and top 1 (@1) for each of the models trained on the Oxford RobotCar. Our SBD^2-Net method achieves the best performance on all benchmarks.

4. Experiments

4.1. Implementation details

In our network, the values for $m1$ and $m2$ used in the Euclidean loss function and the degree loss function are 0.5 and 0.2, respectively. The balance function for $\lambda = 1$ to equally balance each loss value. Hyperparameters $theta$ is equal to three and k is equal to 20 in the DP pooling algorithm.

During the training process, we utilized batches consisting of 1 query, 2 positives, 18 negatives, and 1 other negative, with two such batches being used for each training step. The number of epochs is set to 20. Additionally, we selected negatives using hard negatives after 5 epochs. Our proposed algorithm is implemented on top of PointNetVLAD[31] and 3D-GCN[16]. When a single submap with 4096 points is input into the model, the processing time is 13ms. For PointNetVLAD, the processing time is 5ms. we still get the real-time processing time while maintaining robustness The utilized computer for our experiments are Intel Xeon with NVIDIA RTX A5000 D6 24GB.

4.2. Dataset & Evaluation Metric

Four datasets are utilized to demonstrate including the Oxford RobotCar dataset[3] and three in-house datasets. The Oxford RobotCar dataset comprises data collected from a 10km area covered 44 times using a SICK LMS-151 2D LiDAR, including UTM coordinates. The in-house datasets were obtained using a Velodyne-64 and all include UTM coordinates. Specifically, the university sector dataset covers a 10km area for 5 laps, the residential area dataset covers an 8km area for 5 laps, and the business district dataset covers a 5km area for 5 laps.

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.

Our model is trained via 21,711 submaps from the Oxford dataset and conducted evaluations on 3,030 Oxford submaps not used in the training phase and on 4,542 submaps from three in-house datasets of a university sector, a residential area, and a business district, abbreviated as U.S., R.A., and B.D. respectively. For training, a submap is considered positive if it is within 10 meters of the query, and considered negative if it is beyond 50 meters from the query. Four evaluation datasets each create their own database, within which query submaps search through their respective databases. Success is achieved if at least one of the search results found by a query submap is within a geometric distance of 25 meters. Our evaluation metric uses Average Recall@1 and Average Recall@1%, as used in PointNetVLAD[31].

4.3. Evaluation Results

We conduct evaluations across four datasets and compare the results with other methods, as shown in Table.1. Our method demonstrates considerable performance enhancements on the U.S., R.A., and B.D., indicating significant advancements over the state-of-the-art on three datasets. Specifically, for the U.S., we note a performance increase of 1.3% in AR@1 and 0.7% in AR@1%. For R.A., we report

improvements of 1.5% in AR@1 and 0.4% in AR@1%. Additionally, for B.D., we observe a notable improvement of 1.5% in AR@1 and 1.6% 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.

This robustness is further emphasized by our network's utilization of point pooling, setting it apart from other networks by maintaining high performance without significant degradation even as the domain shifts. This resilience is primarily due to the denoising of local features and the spatial structural features, coupled with an effective aggregation method and directional Euclidean loss, leading to these significant performance improvements

5. Ablation Study

5.1. Quantitative results.

We conduct experiments on nine distinct cases to demonstrate the effectiveness of the proposed methods shown in Table 2. These experiments aim to compare the impact of key features and global feature descriptors, as well as the influence of Euclidean loss and degree loss. Cases 1 through 5 only use one descriptor, while cases 6 through 9 incorporate two kinds of descriptors. In this context, losses derived from key feature descriptors are referred to as key losses, and those derived from global descriptors are called global losses.

Case	Pooling	Descriptor		Loss		Oxford
		Key	Global	key	Global	AR@1
1	Max Pooling	o	x	degree	-	87.6
2		o	x	-	euclidean	91.8
3		x	o	degree	-	89.3
4		x	o	-	euclidean	92.2
5		x	o	-	degree+euclidean	91.3
6		o	o	euclidean	euclidean	92.4
7		o	o	degree	euclidean	92.9
8	DP Pooling	o	o	euclidean	euclidean	92.6
9		o	o	degree	euclidean	**93.3**

Table 2. Experimental comparison with Max pooling and DP Pooling.

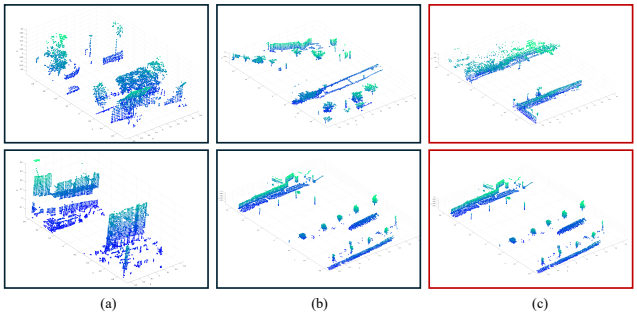


Figure 6. Examples of retrieval successes and failures when using our network. The top part is the query point clouds, and the bottom part is the point clouds of the retrieval. (a) and (b) show the nearest correct matches to the query. (c) shows an erroneous match to the query, indicating a mismatch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

The Effect of Angular Constraints on Learning In experiment cases 1 and 3, where degree loss is applied, positive and negative samples are positioned in opposite directions in the embedding space relative to the query, highlighting a clear separation. In cases 2 and 4, where Euclidean loss is applied, the method adjusts the distances between the query, positive, and negative in the embedding space. When comparing the outcomes of case 1 with 2, and case 3 with 4, it is observed that relying solely on degree loss is insufficient for achieving optimal learning outcomes. The decreased performance with degree loss is attributed to its inability to reduce the distance between the query and positive samples while also failing to expand the distance between the query and negative samples. This implies that degree loss serves only the role of fine-tuning.

Descriptor Comparison Comparing the outcomes of cases 1 and 3, as well as cases 2 and 4, it is observed that cases 3 and 4, which incorporated NetVLAD, demonstrated superior performance. This suggests that NetVLAD is more effective in capturing the entirety of features, leading to enhanced performance.

The performance in experiment case 5, which applies Euclidean loss and degree loss to one descriptor, is inferior to that in case 4, which applies Euclidean loss to the same descriptor. This is attributed to the fact that in the embedding space, reducing the distance between the query and positive samples is not equivalent to making their angle 0 degrees, and similarly, increasing the distance between the query and negative samples is not equivalent to making their angle 180 degrees. As a result, applying two kinds of losses to one descriptor is not proper.

The results in case 6, which uses two different descriptors, surpass those of case 4, which uses one descriptor. Despite being trained with the same loss, important features are subjected to the same constraints twice. Even though the constraints are identical, focusing attention on important features has a positive impact on performance enhancement. This suggests that applying constraints multiple times to key features, especially through the use of key feature descriptors, significantly enhances the effectiveness of the retrieval task. This result proves to us that training with two types of descriptors is effective.

Merits of Directional Euclidean Loss In experiment cases 6 and 7, though both use two different descriptors, the results show that case 7 performs better. Similarly, when comparing cases 8 and 9, which also use two different descriptors, the results of case 9 are superior. Notably, cases 7 and 9 utilize degree loss, whereas cases 6 and 8 do not. This

indicates that our proposed directional Euclidean loss helps in creating more discriminative descriptors.

Merits of DP Pooling In the comparisons, case 8 outperforms case 6, and case 9 surpasses case 7 in performance. Cases 6 and 7 use Max Pooling, whereas cases 8 and 9 utilize DP Pooling. The results demonstrate that DP Pooling outperforms Max Pooling, indicating that DP Pooling's ability to reduce noise contributes to enhanced performance.

6. Limitations and future work

DP Pooling By extracting local features from a raw point cloud using our proposed DP pooling, we can obtain local features that are robust to noise. This method has been shown to improve performance. However, our approach has a limitation: the pooling is based on a density that we define. If this process could be improved beyond the hand-crafted method, it would be possible to pool truly significant points.

Directional Euclidean Loss As shown in Figure 5, even when using directional Euclidean loss, positive and negative samples are not always positioned in opposite directions in the embedding space relative to the query. Furthermore, some negative samples are positioned in the same direction as the positive samples. Although degree loss succeeds to some extent in distancing the positive from the negative samples, there are limitations to perfectly classifying them. Improving this to ensure that positive and negative samples are precisely positioned opposite each other relative to the query in the embedding space could enhance performance.

Retrieval Results Success and failure cases can be found in Figure 6. Due to previously mentioned limitations, our proposed network sometimes fails in retrieval tasks. However, our novel approaches enable it to successfully execute significantly challenging retrieval tasks. These examples highlight the challenges and potential areas for improvement in our network's matching accuracy.

7. Conclusion

In this paper, a domain-invariant network for place recognition incorporates a 3D structural block, significantly enhancing the comprehension of spatial structural features. The proposed place recognition SBD^2-Net is proven with dominant cross-domain robustness in different datasets without overfitting to each domain. Nevertheless, this approach may not yield optimal results when datasets are sufficiently large and the application is strictly limited to fitting within a specific data domain. To achieve the most superior performance under both intra-domain and inter-domain conditions, it is imperative to enhance model generalization capabilities or to employ reinforcement learning strategies for more effective optimization of cross-attention parameters.

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, 2, 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] Dan Barnes, Matthew Gadd, Paul Murcutt, Paul Newman, and Ingmar Posner. The oxford radar robotcar dataset: A radar extension to the oxford robotcar dataset. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pages 6433–6438. IEEE, 2020. 6

[4] 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

[5] 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

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

[7] 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. 1, 2, 3, 6

[8] 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. 3, 6

[9] 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

[10] 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

[11] 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

[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. 2, 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. 2

[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. 1, 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, 3, 4, 6

[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

[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

[26] 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

[27] 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

[28] 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

[29] 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. 2, 3

[30] 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. 2

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

[32] 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

[33] 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

[34] 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

[35] 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

[36] 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

[37] Zhirong Wu, Shuran Song, Aditya Khosla, Fisher Yu, Linguang Zhang, Xiaoou Tang, and Jianxiong Xiao. 3d shapenets: A deep representation for volumetric shapes. In *CVPR*, 2015. 3

[38] 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

[39] 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

[40] 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. 1, 2, 3, 6

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