2024 IEEE International Conference on Robotics and Automation (ICRA 2024)
May 13-17, 2024. Yokohama, Japan

LPS-Net: Lightweight Parameter-shared Network for Point Cloud-based Place Recognition

Chengxin Liu[1], Guiyou Chen[1*] and Ran Song[1]

***Abstract*— With innovation in fields such as autonomous driving and augmented reality, point cloud-based place recognition has gained significant attention. Many methods try to address this problem by extracting and matching global descriptors in a database, but they often must balance the extraction of comprehensive contextual information and large model sizes. To overcome this challenge, we propose a lightweight parameter-shared network (LPS-Net), which includes multiple bidirectional perception units (BPUs) to extract multi-scale long-range contextual information and parameter-shared NetVLADs (PS-VLADs) to aggregate descriptors. A BPU includes a parameter-shared convolution module (SharedConv) that significantly compresses the model and enhances its ability to capture informative features. In PS-VLADs, we replace half the parameters used in the original NetVLAD with trainable scalars, which further reduces the model size, and theoretically prove their equivalence. Experimental results demonstrate that LPS-Net achieves state-of-the-art performance at the task of point cloud-based place recognition while maintaining a small model size. Code and supplementary materials can be found at https://github.com/Yavinr/LPS-Net.**

I. Introduction

Large-scale place recognition has become a key technology in applications such as simultaneous localization and mapping (SLAM) [1], augmented reality [2], and autonomous driving [3]. A common solution is to extract global descriptors from a set of images and/or point clouds of the unknown scenes, and then find the records in a labeled database that have the closest descriptors in Euclidean space to those descriptors. Finally, key information for place recognition, such as the coordinates of objects in the scene, can be estimated.

This work focuses on the extraction of global descriptors in point cloud-based place recognition tasks. PointNetVLAD [4] first solved this problem by combining PointNet [5] and NetVLAD [6], followed by a few methods [7]–[9] that improved the discrimination of global descriptors such as through handcrafted features, graph neural networks, and attention mechanisms. Nevertheless, these methods can only extract features from shallow to deep, and can only aggregate descriptors from the deepest features. PPT-Net [10] uses a hierarchical network to extract descriptors from features at multiple scales. However, its architecture is based on fully connected layers, which reduces the ability to summarize information. Furthermore, the above methods employ the original VLAD (Ori-VLAD) structure used by PointNetVLAD [4] to extract descriptors. We will show that half of the trainable parameters in this type of VLAD are proportionally correlated.

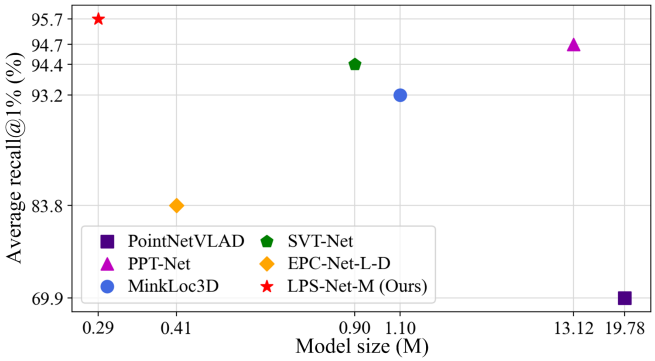

Fig. 1. Performance of different methods trained on Oxford dataset. We measure the mean AR@1% of each method across four datasets under the benchmark proposed by PointNetVLAD [4].

We propose a lightweight parameter-shared network (LPS-Net) for point cloud-based large-scale place recognition tasks. As shown in Fig. 1, it performs well even with significant size compression. LPS-Net has three components: 1) a parameter-shared convolution module (SharedConv); 2) bidirectional perception unit (BPU); and 3) parameter-shared NetVLAD architecture (PS-VLAD). SharedConv is a small module used in a BPU to reduce the dimensionality of vectors with minimal information loss. By stacking multiple BPUs, LPS-Net can better explore long-range contextual features across multiple scales. To elegantly aggregate descriptors from point-wise features extracted by BPUs, PS-VLAD simplifies Ori-VLAD, replacing half of the parameters with the product of a scalar and the remaining parameters, and LPS-Net uses multiple PS-VLADs to aggregate descriptors from both the features generated by BPUs and raw geometric features. Finally, the flattened multi-scale descriptors are synthesized into a global descriptor by SharedConv. We theoretically demonstrate the equivalence between PS-VLAD and Ori-VLAD, and conducted experiments on the Oxford dataset and three in-house datasets. Both theoretical analysis and experimental results indicate that LPS-Net achieves state-of-the-art performance.

The main contributions of this paper are as follows:

- A lightweight parameter-shared network (LPS-Net) for point cloud-based place recognition achieves state-of-the-art performance;
- The SharedConv module can reduce the dimensionality

*Corresponding author.
[1]The authors are with School of Control Science and Engineering, Shandong University, 250002, Shandong, China, yavinliu@gmail.com, chenguiyou@sdu.edu.cn, ransong@sdu.edu.cn.
This work was supported by the National Natural Science Foundation of China under Grants 61973192.

of features with minimal information loss while significantly compressing the model;
- The BPU structure can explore long-range contextual features across multiple scales;
- The PS-VLAD architecture reduces the size of Ori-VLAD by half; we theoretically prove the equivalence between them.

II. Related Work

Early methods [11]–[13] for large-scale place recognition were based mainly on images. They are limited by the inherent drawbacks of images that may be influenced by environmental factors such as day-night changes and seasonal variations. Due to their resistance to the environmental interference of point clouds, point cloud-based methods gradually gained more attention.

A. Voxel-based Methods for Place Recognition

Initially, many methods [14], [15] attempted to convert point clouds into voxels for feature extraction. Due to the massive computational requirements, they are difficult to apply to place recognition tasks. With the introduction of the 3D sparse convolution (SP-Conv) engine [16], voxel-based methods were explored for place recognition. MinkLoc3D [17] directly used 3D SP-Conv layers to extract local features from voxels. To capture long-range contextual information, SVT-Net [18] introduced transformers to extract features at both the individual voxel and voxel group level belonging to the same geometry. However, despite using the 3D SP-Conv engine, the increased resolution still results in a significant surge in the voxel count, which consumes substantial computing resources. Hence such methods suffer from a contradiction between fine-grained geometric structures and enormous computational complexity.

B. Point-wise Methods for Place Recognition

In contrast to the voxel-based method, the point-wise method directly processes the coordinates of the points, can capture more accurate geometric features while requiring fewer computations. Once this method was proposed by PointNet [5], it quickly gained widespread application. PointNetVLAD [4] first addressed point cloud-based place recognition tasks by using PointNet and NetVLAD [6] to produce a global descriptor. Methods such as the graph-based structure [8], self-attention unit [7], and orientation encoding network [9] have since been proposed to enhance the performance of global descriptors, but the unidirectional and sequential network structure inhibits their ability to capture long-range contextual relationships. PPT-Net [10] and its successor, ERINet [19], used hierarchical networks to aggregate multi-scale features into global descriptors, but they still rely on the multi-layer perceptron as a fundamental building block, which may lead to problems such as poor generalization caused by overfitting, and a higher computational cost. EPC-Net [20] uses a grouped fully connected layer to extract the global descriptors, so as to alleviate this issue, but still does not completely address the adverse impact of fully connected layers on generalization. As mentioned earlier, most successor methods of PointNetVLAD based on the original NetVLAD approach, such as [7], [8], [10], and [20], overlook the proportional relationship between its parameters, and hence suffer from unnecessarily large models and relatively low expressivity.

III. Methodology

Fig. 2 shows the pipeline of LPS-Net. It utilizes hierarchical BPUs to extract multi-scale features. To obtain local descriptors, we use PS-VLADs to aggregate the original point cloud coordinates and the features mined by BPUs. All local descriptors are condensed into the final global descriptor by a descriptor concatenation module that consists merely of vector flattening, concatenation, and SharedConv modules, without the extensively used context gating mechanisms. We next introduce the main components of LPS-Net: SharedConv, BPU, and PS-VLAD. More details about the structure of our network can be found in Section I of the supplementary materials.

A. Parameter-shared Convolution

Parameter-shared convolution (SharedConv) can reduce the dimensionality of feature vectors with minimal information loss while significantly compressing the model.

The famous parameter-shared multi-layer perceptron (SharedMLP) in PointNet [5] is the fundamental structure of many point-wise methods (e.g., EdgeConv [21] with 1×1 convolution, the PPCNN module of EPC-Net [20], and the pyramid VLAD layer of ERINet [19]), using one multi-layer perceptron (MLP) to process all points. Since every input in SharedMLP is connected to every output, perceptron units may receive dispensable information and cause the size of the network to grow rapidly with the dimensionality of layers.

SharedConv uses a set of 1D convolutional layers, whose input and output channels are fixed to 1, to process all input features. We denote the dimensions of the kernel, input vector, and output vector of a convolutional layer as h_k, h_{in}, and h_{out}, respectively. To evenly distribute the kernel over the high-dimensional vector, we set the stride as

$$s = \max\left(\left\lceil \frac{h_{in} - h_k}{h_{out} - 1} \right\rceil, 1\right). \tag{1}$$

To obtain an output vector with the specified dimensionality, we set the padding as

$$p = \frac{s \cdot (h_{out} - 1) - (h_{in} - h_k)}{2}. \tag{2}$$

Section II of the supplementary materials further discusses SharedConv, which allows elements in low-dimensional vectors to only receive information from the corresponding elements in high-dimensional vectors, reducing their interference, and enables a considerably smaller model size than SharedMLP.

Consider a two-layer non-biased SharedConv whose respective input, output, and hidden layer dimensionality are d, o, and h. The kernel sizes for the hidden and output layers are k_h and k_o, respectively. The parameter count and

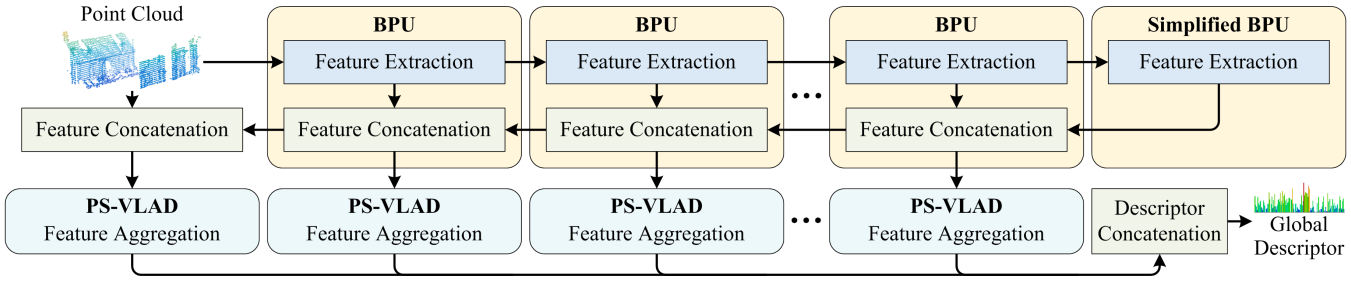

Fig. 2. Hierarchical pipeline of lightweight parameter-shared network (LPS-Net).

computational complexity of SharedConv are $k_h + k_0$ and $O(k_h \cdot h + k_o \cdot o)$, respectively, while for SharedMLP with the same layer sizes, they are $d \cdot h + h \cdot o$ and $O(d \cdot h + h \cdot o)$. When the dimensions for layers and kernels are 256 and 64, respectively, the ratio between the number of parameters of SharedConv and that of SharedMLP can be as low as 25%, and the computational complexity ratio can fall below 0.1%.

B. Bidirectional Perceptron Unit

As mentioned, the absence of a cross-layer connection makes many networks [7]–[9] weak at long-range contextual extraction and multi-scale information aggregation. To address this limitation, inspired by [10], we introduce a bidirectional perceptron unit (BPU), enabling efficient concatenation of extracted shallow features and deep contextual information.

As shown in Fig. 3, a BPU has an upper pathway consisting of downsampling and feature extraction (FE) modules, and a lower pathway with upsampling and feature concatenation (FC) modules. For the l-th BPU, the upper pathway receives a d_l^P-dimensional point set $P_l \in \mathrm{R}^{n_l \times d_l^P}$ from the $(l-1)$-th BPU, where n_l is the number of points. We apply farthest point sampling (FPS) in the upper pathway to capture skeleton points, and mine the semantic information using the FE module, which is composed of EdgeConv [21] and a grouped self-attention module [10]. More details about the upper pathway can be found in Section III of the supplementary materials. Through the upper pathway, shallow features are concentrated into a high-dimensional point set $P_{l+1} \in \mathrm{R}^{n_{l+1} \times d_{l+1}^P}$ with fewer points, i.e., $n_l > n_{l+1}$ and $d_l^P < d_{l+1}^P$.

P_{l+1} is then fed into the upper pathway of the $(l+1)$-th BPU and lower pathway of the l-th BPU. In the FC module of the lower pathway, P_{l+1} is combined with the d_{l+1}^F-dimensional feature map $F_{l+1} \in \mathrm{R}^{n_{l+1} \times d_{l+1}^F}$ generated by the $(l+1)$-th BPU, and fed to SharedConv to produce a d_l^F-dimensional feature map $F_o \in \mathrm{R}^{n_{l+1} \times d_l^F}$. Through the FC module, F_o contains both deep and shallow information, and can be aggregated into the descriptors.

Finally, to ensure that the vector merging in the lower pathway of the $(l-1)$-th BPU does not encounter issues of inconsistent quantities, F_o is upsampled to $F_l \in \mathrm{R}^{n_l \times d_l^F}$ by inverse distance weighted average interpolation (IDW) [22]. Hence the number of features in F_l and of points in P_l are both n_l. In addition, to connect the upper and lower pathways, we simplify the last BPU of LPS-Net by directly treating P_{l+1} as F_o.

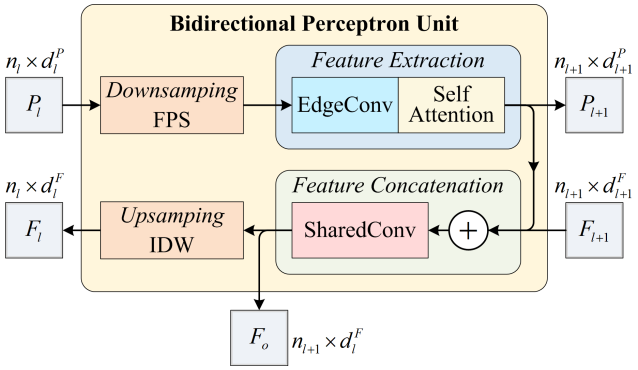

Fig. 3. Structure of l-th BPU; $\oplus$ denotes matrix concatenation.

C. Parameter-shared NetVLAD

Similar to PointNetVLAD [4], previous work, such as [8], [10] and [20], utilized the original NetVLAD [6] architecture (Ori-VLAD) for descriptor extraction. In practical implementation, we find that the trainable parameters in this type of VLAD exhibit a proportional relationship. Based on this, we propose a parameter-shared NetVLAD architecture (PS-VLAD) with half the parameters of Ori-VLAD.

PS-VLAD and Ori-VLAD both generate k sets of d-dimensional descriptors, which are denoted as $V(F) = \{V_1(F), \ldots, V_k(F) | V_k(F) \in \mathrm{R}^{1 \times d}\}$, for a d-dimensional feature map $F = \{f_1, \ldots, f_n | f_n \in \mathrm{R}^{1 \times d}\}$, where n is the number of features. To achieve this, Ori-VLAD (Fig. 2, Section IV of supplementary materials) must learn k cluster centroids, $C = \{c_1, \ldots, c_k | c_k \in \mathrm{R}^{1 \times d}\}$, and a matrix $\hat{C} = \{\hat{c}_1, \ldots, \hat{c}_k | \hat{c}_k \in \mathrm{R}^{1 \times d}\}$ with the same shape, while we only need $\hat{C}$ and a set of trainable scalars, $\Gamma = \{\gamma_1, \ldots, \gamma_k\}$, regarding $\gamma_i \hat{c}_i$ as c_i.

As shown in Fig. 4, in PS-VLAD, the soft-assignment representing the correlation between the i-th feature f_i and k-th cluster centroid c_k is expressed as

$$a_k(f_i) = \frac{\exp(\hat{c_k} f_i^T)}{\sum_{k'} \exp(\hat{c_{k'}} f_i^T)}, \tag{3}$$

which can be computed by the softmax function, and all soft-assignments are arranged into a matrix, $A \in \mathrm{R}^{k \times d}$. By multiplying the k-th row of A and the difference between all

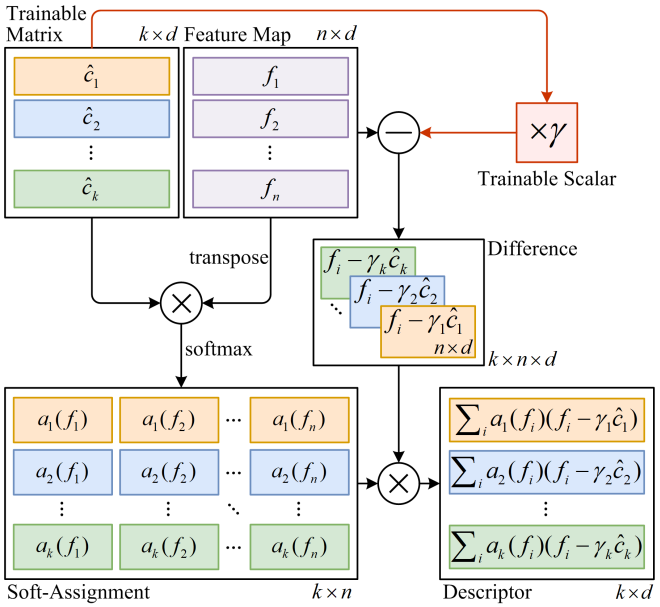

Fig. 4. Architecture of PS-VLAD; $\ominus$ and $\otimes$ denote matrix subtraction and multiplication, respectively.

features and $\gamma_k\hat{c}_k$, denoted as $D_k = \{f_1 - \gamma_k\hat{c}_k, \ldots, f_n - \gamma_k\hat{c}_k\}$, the k-th descriptor $V_k(F)$ can be expressed as

$$V_k(F) = \sum_{i=1}^{n} a_k(f_i)(f_i - \gamma_k\hat{c}_k). \tag{4}$$

More deduction processes can be found in Section IV of the supplementary materials. Next, we offer a theoretical proof regarding the equivalence between the two VLAD variants.

Lemma 1: PS-VLAD is equivalent to Ori-VLAD, as used in PointNetVLAD [4].

Proof: In Ori-VLAD, as used in PointNetVLAD [4], the k-th descriptor $V_k^{PNV}(F)$ is formulated as

$$V_k^{PNV}(F) = \sum_{i=1}^{n} a_k^{PNV}(f_i)(f_i - c_k). \tag{5}$$

Therefore, we must only prove the equivalence between eqs. (4) and (5) to establish Lemma 1.

According to PointNetVLAD, the soft-assignment of f_i with respect to the c_k is given by

$$a_k^{PNV}(f_i) = \frac{\exp(\widetilde{c}_k^T f_i - b_k)}{\sum_{k'} \exp(\widetilde{c}_{k'}^T f_i - b_{k'})}, \tag{6}$$

where $\widetilde{c}_k$ and b_k are the weights and biases, respectively, and determine the contribution of f_i to c_k. According to NetVLAD [6], the original soft-assignment between f_i and c_k is defined as

$$a_k^{NV}(f_i) = \frac{\exp(2\alpha_k c_k^T f_i - \alpha_k c_k c_k^T)}{\sum_{k'} \exp(2\alpha_k c_{k'}^T f_i - \alpha_k c_{k'} c_{k'}^T)}, \tag{7}$$

where α_k is the weight that determines the contribution of feature f_i to the k-th descriptor.

Since PointNetVLAD makes no modification to the main structure of NetVLAD, eqs. (6) and (7) should be equivalent. So, we can conclude that the weights and biases of eq. (6) can be written as $\widetilde{c}_k = 2\alpha_k c_k$ and $b_k = \alpha_k c_k c_k^T$, respectively. Considering that PointNetVLAD removes the biases b_k in practice to simplify the calculation, it is easy to obtain that eq. (6) is equivalent to eq. (3), and $\widetilde{c}_k$ is equivalent to our $\hat{c}_k$. We can clearly conclude that there exists a proportional relationship between c_k and $\hat{c}_k$. Therefore, when the trainable scalar γ_k is defined as $(2\alpha_k)^{-1}$, c_k and $\gamma_k\hat{c}_k$ become equivalent, as are eq. (4) and eq. (5). Thus, Lemma 1 is proved.

IV. Experiments

A. Datasets and Implementation

We utilized the datasets suggested for PointNetVLAD [4] to train and evaluate our approach. These include part of the Oxford RobotCar dataset [23], and point clouds from three in-house scenes: university sector (U.S.), residential area (R.A.), and business district (B.D.). Following the approach of PointNetVLAD, we considered two point clouds as a positive pair if they were at most 10 m apart according to Universal Transverse Mercator (UTM) coordinates, and as a negative pair if they were not closer than 50 m. To better evaluate the generalization capability of the model, in all experiments, we only used the data from the Oxford dataset to train the models, and used all datasets to evaluate them. We employed the average recall@1% (AR@1%) and average recall@1 (AR@1) metrics to assess place recognition accuracy.

We designed variants of LPS-Net for three application scenarios. For limited computational resources, LPS-Net-S only includes one BPU. To balance between model performance and size, LPS-Net-M includes two BPUs. For scenarios requiring exceptional performance, LPS-Net-L includes three BPUs for higher accuracy.

We train our networks on a RTX 3090 GPU with a PyTorch 1.8 [24] deep learning framework. In all three proposed networks, we integrate batch normalization [25] and ReLU [26] functions at the end of each layer, and use Adam [27] and lazy quadruplet loss [4] for training. In the loss function, we fix the hyperparameters α and β to 0.5 and 0.2, respectively. At the beginning of each training session, we initialize all trainable parameters in our networks using the default method provided by PyTorch, and set the learning rate γ to 0.0005. During the training process, γ is divided by 5 every 10 epochs, and each complete process is terminated after 30 epochs. More details about the experiments can be found in Section V of the supplementary materials.

B. Main Results

We demonstrate the effectiveness of LPS-Net through experiments. We first compare the place recognition accuracy of LPS-Net with that of state-of-the-art methods, including PointNetVLAD [4], PCAN [7], PPT-Net [10], MinkLoc3D [17], SVT-Net [18], and EPC-Net [20]. Then we provide some visual results and evaluate the computational resource

TABLE I

EVALUATION RESULTS OF DIFFERENT PLACE RECOGNITION METHODS TRAINED ON OXFORD DATASET. BEST RESULTS AMONG ALL COMPETING METHODS ARE IN BOLDFACE; BEST EXCLUDING OURS ARE UNDERSCORED. LPS-NET-S, LPS-NET-M, AND LPS-NET-L ARE OUR METHODS.

Method	Parameters	Average recall at top-1% (%)					Average recall at top-1 (%)				
		Oxford	U.S.	R.A.	B.D.	Mean	Oxford	U.S.	R.A.	B.D.	Mean
PointNetVLAD [4]	19.78M	80.9	72.7	60.8	65.3	69.9	62.6	63.2	56.1	57.2	59.8
PPT-Net [10]	13.12M	**98.1**	97.5	93.3	90.0	94.7	93.5	90.1	84.1	84.6	88.1
MinkLoc3D [17]	1.10M	97.9	95.0	91.2	88.5	93.2	**93.8**	86.0	81.1	82.7	85.9
SVT-Net [18]	0.90M	97.8	96.5	92.7	90.7	94.4	93.1	90.1	84.3	85.5	88.3
EPC-Net [20]	4.70M	94.7	96.5	88.6	84.9	91.2	86.2	88.2	80.2	78.1	83.2
EPC-Net-L-D [20]	0.41M	92.2	87.2	80.0	75.5	83.8	80.3	74.9	66.8	67.0	72.3
LPS-Net-S (Ours)	**0.09M**	96.4	97.0	92.3	89.1	93.7	89.6	89.5	83.7	84.2	86.8
LPS-Net-M (Ours)	0.29M	97.3	98.6	94.4	**92.4**	95.7	92.7	93.0	88.5	87.6	90.5
LPS-Net-L (Ours)	1.12M	97.6	**99.1**	**95.5**	92.3	**96.1**	93.4	**95.2**	**88.7**	**88.6**	**91.5**

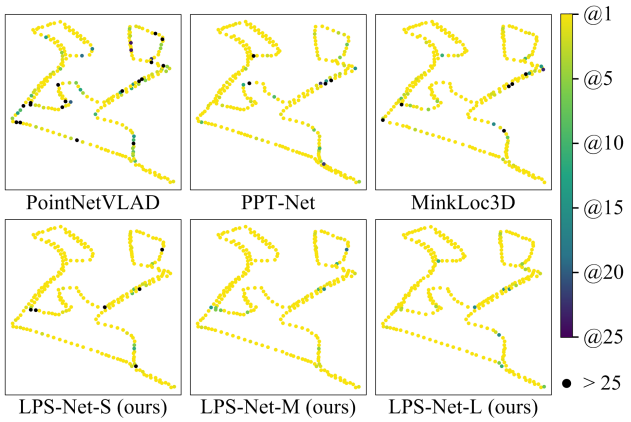

Fig. 5. Heat maps of place recognition results of different methods. Yellow denotes higher accuracy.

TABLE II

COMPUTATIONAL CONSUMPTION OF DIFFERENT METHODS.

Method	Parameters	FLOPs
PointNetVLAD [4]	19.78M	4.21G
PPT-Net [10]	13.12M	3.20G
MinkLoc3D [17]	1.10M	3.50G
SVT-Net [18]	0.90M	-
EPC-Net [20]	4.70M	3.25G
EPC-Net-L-D [20]	0.41M	1.37G
LPS-Net-S (Ours)	**0.09M**	**0.44G**
LPS-Net-M (Ours)	0.29M	0.55G
LPS-Net-L (Ours)	1.12M	0.65G

TABLE III

EVALUATION RESULTS OF DIFFERENT METHODS WHEN RANDOMLY ROTATING POINT CLOUDS FROM 0 TO $\pm$30 DEGREES.

Method	AR@1% (%)	AR@1 (%)
PointNetVLAD [4]	58.9	38.0
PPT-Net [10]	88.6	73.4
MinkLoc3D [17]	64.0	45.3
LPS-Net-S (Ours)	84.6	69.5
LPS-Net-M (Ours)	89.4	77.1
LPS-Net-L (Ours)	**91.4**	**80.2**

consumption of LPS-Net. Finally, we analyze the robustness of our method. More evaluation results can be found in Section VI of the supplementary materials.

1) Place Recognition Accuracy: We provide the AR@1% and AR@1 metrics for various place recognition methods trained on the Oxford dataset. Table I shows that different versions of LPS-Net outperform state-of-the-art methods.

Because the BPU is integrated with SharedConv, LPS-Net can fuse more deep contextual information into shallow semantic features, so as to outperform PPT-Net with a similar hierarchical structure. Notably, on the AR@1 metric on the U.S. dataset, LPS-Net-L achieves a 5.1% improvement over PPT-Net. Despite the higher accuracy of voxel-based methods such as SVT-Net and MinkLoc3D on the Oxford dataset, they perform significantly poorer than our method on other datasets. LPS-Net demonstrates better generalization capability, and its accuracy on the mean AR@1 metric across the four datasets reaches 90.5%, showing a 2.2% improvement over SVT-Net. EPC-Net-L-D is a lightweight version trained with knowledge distillation based on a larger network, EPC-Net. Its fully connected structure has a lower generalization capability than the proposed SharedConv, and the mean AR@1% of LPS-Net-S without knowledge distillation is approximately 10% higher than that of EPC-Net-L-D even at one-fifth of the size. Such results demonstrate that SharedConvs and PS-VLADs significantly simplify the network and effectively mitigate overfitting.

2) Visual Results: We use heat maps to show the place recognition results of different methods. In Fig. 5, each dot represents a randomly selected scene from the B.D. dataset for retrieval, and the relative positions of dots correspond to the actual geographical locations of corresponding scenes. Yellow represents higher accuracy, and it can be observed that LPS-Net outperforms other networks, especially LPS-Net-M and LPS-Net-L.

3) Computational Consumption: Table II reports the computational resource consumption of different methods. LPS-Net-S and LPS-Net-M have the fewest parameters and the lowest consumption of computational resources. This demon-

TABLE IV

ABLATION STUDIES OF DIFFERENT COMPONENTS IN LPS-NET; ROW E: VANILLA LPS-NET-L.

Model	Parameters	Oxford	U.S.	R.A.	B.D.	Mean
A: LPS-Net with SharedMLP	7.74M	97.4	97.4	89.7	85.6	92.5
B: LPS-Net with Ori-VLAD	1.14M	96.9	98.6	93.6	90.8	95.0
C: LPS-Net w/o raw geometric features	**1.10M**	93.5	96.5	91.1	85.5	91.7
D: LPS-Net with context gating mechanism	1.18M	96.7	97.6	93.1	89.9	94.3
E: LPS-Net	1.12M	**97.6**	**99.1**	**95.5**	**92.3**	**96.1**

strates that the use of SharedConv and PS-VLAD simplifies LPS-Net.

4) Robustness Analysis: To more rigorously evaluate LPS-Net, we investigate the impact of point cloud rotation on the performance of different methods. Table III displays AR@1% and AR@1 for different methods on the Oxford dataset when randomly rotating point clouds from 0 to ± 30 degrees. It can be observed that, compared to previous methods, all three versions of LPS-Net are more robust to randomly rotated point clouds.

C. Ablation Studies

We demonstrate the necessity of the key components of LPS-Net. Since its three versions have a similar basic structure, we used LPS-Net-L and its variants in ablation studies. All models were evaluated using the AR@1% metric.

1) Dimensionality Reduction Methods: We investigated the necessity of the SharedConv structure by replacing it with SharedMLP. From row A of Table IV, it can be seen that the size of the version using SharedConv is approximately 14% that of the version using SharedMLP, and its accuracy is better. This demonstrates that our SharedConv structure improves the performance of the method.

2) Different Sizes of Kernels in SharedConvs: We notice that the ratio λ between kernel sizes and dimensions of the input features is crucial for the performance of SharedConv in a BPU. We changed λ, and show the results in Table V. It can be found that when λ is set to approximately 25%, SharedConv demonstrates optimal performance. Excessively large kernels do not improve the accuracy and lead to an unnecessary increase in computational complexity.

3) Different Structures of VLAD: We investigated the impact of different VLAD structures on network performance by replacing PS-VLAD with Ori-VLAD. The results are shown in row B of Table IV. It can be observed that PS-VLAD indeed reduces the number of parameters, and improves the generalization capability of the model.

4) Ways to Generate Global Descriptors: As shown in Fig. 2, LPS-Net uses a separate PS-VLAD to directly extract raw geometric features. Referring to row C of Table IV, it can be observed that although the model excluding raw geometric features has a slightly reduced size, the significant decline in performance highlights the necessity of raw geometric features for LPS-Net. Furthermore, as shown in row D of Table IV, when the context gating mechanism is added at the descriptor concatenation module, LPS-Net experiences a significant decrease in accuracy and generalization with an increase in size, indicating that the gate structure is unnecessary for LPS-Net.

TABLE V

ABLATION STUDIES OF DIFFERENT KERNEL SIZES IN SHAREDCONV.

Kernel size	FLOPs	Oxford	U.S.	R.A.	B.D.
$\lambda = 12.5\%$	**0.53G**	96.3	98.8	94.2	89.1
$\lambda = 25.0\%$	0.65G	**97.6**	**99.0**	**94.7**	**92.0**
$\lambda = 37.5\%$	0.78G	96.5	99.0	93.0	90.9
$\lambda = 50.0\%$	0.91G	84.1	73.1	73.4	76.2

TABLE VI

ABLATION STUDIES OF DIFFERENT NUMBERS n OF BPUS.

BPU	Parameters	Oxford	U.S.	R.A.	B.D.
$n = 1$	**0.09M**	96.4	97.0	92.3	89.1
$n = 2$	0.29M	97.3	98.6	94.4	**92.4**
$n = 3$	1.12M	**97.6**	**99.1**	**95.5**	92.3
$n = 4$	4.79M	96.2	98.0	92.6	90.2
$n = 5$	21.65M	95.0	97.7	92.6	88.7

5) Different Numbers of BPUs: LPS-Net uses hierarchical BPUs to progressively extract contextual information at different scales. We investigated the impact of different numbers of BPUs on performance, and report the results in Table VI. In this experiment, all models included a simplified BPU to ensure connectivity between the upper and lower pathways. It can be observed that an excessive number of BPUs leads to a rapid increase in size and a decrease in accuracy. Therefore, we used 1, 2, and 3 BPUs to construct the three types of LPS-Net, achieving a delicate balance between model performance and size.

V. CONCLUSION AND FUTURE WORK

We proposed a lightweight parameter-shared network (LPS-Net) for point cloud-based place recognition. We utilize multiple BPUs to extract features at different scales and PS-VLADs to aggregate the descriptors. By incorporating SharedConv, which replaces fully-connected layers with convolutional layers, in the BPU and replacing half the parameters in the original NetVLAD with a trainable scalar in PS-VLAD, we significantly reduced the model size. LPS-Net achieved state-of-the-art performance in experiments while effectively compressing the model. In the future, we plan to integrate point clouds with RGB to further enhance the robustness of our method.

REFERENCES

[1] Cesar Cadena, Luca Carlone, Henry Carrillo, Yasir Latif, Davide Scaramuzza, José Neira, Ian Reid, and John J Leonard. Past, present, and future of simultaneous localization and mapping: Toward the robust-perception age. *IEEE Transactions on Robotics*, 32(6):1309–1332, 2016.

[2] Liyuan Pan, Yuchao Dai, Miaomiao Liu, and Fatih Porikli. Simultaneous stereo video deblurring and scene flow estimation. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 4382–4391, 2017.

[3] 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.

[4] Mikaela Angelina Uy and Gim Hee Lee. Pointnetvlad: Deep point cloud based retrieval for large-scale place recognition. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 4470–4479, 2018.

[5] Charles R Qi, Hao Su, Kaichun Mo, and Leonidas J Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 652–660, 2017.

[6] Relja Arandjelovic, Petr Gronat, Akihiko Torii, Tomas Pajdla, and Josef Sivic. Netvlad: Cnn architecture for weakly supervised place recognition. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 5297–5307, 2016.

[7] Wenxiao Zhang and Chunxia Xiao. Pcan: 3d attention map learning using contextual information for point cloud based retrieval. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12436–12445, 2019.

[8] 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 *IEEE/CVF International Conference on Computer Vision*, pages 2831–2840, 2019.

[9] 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 *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11348–11357, 2021.

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

[11] Edward Johns and Guang-Zhong Yang. From images to scenes: Compressing an image cluster into a single scene model for place recognition. In *IEEE International Conference on Computer Vision*, pages 874–881, 2011.

[12] Dorian Gálvez-López and Juan D Tardos. Bags of binary words for fast place recognition in image sequences. *IEEE Transactions on Robotics*, 28(5):1188–1197, 2012.

[13] Eduardo Fernández-Moral, Walterio Mayol-Cuevas, Vicente Arevalo, and Javier Gonzalez-Jimenez. Fast place recognition with plane-based maps. In *IEEE International Conference on Robotics and Automation*, pages 2719–2724, 2013.

[14] Zhirong Wu, Shuran Song, Aditya Khosla, Fisher Yu, Linguang Zhang, Xiaoou Tang, and Jianxiong Xiao. 3d shapenets: A deep representation for volumetric shapes. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1912–1920, 2015.

[15] Gernot Riegler, Ali Osman Ulusoy, and Andreas Geiger. Octnet: Learning deep 3d representations at high resolutions. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 3577–3586, 2017.

[16] Christopher Choy, JunYoung Gwak, and Silvio Savarese. 4d spatio-temporal convnets: Minkowski convolutional neural networks. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3075–3084, 2019.

[17] Jacek Komorowski. Minkloc3d: Point cloud based large-scale place recognition. In *IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 1790–1799, 2021.

[18] 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 Conference on Artificial Intelligence*, volume 36, pages 551–560, 2022.

[19] Shichen Weng, Ruonan Zhang, and Ge Li. Erinet: Effective rotation invariant network for point cloud based place recognition. In *IEEE International Conference on Visual Communications and Image Processing*, pages 1–5, 2022.

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

[21] 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 Transactions on Graphics*, 38(5):1–12, 2019.

[22] Charles Ruizhongtai Qi, Li Yi, Hao Su, and Leonidas J Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. *Advances in Neural Information Processing Systems*, 30, 2017.

[23] 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.

[24] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. *Advances in Neural Information Processing Systems*, 32, 2019.

[25] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International Conference on Machine Learning*, pages 448–456, 2015.

[26] Bing Xu, Naiyan Wang, Tianqi Chen, and Mu Li. Empirical evaluation of rectified activations in convolutional network. arXiv:1505.00853, 2015.

[27] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. arXiv:1412.6980, 2014.