Supplementary Materials

This supplement shows further technical details and evaluation results.

Section I discusses the structure of LPS-Net, and Sections II–IV provide technical details of SharedConv, BPU, and PS-VLAD, respectively. Section V provides details of our experiments, and Section VI discusses the results.

I. Network Architecture

We discuss the architecture of LPS-Net, whose pipeline is illustrated in Fig. 2 of the main text. Multiple hierarchical BPUs are integrated to extract multi-scale features. The BPU integrated with SharedConv can generate point-wise features that combine features at different scales. A simplified BPU without any output is added at the end of the network to ensure that the upper and lower pathways in the BPU are interconnected.

Each complete BPU is equipped with a PS-VLAD to generalize descriptors at different scales. To extract raw geometric relationships, the features exported by the topmost BPU and the original input point cloud are merged into another PS-VLAD via an FC module.

Finally, all descriptors generated by the PS-VLADs are flattened and concatenated into the final global descriptor by a descriptor concatenation module. As shown in Fig. 1, since the global descriptors extracted by LPS-Net are already sufficiently refined, the descriptor concatenation module only consists of vector flattening, concatenation, and SharedConv modules, without the extensively used context gating mechanisms.

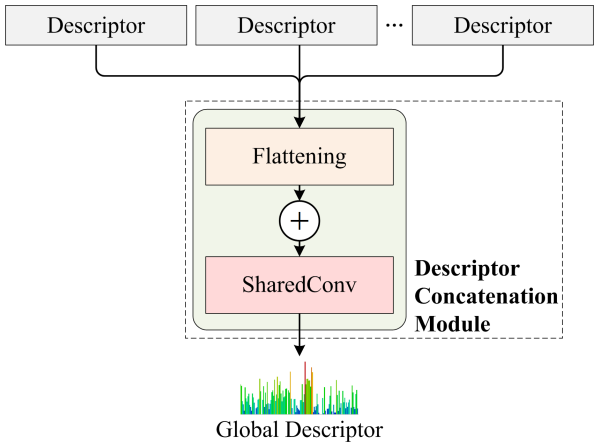


Fig. 1. Structure of descriptor concatenation module; $\oplus$ denotes matrix concatenation.

II. SharedConv

We further discuss SharedConv. During its construction, we match the input dimensionality h_{in}, output dimensionality h_{out}, and kernel size h_k by selecting appropriate padding p and stride s. However, as shown in Eq. (2) in the main text, when the value of the kernel size h_k is not appropriate, p may not be an integer, in which case, both rounding up and rounding down will lead to dimensionality errors, i.e., the convolution layer will be unable to generate an output vector with the specified dimensionality. Therefore, when p is not an integer, we increase the dimensionality of the convolution kernel by 1.

III. BPU

We further discuss the upper pathway of BPU, which is composed of a downsampling and FE module, where the latter consists of EdgeConv and grouped self-attention modules. To better integrate the shallow geometric information with the deep semantic information, in the l-th BPU, the d_l^P-dimensional input point set P_l is concatenated with the corresponding coordinates after downsampling, resulting in a $(d_l^P + 3)$-dimensional point set. In EdgeConv, we do not concatenate nodes and edges, but directly subtract them to reduce the computational complexity. Inspired by the state-of-the-art PPT-Net [1], downsampling (FPS) and EdgeConv are both implemented in the coordinate space.

IV. PS-VLAD

We show the structure of Ori-VLAD in Fig. 2, and elaborate upon the computational process of Eq. (4) in the main paper. We first compute a $(k \times n)$-dimensional soft-assignment matrix $A = \{A_1, \ldots, A_k | A_k \in \mathrm{R}^{1\times n}\}$,

$$A = softmax(\hat{C} \cdot F^T, dim = 0), \tag{1}$$

where $softmax(\cdot, dim = 0)$ signifies that the softmax function is calculated along each column. The k-th n-dimensional row $A_k = \{a_k(f_1), \ldots, a_k(f_n)\}$ captures the soft-assignments of all features in relation to c_k, and the i-th soft-assignment $a_k(f_i)$ of A_k refers to Eq. (3) in the main text.

Subsequently, we calculate the difference $D \in \mathrm{R}^{k\times n\times d}$ between every cluster centroid and all features, denoted by $D = \{D_1, \ldots, D_k | D_k \in \mathrm{R}^{n\times d}\}$. Here, we consider the k-th cluster centroid matrix c_k and the k-th trainable matrices $\hat{c}_k$ to be proportionally correlated, which can be formulated as

$$c_k = \gamma_k \cdot \hat{c}_k. \tag{2}$$

Thus, the difference between all features and c_k can be expressed as $D_k = \{f_1 - \gamma_k \hat{c}_k, \ldots, f_n - \gamma_k \hat{c}_k\}$.

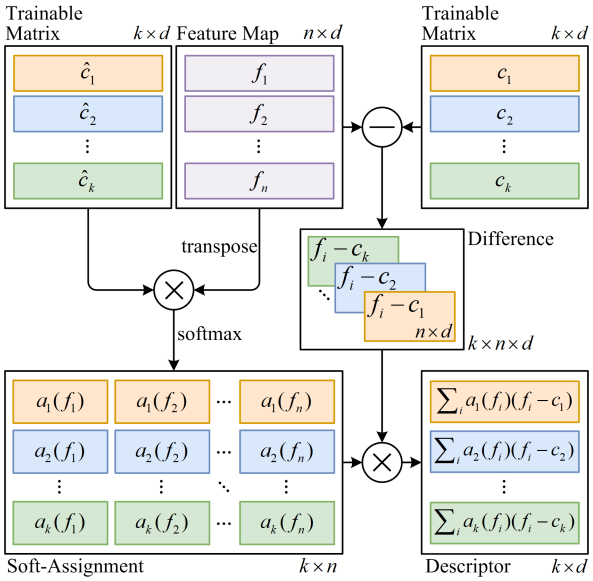


Fig. 2. Architecture of Ori-VLAD; $\ominus$ and $\otimes$ represent matrix subtraction and multiplication, respectively.

Finally, by multiplying the soft-assignments A_k and the difference D_k of the same cluster centroid, the k-th descriptor $V_k(F)$ is formulated as

$$V_k(F) = A_k \cdot D_k. \tag{3}$$

Thus, Eq. (3) is equivalent to Eq. (4) in the main text.

V. Experiments

We proposed three variants of LPS-Net with the same basic structure: LPS-Net-S is extremely lightweight, LPS-Net-M balances accuracy and size, and LPS-Net-L focuses on performance. All three models take point clouds of size 4096×3 as input, and generate a 256-dimensional global descriptor. To balance computational complexity and performance, in the FE module of each BPU, we use 20 nearest neighbors to form the graph in the EdgeConv module, and set the number of groups to 8 in the grouped self-attention module.

Fig. 3 shows the structure of LPS-Net-L, which consists of three complete BPUs and a simplified BPU. Based on LPS-Net-L, we remove the fourth simplified BPU, and simplify the third BPU to create LPS-Net-M. Similarly, the most lightweight version, LPS-Net-S, is based on LPS-Net-M, removing the third BPU and modifying the second BPU to the simplified version. While removing the BPU, the accompanying PS-VLAD is also eliminated, resulting in a corresponding reduction in the total number of descriptors. Therefore, in the descriptor concatenation modules of LPS-Net-M and LPS-Net-L, the input vector dimensionalities for SharedConvs are 21504 and 20480, respectively. However, their kernel sizes remain at 512.

VI. More Evaluation Results

We report more evaluation results on LPS-Net from both quantitative and qualitative perspectives. As in Section IV-C, unless otherwise specified, the experiments in this section still use LPS-Net-L as a representative of LPS-Net.

A. Quantitative Results

To provide a more objective evaluation of LPS-Net, Table I compares the place recognition accuracy of LPS-Net and more previous state-of-the-art methods, including LPD-Net [2], PCAN [3], and the methods given in Table 1 of the main text. We compare their computational consumption in Table II. As can be seen, three variants of LPS-Net still lead other methods by a wide margin in terms of recognition accuracy, model size, and computational complexity. Fig. 4 shows the recall curves of different methods for the top 25 retrieval results on the four datasets. The outstanding performance on three in-house datasets (U.S., R.A., and B.D.) further demonstrates the excellent generalization capability of LPS-Net in completely unseen environments.

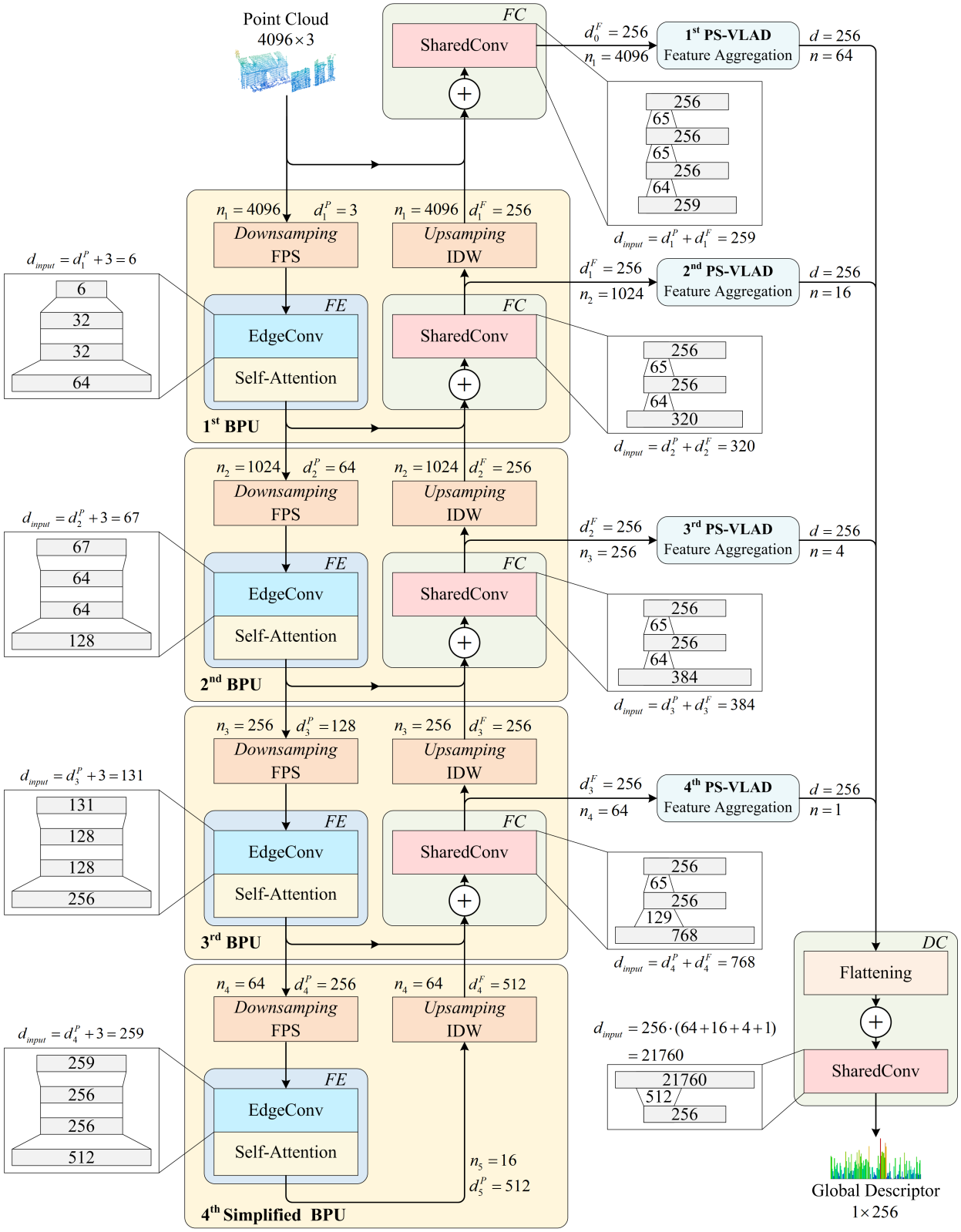
B. Qualitative Results

To better evaluate LPS-Net, we visualize some top-k matching results in Fig. 5. It can be observed that LPS-Net can accurately recall the most matching scenes in various environments. Fig. 6 compares the top-1 recall results of different methods, from which it can be seen that, compared to other methods, LPS-Net can better distinguish scenes with high repetition (such as intersections in lines 1 and 2, and trees in lines 3 and 4) and scenes with fewer distinct features (such as streets in lines 5 and 6, and walls in lines 7 and 8).

To more intuitively demonstrate the stability of our method, Fig. 7 presents the top-1 recall results on different subsets of the Oxford dataset for the same query scene. To demonstrate the changes in the environment, we present images corresponding to the point clouds. It can be seen that, due to factors such as tree growth and traffic conditions, point clouds collected at the same location at different times are not completely identical. LPS-Net can still accurately identify these scenes, highlighting the stability of our method.

VII. Acknowledgments

We thank LetPub (www.letpub.com) for its linguistic assistance during the preparation of this manuscript.



FE: Feature Extraction **FC**: Feature Concatenation **DC**: Descriptor Concatenation
SharedConv: Parameter-Shared Convolution **BPU**: Bidirectional Perceptron Unit **PS-VLAD**: Parameter-Shared NetVLAD

Fig. 3. Structure of LPS-Net-L; $\oplus$ denotes concatenation.

TABLE I

EVALUATION RESULTS OF PLACE RECOGNITION METHODS TRAINED ON OXFORD DATASET. BEST RESULTS ARE IN BOLDFACE; BEST EXCLUDING OURS ARE UNDERSCORED.

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
LPD-Net [2]	19.81M	94.9	96.0	90.4	89.1	92.6	86.3	87.0	83.0	82.3	84.7
PCAN [3]	20.42M	83.9	79.1	71.2	66.8	75.3	69.4	62.4	56.9	58.1	61.7
PPT-Net [1]	13.12M	**98.1**	97.5	93.3	90.0	94.7	93.5	90.1	84.1	84.6	88.1
MinkLoc3D [5]	1.10M	97.9	95.0	91.2	88.5	93.2	**93.8**	86.0	81.1	82.7	85.9
SVT-Net [6]	0.90M	97.8	96.5	92.7	90.7	94.4	93.1	90.1	84.3	85.5	88.3
EPC-Net [7]	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 [7]	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**

TABLE II

COMPUTATIONAL CONSUMPTION OF OTHER METHODS.

Method	Parameters	FLOPs
PointNetVLAD [4]	19.78M	4.21G
LPD-Net [2]	19.81M	7.80G
PCAN [3]	20.42M	7.73G
PPT-Net [1]	13.12M	3.20G
MinkLoc3D [5]	1.10M	3.50G
SVT-Net [6]	0.90M	-
EPC-Net [7]	4.70M	3.25G
EPC-Net-L-D [7]	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

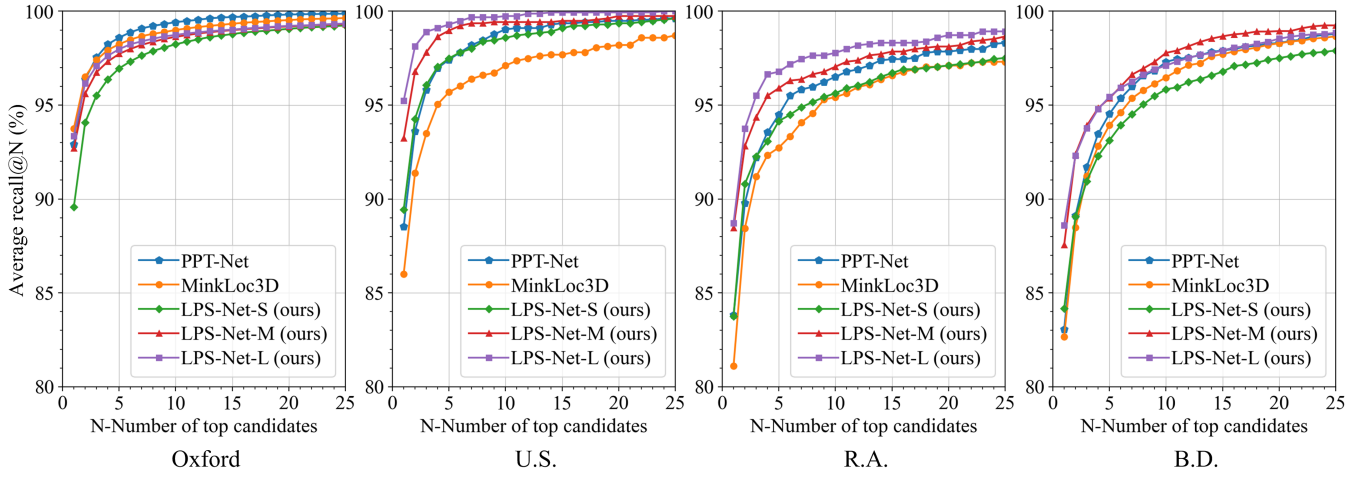


Fig. 4. Average recall curves of different methods trained on Oxford dataset. Outstanding performance on U.S., R.A., and B.D. datasets further demonstrates generalization capability of LPS-Net in completely unseen environments.

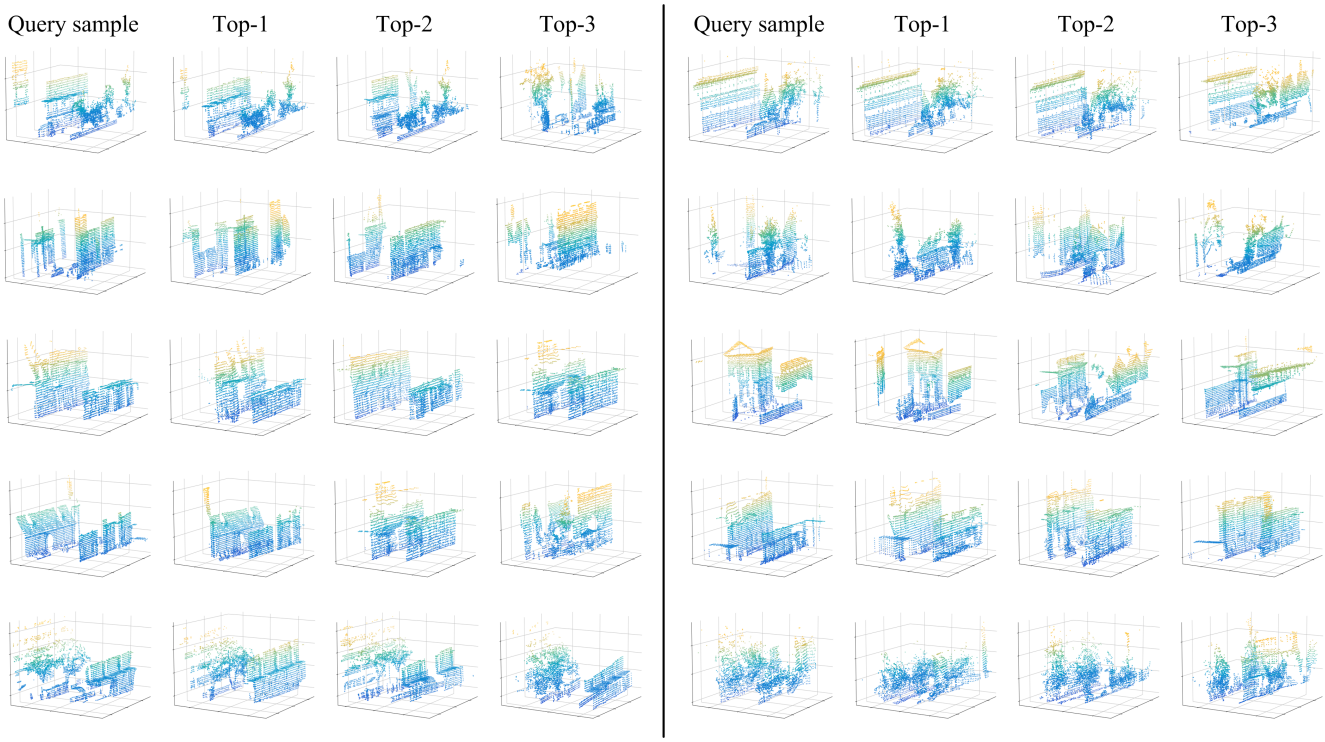


Fig. 5. Some top-3 recall results of LPS-Net corresponding to different query samples. It can be observed that LPS-Net can accurately recall most matching scenes in various environments.

Fig. 6. More top-1 matching results of different methods. Green box: correct result; red boxes: incorrect results.

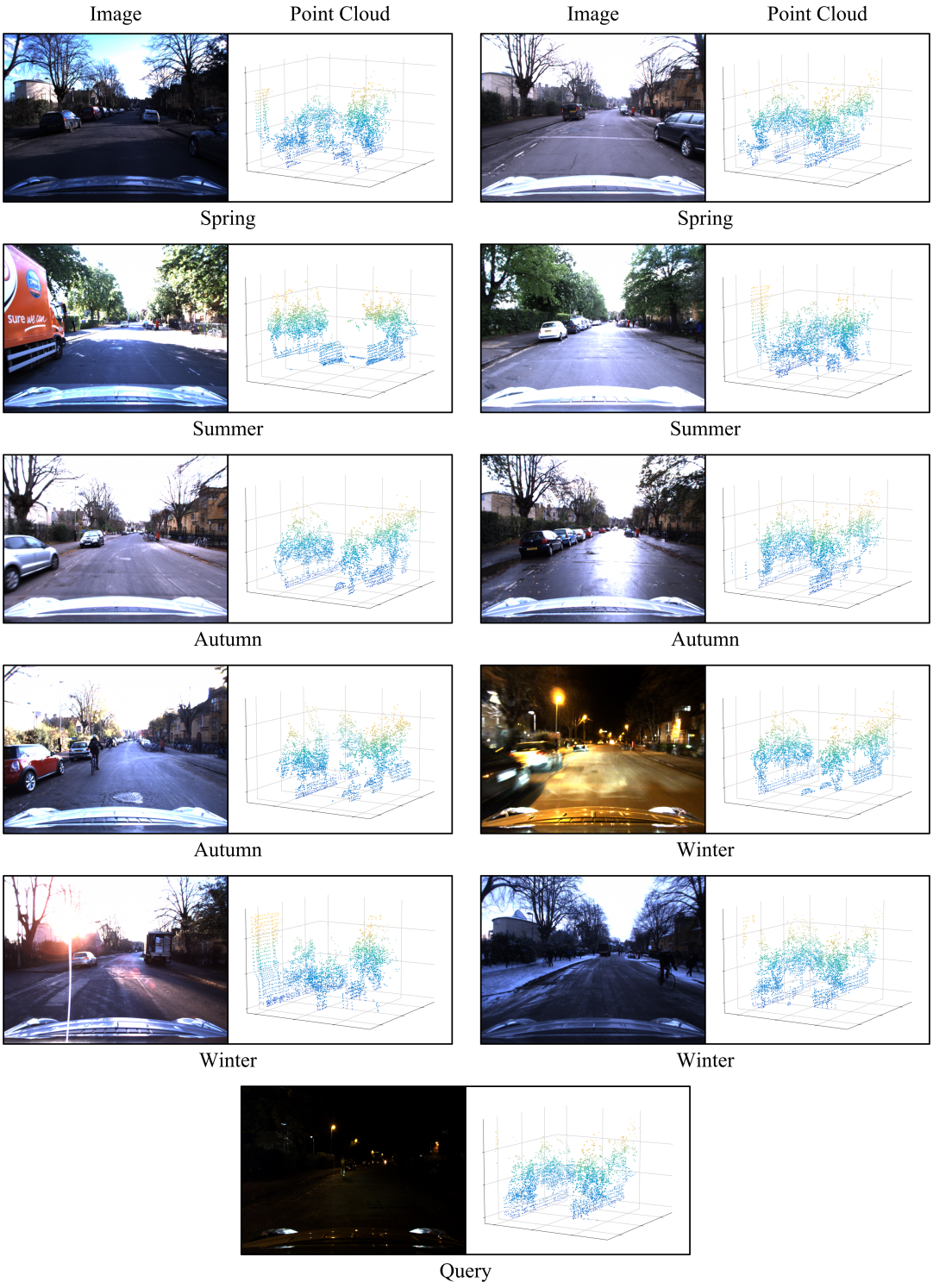


Fig. 7. Top-1 recall results of LPS-Net for same query scene.

REFERENCES

[1] 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.

[2] 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.

[3] 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.

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

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

[7] 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.