Supplementary Materials

This document serves as a supplementary resource to the main paper, offering further technical details and evaluation results.

Firstly, in Sec. A, Sec. B, Sec. C, and Sec. D we provide additional technical details about SharedConv, BPU, PS-VLAD, and descriptor concatenation module, respectively. Then, in Sec. E we provide additional details about the experiment. Finally, in Sec. F, we report more comprehensive experimental results.

A Details on SharedConv

Here, we showcase more details about SharedConv. During the construction of SharedConv, 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 body of paper, when the value of the kernel size h_k is not appropriate, it is possible that p will not be an integer. In this case, both rounding up or rounding down will lead to dimensionality errors, i.e., the convolution layer is 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 to avoid this issue.

B Details on BPU

In this section, we introduce more details about the upper pathway of BPU. The upper pathway is composed of downsampling and FE modules, while the FE module consists of EdgeConv and grouped self-attention modules. In order 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 being downsampled, 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 (Hui et al. 2021), the downsampling algorithm (FPS) and EdgeConv are both implemented in the coordinate space.

C Details on PS-VLAD

In this section, we supplement the structure of Ori-VLAD in Fig. 1. Meanwhile, we elaborate on the computational

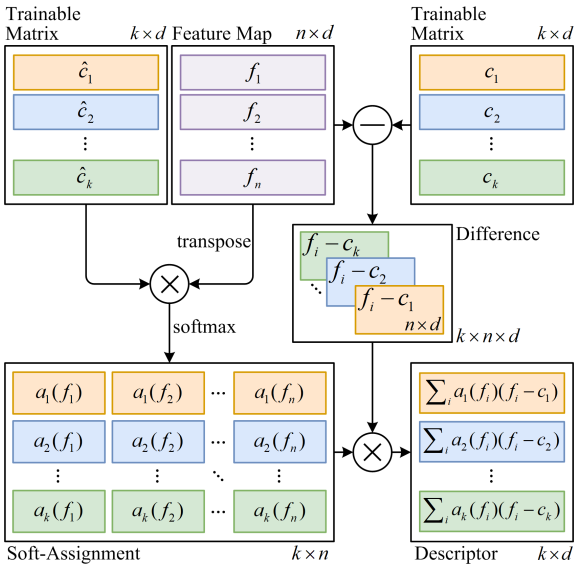


Figure 1: The architecture of Ori-VLAD. The $\ominus$ and $\otimes$ represent the matrix subtraction and multiplication operators.

process of the Eq. (6) in the main paper. Specifically, we first compute a $(k \times n)$-dimensional soft-assignment matrix $A = \{A_1, \ldots, A_k | A_k \in \mathrm{R}^{1\times n}\}$ through:

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

where $softmax(\cdot, dim = 0)$ signifies that the soft-max 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 the c_k, and the i-th soft-assignment $a_k(f_i)$ of A_k refers to Eq. (5) in the main paper.

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 cluster centroid matrix C and the trainable matrices $\hat{C}$ to be proportionally correlated, which can be formulated as:

$$C = \gamma \cdot \hat{C}. \tag{2}$$

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

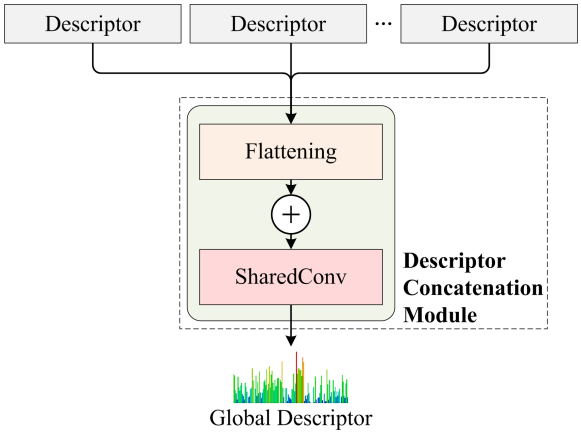


Figure 2: The structure of descriptor concatenation module. The $\oplus$ represents the matrix concatenation operator.

Finally, by multiplying the soft-assignments A_k and the difference D_k regarding 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. (6) in the main paper.

D Details on the Descriptor Concatenation Module

In this section, we supplement the structure of the descriptor concatenation module in Fig. 2.

E Details on the Implementation of Experiments

In this paper, we propose three different variants of LPS-Net, namely LPS-Net-S for extreme lightweight, LPS-Net-M for balancing accuracy and size, and LPS-Net-L for focusing on performance. The basic structure of these variants remains the same. All three models take point clouds of size 4096×3 as input and generate a 256-dimensional global descriptor. To balance the computational complexity and the model performance, in the FE module of each BPU, we use 20 nearest neighbors to form the graph in EdgeConv module and set the number of groups to 8 in the grouped self-attention module.

As shown in Fig. 3, we present the detailed structure of LPS-Net-L. It consists of three complete BPUs and a simplified BPU. Based on LPS-Net-L, we removed the fourth simplified BPU and simplified the third BPU to create LPS-Net-M. Similarly, the most lightweight version, LPS-Net-S, is based on LPS-Net-M but removes the third BPU and modifies 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 generated by that. 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 512.

F More Evaluation Results

In this section, we report more evaluation results on LPS-Net from both quantitative and qualitative perspectives. Like Sec. 4.3, unless otherwise specified, the experiments in this section still use LPS-Net-L as a representative of LPS-Net.

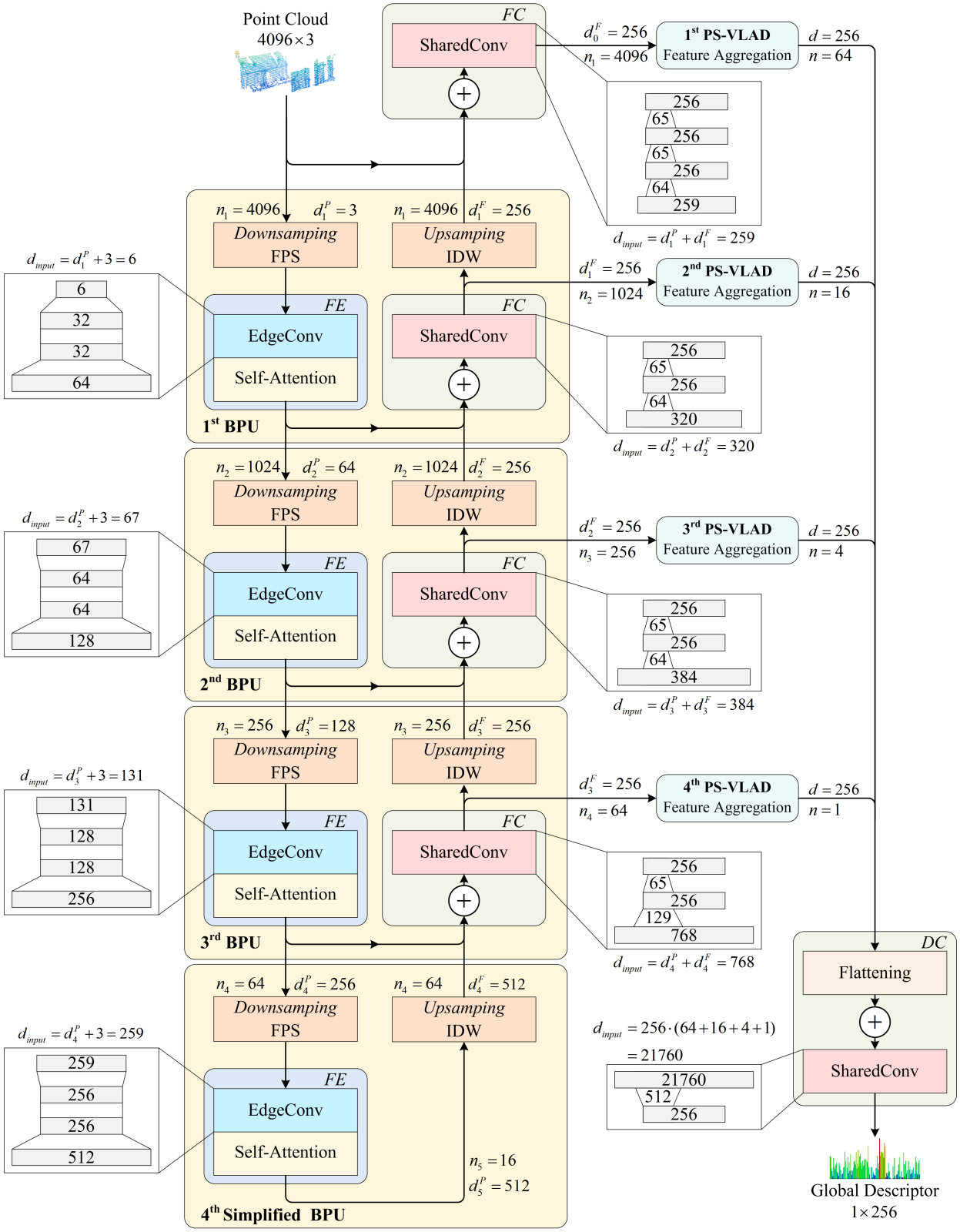
Quantitative Results

In order to provide a more objective evaluation of LPS-Net, in Tab. 1, we present a comparison of place recognition accuracy between LPS-Net and more previous state-of-the-art methods, including LPD-Net (Liu et al. 2019), PCAN (Zhang and Xiao 2019), and the methods mentioned in Tab. 1 of the main paper. We also report a comparison of computational consumption between them in Tab. 2. As can be seen, three variants of LPS-Net still lead by a wide margin in terms of recognition accuracy, model size, and computational complexity compared to other methods. As shown in Fig. 4, we also show the recall curves of different methods for top 25 retrieval results on the four datasets. The outstanding performance on three in-house datasets (U.S., R.A. and B.D. datasets) further demonstrates the excellent generalization capability of LPS-Net in the completely unseen environments.

Qualitative Results

In order to better demonstrate the superiority, we visualize some top-k matching results of LPS-Net in Fig. 5. It can be observed that LPS-Net is capable of accurately recalling the most matching scenes in various environments. In Fig. 6, we present more examples comparing the Top 1 recall results between different methods. It can be seen that, compared to other methods, our LPS-Net is able to 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).

In addition, to demonstrate the stability of our method more intuitively, we present in Fig. 7 the top-1 recall results on the different subsets of the Oxford dataset for the same query scene. In order to demonstrate the changes in the environment, we also 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. However, LPS-Net is still able to accurately identify these scenes, highlighting the stability of our method.



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

Figure 3: the detailed structure of LPS-Net-L. The $\oplus$ represents the concatenation operator. LPS-Net-L consists of three complete BPUs and a simplified BPU. Based on LPS-Net-L, we removed the fourth simplified BPU and simplified the third BPU to create LPS-Net-M. Similarly, the most lightweight version, LPS-Net-S, is based on LPS-Net-M but removes the third BPU and modifies the second BPU to the simplified version.

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 (Uy and Lee 2018)	19.78M	80.9	72.7	60.8	65.3	69.9	62.6	63.2	56.1	57.2	59.8
LPD-Net (Liu et al. 2019)	19.81M	94.9	96.0	90.4	89.1	92.6	86.3	87.0	83.0	82.3	84.7
PCAN (Zhang and Xiao 2019)	20.42M	83.9	79.1	71.2	66.8	75.3	69.4	62.4	56.9	58.1	61.7
PPT-Net (Hui et al. 2021)	13.12M	**98.1**	97.5	93.3	90.0	94.7	93.5	90.1	84.1	84.6	88.1
MinkLoc3D (Komorowski 2021)	1.10M	97.9	95.0	91.2	88.5	93.2	**93.8**	86.0	81.1	82.7	85.9
SVT-Net (Fan et al. 2022)	0.90M	97.8	96.5	92.7	90.7	94.4	93.1	90.1	84.3	85.5	88.3
EPC-Net (Hui et al. 2022)	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 (Hui et al. 2022)	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 1: Evaluation results of more place recognition methods trained on the Oxford dataset. We bold the best results among all competing methods and underscore the best excluding ours.

Method	Parameters	FLOPs
PointNetVLAD (Uy and Lee 2018)	19.78M	4.21G
LPD-Net (Liu et al. 2019)	19.81M	7.80G
PCAN (Zhang and Xiao 2019)	20.42M	7.73G
PPT-Net (Hui et al. 2021)	13.12M	3.20G
MinkLoc3D (Komorowski 2021)	1.10M	3.50G
SVT-Net (Fan et al. 2022)	0.90M	-
EPC-Net (Hui et al. 2022)	4.70M	3.25G
EPC-Net-L-D (Hui et al. 2022)	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 2: Computational consumption of more methods.

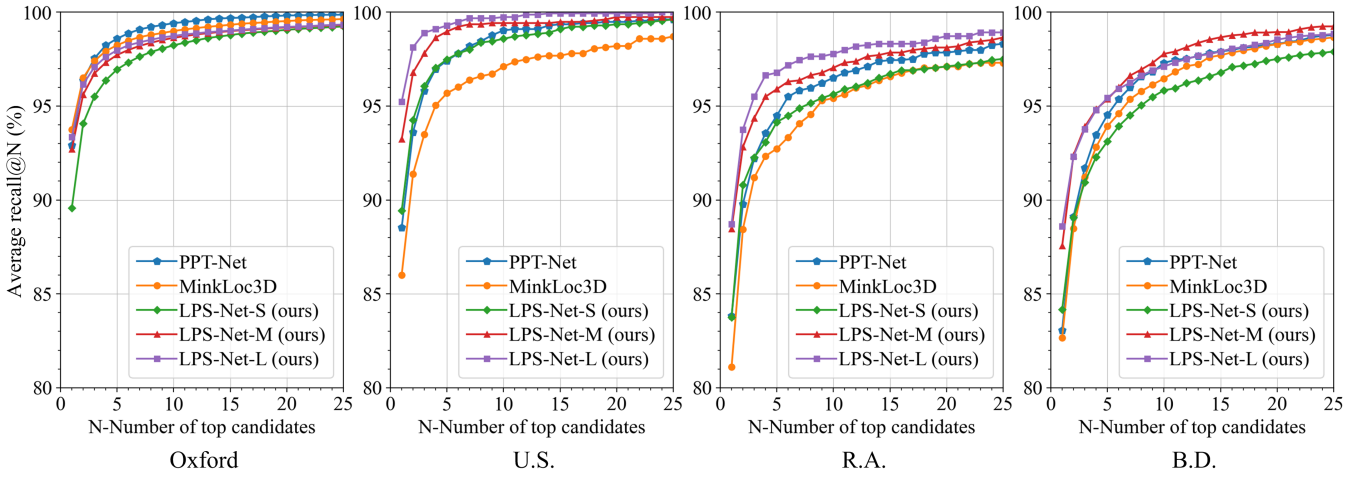


Figure 4: Average recall curves of different methods trained on the Oxford dataset. The outstanding performance on three in-house datasets (U.S., R.A. and B.D. datasets) further demonstrates the excellent generalization capability of LPS-Net in the completely unseen environments.

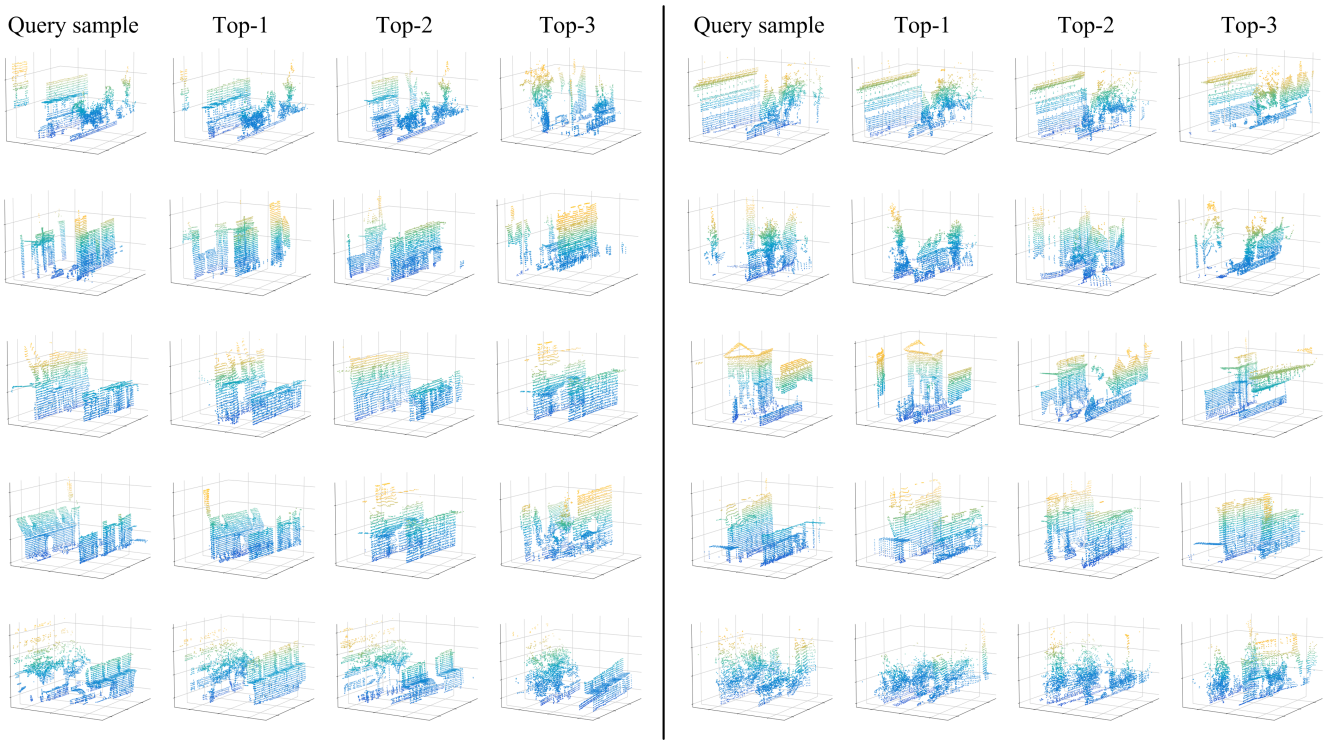


Figure 5: Some top-3 recall results of LPS-Net corresponding to different query samples. It can be observed that LPS-Net is capable of accurately recalling the most matching scenes in various environments.

Figure 6: More top-1 matching results of different methods. The green box represents the correct result, while the red boxes represent the incorrect results. Compared to other methods, our LPS-Net is able to 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).

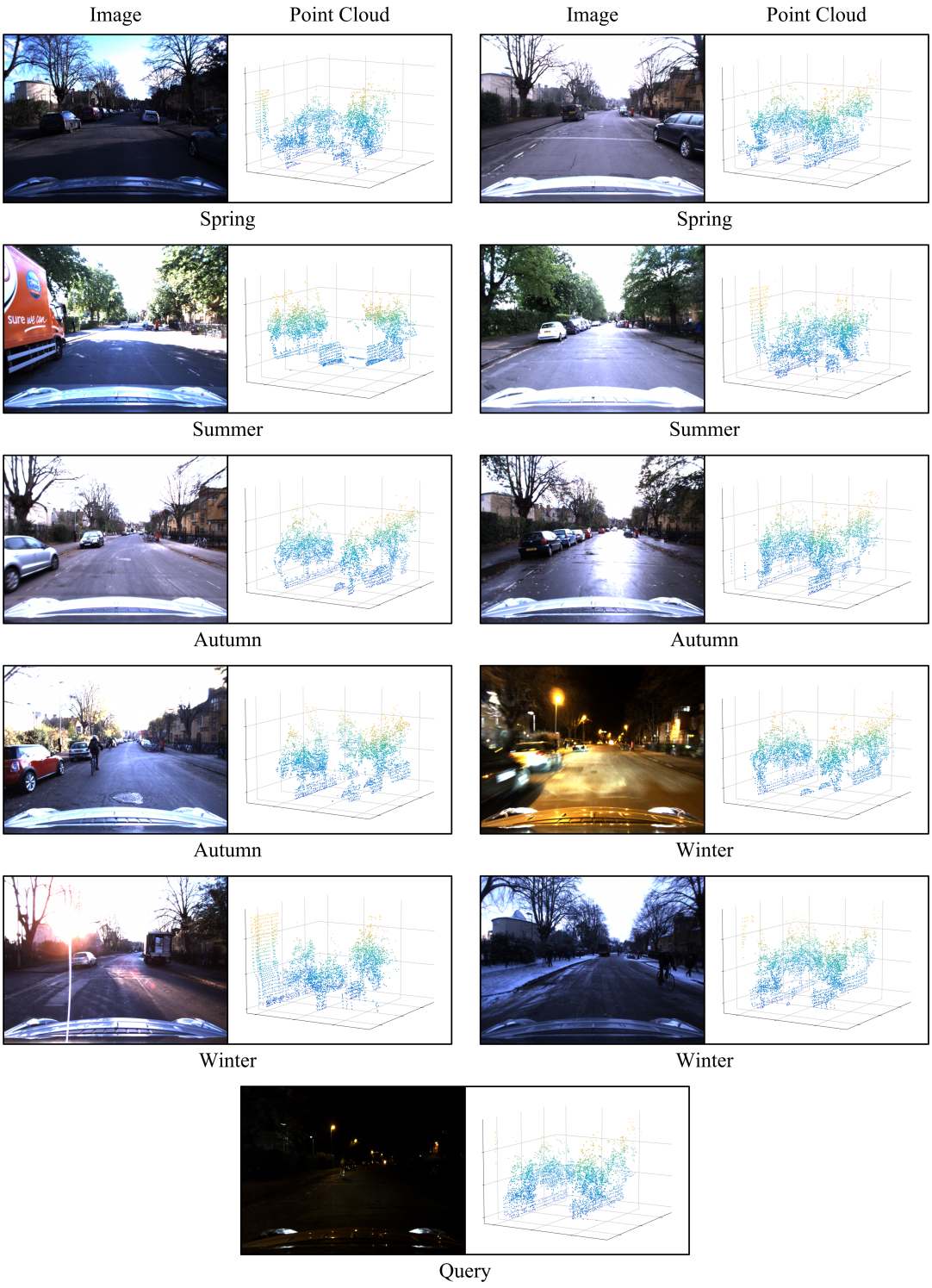


Figure 7: The top-1 recall results of LPS-Net for the same query scene. In order to demonstrate the changes in the environment, we present images (left side) corresponding to the point clouds (right side). It can be seen that due to factors such as tree growth and traffic conditions, point clouds collected at the same location at different times are not completely identical. However, LPS-Net is still able to accurately identify these scenes, highlighting the stability of our method.

References

Fan, Z.; Song, Z.; Liu, H.; Lu, Z.; He, J.; and Du, X. 2022. Svt-net: Super light-weight sparse voxel transformer for large scale place recognition. In *AAAI Conference on Artificial Intelligence*, volume 36, 551–560.

Hui, L.; Cheng, M.; Xie, J.; Yang, J.; and Cheng, M.-M. 2022. Efficient 3D point cloud feature learning for large-scale place recognition. *IEEE Transactions on Image Processing*, 31: 1258–1270.

Hui, L.; Yang, H.; Cheng, M.; Xie, J.; and Yang, J. 2021. Pyramid Point Cloud Transformer for Large-Scale Place Recognition. In *IEEE/CVF International Conference on Computer Vision*, 6078–6087.

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

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

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

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