# SVT-Net: Super Light-Weight Sparse Voxel Transformer for Large Scale Place Recognition

**Zhaoxin Fan[1], Zhenbo Song[3], Hongyan Liu[4*], Zhiwu Lu[2], Jun He[1*], and Xiaoyong Du[1]**

[1] Key Laboratory of Data Engineering and Knowledge Engineering of MOE,
School of Information, Renmin University of China, 100872, Beijing, China
[2] Gaoling School of Artificial Intelligence, Renmin University of China, 100872, Beijing, China
[3] School of Computer Science and Engineering, Nanjing University of Science and Technology, 210094, Nanjing, China
[4] Department of Management Science and Engineering, Tsinghua University, 100084, Beijing, China
{fanzhaoxin, luzhiwu, hejun}@ruc.edu.cn, hyliu@tsinghua.edu.cn, songzb@njust.edu.cn

## Abstract

Simultaneous Localization and Mapping (SLAM) and Autonomous Driving are becoming increasingly more important in recent years. Point cloud-based large scale place recognition is the spine of them. While many models have been proposed and have achieved acceptable performance by learning short-range local features, they always skip long-range contextual properties. Moreover, the model size also becomes a serious shackle for their wide applications. To overcome these challenges, we propose a super light-weight network model termed SVT-Net. On top of the highly efficient 3D Sparse Convolution (SP-Conv), an Atom-based Sparse Voxel Transformer (ASVT) and a Cluster-based Sparse Voxel Transformer (CSVT) are proposed respectively to learn both short-range local features and long-range contextual features. Consisting of ASVT and CSVT, SVT-Net can achieve state-of-the-art performance in terms of both recognition accuracy and running speed with a super-light model size (0.9M parameters). Meanwhile, for the purpose of further boosting efficiency, we introduce two simplified versions, which also achieve state-of-the-art performance and further reduce the model size to 0.8M and 0.4M respectively.

## Introduction

Large scale place recognition is the spine of a wide range of applications like Simultaneous Localization and Mapping (SLAM) (Mur-Artal and Tardós 2017), Autonomous Driving (Levinson et al. 2011), Robot Navigation (Ravankar et al. 2018), etc. Commonly, the place recognition result can be used for loop-closure (Chen et al. 2020) in a SLAM system or for user location in a indoor vision positioning system, when GPS signal is not available. Fig. 1 (Top) illustrates a common pipeline of large place recognition. For a large scale region, a database of scenes (usually represented by point clouds or images ) tagged with UTM coordinates acquired from GPS/INS readings are constructed in advance. When a user traverses the same region, he/her may collect a query scene from scratch. Then the most similar scene to the query scene should be retrieved from the database to determine where the location of the query scene is.

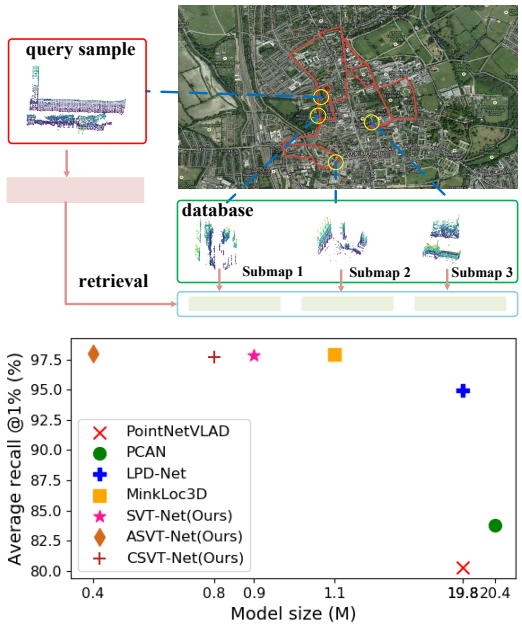


Figure 1: (Top) Pipeline of point cloud based place recognition. (Bottom) Model size and accuracy.

A straight-forward idea for this task is to use images to learn global descriptors for accurate and efficient scene retrieval (Li, Snavely, and Huttenlocher 2010; Han et al. 2017; Yu et al. 2019). However, images are sensitive to illumination, weather change, diurnal variation, etc, making models based on them unstable and unreliable. Besides, images are short of perceiving 3D scenes due to lack of depth information. Recently, a line of point cloud based deep learning models (Uy and Lee 2018; Zhang and Xiao 2019; Sun et al. 2020; Liu et al. 2019; Fan et al. 2020; Xia et al. 2021; Komorowski 2021) for large scale place recognition have been proposed. Since point clouds are invariant to illumination and weather changes, point cloud based methods are more robust than image based methods. Besides, since point clouds contain richer 3D information, global descriptors learned from them are stronger in describing 3D scenes

*Corresponding authors

than image descriptors and therefore they always achieve better performance.

Though better, existing point cloud based methods still face three main challenges. 1) Most of existing methods learn descriptors from point-wise point cloud encoders, which are sensitive to local noise. These local noise may stand for scene details and being important for some fine grained level tasks such as segmentation. However, they are useless for place recognition but even become a burden for the network to understand the scene. Therefore, they should be regarded as noise and outliers. 2) We observe that most of previous methods only consider how to better extract short-range local features, while the equally important long-range contextual properties have long been skipped. And we argue that lacking awareness of long-range contextual properties, power of the learned descriptors would be greatly limited. 3) Most of existing models are suffered from huge model size, which stops their application in resource constrained portable devices. Considering the above issues, we claim that designing a local noise-insensitive light-weight point cloud descriptor extraction model that can capture long-range contextual features is necessary.

In this paper, we propose a novel super light-weight network named SVT-Net for point cloud based large scale place recognition. SVT-Net's nework architecture is built upon the delicate light weight 3D Sparse Convolution (SP-Conv) (Choy, Gwak, and Savarese 2019). The reason why we choose SP-Conv lies in two aspects. First, the sparse voxel representation require to voxelize point cloud, which reduces local noise but retains most of overall scene geometries. Therefore, it can liberate the model from understanding useless scene details. Second, the SP-Conv is efficient and fast. It only computes outputs for predefined coordinates and saves them into a compact sparse tensor. In other words, it meets our requirements for building a light-weight model.

However, simply stacking SP-Conv layers may cause neglect of long-range contextual properties. A direct way to solve this problem is introducing Vision Transformers (Dosovitskiy et al. 2020) for learning long-range contextual features. There indeed exists point cloud Transformers (Guo et al. 2020) in literature, however, they are not suitable for the point cloud based place recognition task. It is because all existing point cloud Transformers are point-wise modules and therefore not efficient enough. Besides, as mentioned before, point-wise modules may suffer from local noise. Therefore, we propose two kinds of Sparse Voxel Transformers (SVTs) tailored for large scale place recognition on top of SP-Conv layers named Atom-based Sparse Voxel Transformer (ASVT) and Cluster-based Sparse Voxel Transformer (CSVT) respectively. ASVT and CSVT implicitly extract long-range contextual features from the sparse voxel representation through two perspectives: attending on different key atoms and clustering different key regions in the feature space, thereby helping to obtain more discriminative descriptors through interacting different atoms (to learn inter-atoms long-range features) and different clusters (to learn inter-clusters long-range features) respectively. Since SP-Conv only conducts convolution operation on non-empty voxels, it is computational efficient and flexible, so do the two SVTs built upon it. Thanks to the strong capabilities of the two SVTs, our proposed model can learn sufficiently powerful descriptors from an extremely shallow network architecture. And thanks to the shallow network architecture, model size of SVT-Net is very small as shown in Fig. 1 (Bottom).

We conduct extensive experiments on Oxford RobotCar dataset (Maddern et al. 2017) and three in-house datasets (Uy and Lee 2018) to verify the effectiveness and efficiency of SVT-Net. Results show that though light-weight, SVT-Net can achieve state-of-the-art performance in terms of both accuracy and speed. What's more, to further increase speed and reduce model size, we introduce two simplified version of SVT-Net: ASVT-Net and CSVT-Net, which also achieve state-of-the-art performances with further reduced model sizes of only 0.8M parameters and 0.4M parameters respectively.

Our main contributions are three folds. 1) We propose a novel light-weight point cloud based place recognition model named SVT-Net as well as two simplified versions: ASVT-Net and CSVT-Net, which all achieve state-of-the-art performance in terms of both accuracy and speed with a extremely small model size. 2) We propose Atom-based Sparse Voxel Transformer (ASVT) and Cluster-based Sparse Voxel Transformer (CSVT) for learning long-range contextual features hidden in point clouds. To the best of our knowledge, we are the first to propose Transformers for sparse voxel representations. 3) We have conducted extensive quantitative and qualitative experiments to verify the effectiveness and efficiency of our proposed models and analysed what the two proposed Transformers actually learn.

## Related Work

### Large Scale Place Recognition

Large scale place recognition plays an important role in SLAM and autonomous driving and has been interested in by many researchers for a long time. In early years, handcraft features (Gálvez-López and Tardos 2012; Fernández-Moral et al. 2013; Johns and Yang 2011) or learned features (Arandjelovic et al. 2016; Yu et al. 2019; Hausler et al. 2021) extracted from images are used for place recognition. These methods, though straight-forward, are suffered from vulnerability of features caused by images' sensitivity towards illumination, weather change, diurnal variation, etc.

Compared to image, point cloud is more insensitive to environmental changes, therefore it is a better alternative for place recognition. PointNetVLAD (Uy and Lee 2018) adopts PointNet (Qi et al. 2017) and NetVLAD (Arandjelovic et al. 2016) to learn global point cloud descriptors for this task. Then, a series of following works (Zhang and Xiao 2019; Sun et al. 2020; Fan et al. 2020; Liu et al. 2019; Xia et al. 2021; Komorowski 2021) are proposed. They use graph networks, attentions and voxel representation to learn powerful global descriptors for this task respectively. However, most of them are suffered from three aspects: first, they fail to learn long-range contextual features of scenes from point cloud; second, model size and efficiency are not considered in their methods; third, they are sensitive to local

noise. In our work, we design two light-weight but strong Sparse Voxel Tranformers to tackle the above problems.

## Vision Transformers

Transformer (Vaswani et al. 2017) is one of the most successful design for natural language processing (NLP)(Devlin et al. 2018; Hu, Shen, and Sun 2018; Yang et al. 2019; Kim, Son, and Kim 2021; Chen, Fan, and Panda 2021; Chen et al. 2018), the core of which is a self-attention mechanism to capture long-range contextual features. Recently, inspired by the great success of Transformer in NLP, researchers begin to design Transformers tailored for computer vision tasks.

Therefore, Vision Transformer (ViT) (Dosovitskiy et al. 2020) is proposed recently. It adopts the idea of self-attention and divides images to 16x16 visual words. In this way, images can be processed like nature language. Then, a variety of following works (Wu et al. 2020; Wang et al. 2021; Liu et al. 2021; Jiang, Chang, and Wang 2021) are proposed based on it. However, all the above introduced vision Transformers are designed for learning from images. To boost the performance of point cloud based tasks, point-wise vision Transformers like (Zhao et al. 2020; Guo et al. 2020) are proposed. Though tailored for point clouds, they are not suitable for the place recognition task. Because they are not light-weight enough and are suffered from small local noise in raw point clouds. In contrast, we propose two kinds of super-light Sparse Voxel Transformers to learn global features from scenes, which are less suffered from local noise and are much more efficient. To our knowledge, this is the first work designs Sparse Voxel Transformers for point clouds.

# Methodology

## Problem Definition

Let $M_r = \{m_i | i = 1, 2, ..., M\}$ be a database of pre-defined 3D submaps (represented as point clouds), and Q be a query point cloud scan. The place recognition problem is defined as retrieving a submap $m_s$ from $M_r$ with the goal of $m_s$ is the closest one to Q. To achieve accurate retrieving, a deep learning model $F(*)$ that can embed all point clouds into discriminative global descriptors, e.g. $Q \rightarrow f_q \in R^d$, is required so that a following KNNs algorithm can be used for finding $m_s$.

To meet the goal, we choose to use the sparse voxel representation of point cloud as input and choose 3D Sparse Convolution (SP-Conv) (Choy, Gwak, and Savarese 2019) as the basic unit to build the deep learning model. To employ SP-Conv, we first voxelize all point clouds into sparse voxel representations, e.g. $Q \rightarrow Q^v \in R^{L \times W \times H \times 1}$, where for each voxel, 1 means that it is occupied by any points in $Q$, called non-empty voxel, and otherwise 0, called empty voxel. SP-Conv operation is only conducted on non-empty voxels. Hence, it is very efficient and flexible. Next, we will introduce the two proposed Transformers: the Atom-based Sparse Voxel Transformer (ASVT) and the Cluster-based Sparse Voxel Transformer (CSVT) respectively. And then, the overall network architecture of SVT-Net as well as network architectures of the two simplified versions (ASVT-Net and CSVT-Net) will be introduced in detail. The loss function will be presented finally.

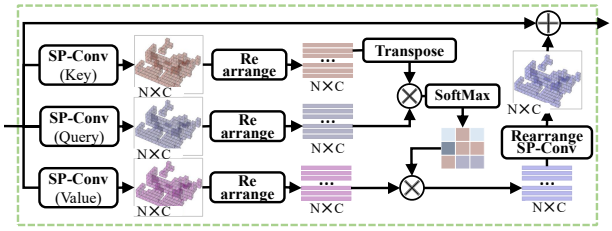


Figure 2: Network architecture of ASVT.

## Atom-Based Sparse Voxel Transformer

As mentioned before, simply stacking SP-Conv layers may cause the loss of learning long-range contextual features. To make up for this loss, we design the first Transformer, ASVT, which adopts the idea of self-attention to aggregate information from both nearby and far-away voxels to better capture sparse voxel features. In ASVT, we define each individual voxel as an atom. During processing, each atom should be interacted with all other atoms according to the learned per-atom contributions. By doing so, different key atoms could be attended by other atoms so that both local relationship of nearby atoms and long-range contextual relationship of far way atoms will be learned, i.e, inter-atoms long-range contextual features are learned. Note that learning such kind of inter-atoms long-range contextual relationship is very important for the model. For example, in a scene, assume there are two atoms that belong to different instances of the same category. If only SP-Conv is used, the "same-category" information may be ignored due to the small receptive field. While if AVST is added to learn such kind of information, the model can better encode what the scene describes. Hence the final global descriptor would be more powerful. The architecture of ASVT is illustrated in Fig. 2.

Let $X_{in} \in R^{L \times W \times H \times C}$ be the input sparse voxel features learned by SP-Convs (SP-voxel features for simplicity). We first learn the sparse voxel values (SP-values for simplicity) $X_v \in R^{L \times W \times H \times C}$, SP-queries $X_q \in R^{L \times W \times H \times C_r}$, and SP-keys $X_k \in R^{L \times W \times H \times C_r}$ through three different SP-Convs respectively:

$$\begin{aligned} X_v &= SPConv(X_{in}) \\ X_q &= SPConv(X_{in}) \\ X_k &= SPConv(X_{in}) \end{aligned} \tag{1}$$

where we often set $C_r < C$ to reduce computational cost in later steps. That is to say, the dimension of SP-queries and SP-keys are reduced from $C$ to $C_r$ for efficiency. After that, SP-voxel features of SP-values (SP-queries/keys) are rearranged to a tensor of $N \times C$ ($N \times C_r$), where $N$ is the number of non-empty voxels. The rearrange step is easy. Since coordinates and features of non-empty voxels have been already stored as sparse tensors in SP-Conv's output, we only need to take out the feature tensor from its data structure for

rearrange. Note that $N \ll L \times W \times H$ and $N \ll N_p$, where $N_p$ is point number of the raw point cloud, therefore, the SP-Conv and the following matrix multiplications based on the feature tensor are all very computational efficient.

Then, we use $X_q$ and $X_k$ to calculate the SP-attention map $S$:

$$S = softmax(X_q \cdot X_k^T) \tag{2}$$

where $S \in R^{N \times N}$ encodes the contribution relationship of each atom with all the other atoms. In the following attending operation, these relationships will contribute to aggregating both short-range local information and long-range contextual information by interacting atoms. The attending operation can be summarized as:

$$X_s = SPConv(S \cdot X_v) \tag{3}$$

where $X_s \in R^{N \times C}$ is called atom-attended SP-voxel features. In $X_s$, features of each atom $x_i$ have adaptively accepted contributions from all the other atoms according to the implicit mode hidden in $S$. Thus meaningful contextual and semantic information can be represented in $X_s$ to describe the scene.

Finally, we rearrange $X_s$ back to sparse voxel representations with a dimension of $L \times W \times H \times C$ and regard it as a residual term. The final ASVT feature is defined as the sum of $X_{in}$ and $X_s$:

$$X_{asvt} = X_{in} + X_s \tag{4}$$

## Cluster-Based Sparse Voxel Transformer

Another observation we find is: in the sparse voxel representation, some atoms may share the same characteristics. For example, atoms of a wall always form a plane like structure, while atoms of a flower bed easily form a cylinder like structure. This means that atoms can actually cluster into different clusters according to their geometric or semantic characteristics, and the long-range contextual properties can also be extracted from the perspective of interacting between these clusters, i.e, learning inter-clusters long-range contextual features. Motivated by this intuition, we propose the second Transformer: CSVT. As shown in Fig. 3, CSVT consists of three component, a Tokenizer module, a Transformer module and a Projector module. They cooperatively learn how to implicitly group atoms into characteristics-similar clusters and interact clusters for enhancing learned features. Next, we will introduce them in detail.

**The Tokenizer module** is used to transform the input SP-voxel features into tokens, where each token represents a cluster in the latent space. We again define $X_{in} \in R^{L \times W \times H \times C}$ as the initial SP-voxel features. To achieve the goals of the tokenizer, we first use a SP-Conv operation followed by a rearrange operation to generate a grouping map $X_g \in R^{N \times L_t}$ :

$$X_g = softmax(RE(SPConv(X_{in}))) \tag{5}$$

where $RE$ is the rearrange operation. $L_t$ is the number of tokens we choose to generate and $N$ is the number of non-empty voxels. $X_g$ stores the probabilities of each voxel belonging to each token. Therefore, we can use $X_g$ to capture representations of tokens as grouping different tokens into different clusters in an implicit way:

$$T = X_g^T \cdot SPConv(X_{in}) \tag{6}$$

where $T \in R^{L_t \times C}$ denotes representations of $L_t$ tokens with each of them being described by $C$ features.

**The Transformer module** is then used to learn inter-clusters long-range contextual features through interacting these tokens. First, we generate values, keys, and queries using shared convolutional kernel Conv1d:

$$T_v = Conv1d(T), T_q = Conv1d(T), T_k = Conv1d(T) \tag{7}$$

Then, tokens are interacted with each other through the following attention operation:

$$T_s = T + Conv1d(softmax(T_q \cdot T_k^T) \cdot T_v) \tag{8}$$

where $T_s \in R^{L_t \times C}$ is the attended tokens. Through the Transformer module, relationship between different clusters are learned to characterize the distribution characteristics of the scene with high quality. For example, the final descriptor may memorize that there is a rectangular building in the scene stands 5 meters away from a cylindrical building, or remember that there is a spherical building stands behind a slender tree.

**The Projector module** is then used to project token features back to the sparse voxel representation. Specifically, we use $T_s$ to calculate a re-projection map $M_p \in R^{N \times L_t}$:

$$T_p = Conv1d(T_s) \tag{9}$$

$$M_p = \text{softmax}(RE(SPConv(X_{in})) \cdot T_p^T) \tag{10}$$

where $T_p \in R^{L_t \times C}$. Then, the re-projection operation is defined as:

$$X_s = SPConv(M_p \cdot T_p) \tag{11}$$

Again, we rearrange $X_s$ back to sparse voxel representations with a dimension of $L \times W \times H \times C$ and regard it as a residual term. The final CSVT feature is defined as:

$$X_{csvt} = X_{in} + X_s \tag{12}$$

Note that, though both aim to learn long-range contextual features, roles and working mechanisms of ASVT and CSVT are different. The ASVT focus on learning relationship between similar and dissimilar individual atoms and learns inter-atoms long-range contextual features in a fine-grained level, while CSVT focus on learning relationship between different characteristics-similar clusters so that it learns inter-clusters long-range contextual features in a relative coarser level. They are complementary to each other.

## Network Architecture

The overall architecture of SVT-Net is built upon the above introduced ASVT and CSVT as well as the light-weight SP-Conv. Specifically, as shown in Fig. 4. The initial sparse voxel representation is fed into the first SP-Conv layer with an output dimension of 32 to learn initial SP-features. Then two SP-Res-Blocks (each consists of two SP-Convs with a

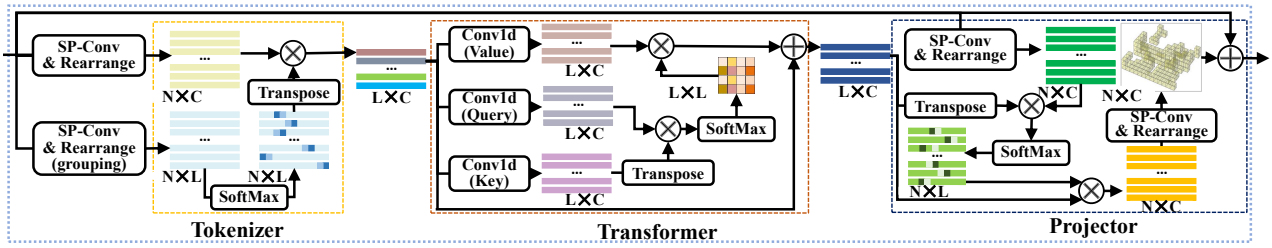


Figure 3: Network architecture of CSVT.

skip connection) are used to enhance learned features and increase the feature dimension to 64. Next, another SP-conv layer is used to increase the feature dimension to the final descriptor's dimension $d$. After that, the SP-features are fed into two branches for learning ASVT features and CSVT features using the two proposed Sparse Voxel Transformers(SVTs) respectively. Then, the learned ASVT features and CSVT features are fused by directly adding them together. Finally, the final global descriptor is calculated by using a GeM Pooling operation (Radenović, Tolias, and Chum 2018):

$$f = [f_1, \cdots, f_k, \cdots, f_d], f_k = \frac{1}{|X_{final,k}|} \sum_{x \in X_{final,k}} (x^{p_k})^{\frac{1}{p_k}} \quad (13)$$

where $f \in d$ is the final descriptor, $X_{final}$ is $X_{csvt} + X_{asvt}$, and $p_k$ is a learnable control parameter.

Other details of the network architecture can be found in **Supp**. Thanks to the strong power of ASVT and CSVT, our proposed model SVT-Net can achieve superior performance compared to previous methods, even though our network architecture is simpler and smaller (from another words, it is shallower). Note that ASVT and CSVT can also be individually utilized in different networks. Therefore, we propose two simplified versions of SVT-Net: ASVT-Net and CSVT-Net, by eliminating the ASVT module and CSVT module, respectively, to verify the effectiveness of the two modules. According to experimental results, both ASVT-Net and CSVT-Net also achieve state-of-the-art performances but further reduce the model size for a large margin.

## Loss Function

In view of its superior performance in (Komorowski 2021), we adopt the following triplet loss to train our model:

$$L(f_i, f_i^p, f_i^n) = max\{d(f_i, f_i^p) - d(f_i, f_i^n) + m, 0\} \quad (14)$$

where $f_i$ is the descriptor of the query scan, $f_i^p$ and $f_i^n$ are descriptors of positive sample and negative sample respectively, and $m$ is a margin. $d(x, y)$ means the Euclidean distance between $x$ and $y$. To build informative triplets, we use batch-hard negative mining following (Komorowski 2021).

After the network is trained, all point clouds are embedded into descriptors using the model. And we use the KNNs algorithm to find $m_s$ in the database, which is the closest one to the query scan $Q$.

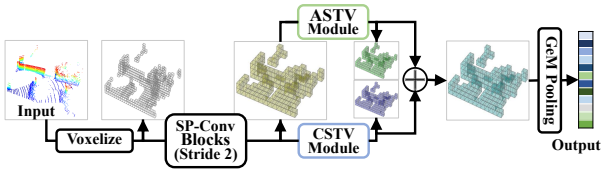


Figure 4: Pipeline of SVT-Net. The circle-add symbol means element-wise sum.

# Experiments

## Datasets and Metrics

To fairly compare with other methods, we use the benchmark datasets proposed by (Uy and Lee 2018) to evaluate our method, which are now recognized as the most influential datasets for point cloud based place recognition. The benchmark contains four datasets: one outdoor dataset named Oxford generated from Oxford RobotCar (Maddern et al. 2017) and three in-house datasets: university sector (U.S.), residential area (R.A.) and business district (B.D.). The benchmark contains 21711, 400, 320, 200 submaps for training and 3030, 80, 75, 200 submaps for testing for Oxford., U.S., R.A. and B.D. respectively. Each point cloud contains 4096 points, which is the common setting of point cloud based place recognition. We use average recall at top 1% and average recall at top 1 as main metrics as previous methods for a fair comparison.

## Implementation Details

In all experiments, we voxelize 3D point coordinates with 0.01 quantization step. The voxelization and the following SP-Conv operation are performed by the MinkowskiEngine auto differentiation library (Choy, Gwak, and Savarese 2019). The dimension of the final descriptor is set to 256. The number of tokens $L_t$ is set to 8. Following previous work, we train two versions of models: baseline model and refined model. The baseline model is trained only using the training set of Oxford dataset, and the refined model is trained by adding the training set of U.S. and R.A. (Note that training set of B.D. is not added). Random jitter, random translation, random points removal and random erasing augmentation are adopted for data augmentation during training. All experiments are performed on a Tesla V100 GPU with a memory of 32G. More details can be found in the **Supp**.

| Method | Average recall at top-1% (%) | | | | Average recall at top-1 (%) | | | |
|---|---|---|---|---|---|---|---|---|
| | Oxford | U.S. | R.A. | B.D. | Oxford | U.S. | R.A. | B.D. |
| **PointNetVLAD** | 80.3 | 72.6 | 60.3 | 65.3 | - | - | - | - |
| **PCAN** | 83.8 | 79.1 | 71.2 | 66.8 | - | - | - | - |
| **DAGC** | 87.5 | 83.5 | 75.7 | 71.2 | - | - | - | - |
| **SOE-Net** | 96.4 | 93.2 | 91.5 | 88.5 | - | - | - | - |
| **SR-Net** | 94.6 | 94.3 | 89.2 | 83.5 | 86.8 | 86.8 | 80.2 | 77.3 |
| **LPD-Net** | 94.9 | 96.0 | 90.5 | 89.1 | 86.3 | 87.0 | 83.1 | 82.3 |
| **Minkloc3D** | 97.9 | 95.0 | 91.2 | 88.5 | 93.0 | 86.7 | 80.4 | 81.5 |
| **SVT-Net(Ours)** | 97.8 | **96.5** | **92.7** | **90.7** | 93.7 | **90.1** | **84.3** | **85.5** |
| **ASVT-Net(Ours)** | **98.0** | 96.1 | 92.0 | 88.4 | **93.9** | 87.9 | 83.3 | 82.3 |
| **CSVT-Net(Ours)** | 97.7 | 95.5 | 92.3 | 89.5 | 93.1 | 88.3 | 82.7 | 83.3 |

Table 1: Comparison with the state-of-the-art methods under the baseline setting.

## Main Results

In this section, we experimentally verify the effectiveness and efficiency of our method. Specifically, we first compare our models with PointNetVLAD (Uy and Lee 2018), PCAN (Zhang and Xiao 2019), DAGC (Sun et al. 2020), SR-Net (Fan et al. 2020), LPD-Net (Liu et al. 2019), SOE-Net (Xia et al. 2021) and Minkloc3D (Komorowski 2021) in terms of recognition accuracy. Then, we compare the inference time and model size between our models with them. Finally, we qualitatively analyze what the two SVTs have learned.

**Accuracy**: In Table 1, we show the results of all methods on the baseline setting. It can be found that SVT-Net significantly outperforms all state-of-the-art methods, especially for the average recall at top 1 metric on U.S., R.A., and B.D., where SVT-Net wins for 3.4%, 3.9%, 4% compared to Minkloc3D respectively. Compared to SVT-Net, performances of ASVT-Net and CSVT-Net drop to some extent. However, their performances still largely outperform the previous best model Minkloc3D. We contribute the accuracy gain to the two novel SVTs we design. Note that Minkloc3D is also built upon SP-Conv and shares the same loss function as our model, while its performance is not as excellent as our models, which further confirms the superiority of our two proposed SVTs. Specifically, our SVT-Net build lightweight sparse voxel transformers based on SPConv, while Minkloc3D simply stacks SPConv layers, which is the main difference between Minkloc3D and our model, and therefore it is the two SVTs being the main force make our model perform better. What's more, SOE-Net also use self-attention in its network architecture to learn long range context dependencies, but our model outperforms SOE-Net. This demonstrates that sparse voxel transformers are more effective than point-wise transformers for large scale place recognition. We also note the self-attention module in SOE-Net is inefficient especially when the number of points is large due to computing attention weights for each of the $N_p$ raw points. In contrast, the novel ASVT and CSVT in our SVT-Net are built for processing sparse voxels, which are much more efficient because we only need to compute attention weights for each of $N$ ($N \ll N_p$) non-empty voxels. Recall curves of the baseline setting can be found in **Supp**. We also visualize some top-k matching results in Fig. 5 to provide readers with a comprehensive view to understand our place recognition results.

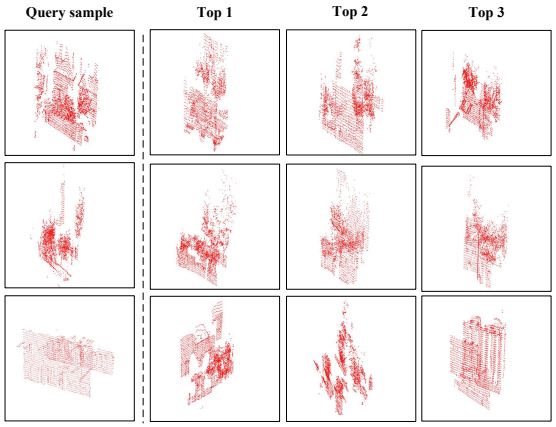


Figure 5: Visualization of top 3 matching results.

| Method | Time | Parameters |
|---|---|---|
| **PointNetVLAD** | - | 19.8M |
| **PCAN** | - | 20.4M |
| **LPD-Net** | - | 19.8M |
| **Minkloc3D** | 12.16ms | 1.1M |
| **SVT-Net(Ours)** | 12.97ms | 0.9M |
| **ASVT-Net(Ours)** | **11.04ms** | **0.4M** |
| **CSVT-Net(Ours)** | 11.75ms | 0.8M |

Table 2: Efficiency comparison.

For a comprehensive comparison, we also show the results of all models at the refined setting in **Supp**. We find that at the refined setting, our models still significantly outperform all models except Minkloc3D. In fact, our models still perform better than Minklo3D in most cases, although only by a small margin. The difference between our three models becomes narrow. We attribute this to that all models have reached the performance upper bound.

**Model size and speed**: To verify the efficiency of our method, we compare our models with previous works in terms of model size and inference time in Table 2 and Fig.

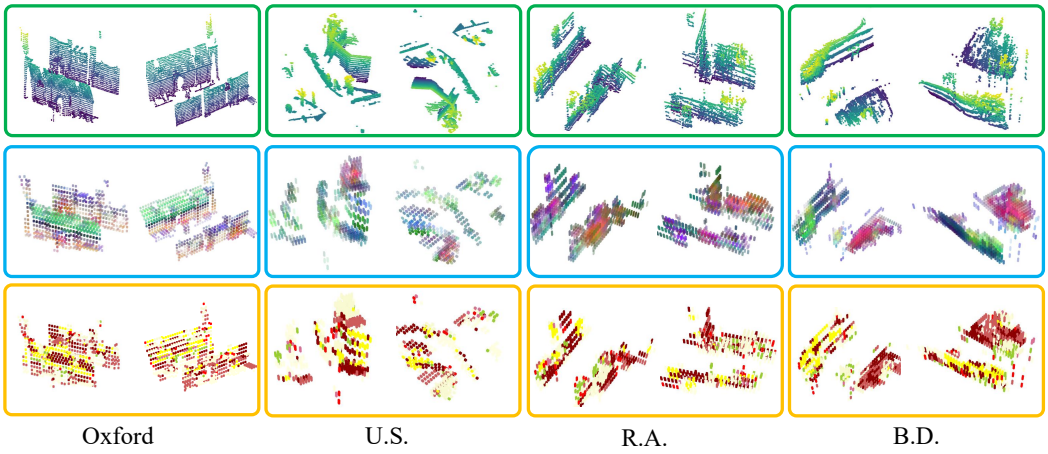


Figure 6: Visualization of what ASVT and CSVT have learned. First row: original point clouds. Second row: features learned by ASVT, "same category" atoms are attended similarly, e.g, in Oxford, two walls of the same height share the same color. Third row: features learned by CSVT, atoms belong to the same geometric shape are clustered together and interacted with each other, e.g, in B.D., all the atoms in the same flowerbed (colored in Crimson) form a cube and are clustered together.

| | Average recall at top-1% (%) | | | | Average recall at top-1 (%) | | | |
|---|---|---|---|---|---|---|---|---|
| **Method** | **Oxford** | **U.S.** | **R.A.** | **B.D.** | **Oxford** | **U.S.** | **R.A.** | **B.D.** |
| **A:** $L_t$**=4,** $d$**=256,** $add$ | 97.9 | 96.4 | 92.5 | 89.0 | 93.7 | 89.0 | 83.9 | 82.5 |
| **B:** $L_t$**=6,** $d$**=256,** $add$ | 98.0 | 96.2 | 92.3 | 90.1 | 93.8 | 88.3 | 83.7 | 84.4 |
| **C:** $L_t$**=10,** $d$**=256,** $add$ | 97.9 | 96.2 | 92.0 | 89.4 | 93.8 | 87.2 | 83.3 | 83.5 |
| **D:** $L_t$**=8,** $d$**=128,** $add$ | 97.8 | 95.2 | 92.0 | 89.0 | 93.3 | 88.9 | 81.9 | 82.5 |
| **E:** $L_t$**=8,** $d$**=384,** $add$ | **98.2** | 94.8 | 92.5 | 89 | **94.4** | 86.9 | **84.9** | 83.7 |
| **F:** $L_t$**=8,** $d$**=512,** $add$ | 98.0 | **97.3** | 92.1 | 88.2 | 93.9 | **90.1** | 84.0 | 82.7 |
| **G:** $L_t$**=8,**$d$ **=512,** $cat$ | 97.5 | 93.4 | 85.8 | 84.7 | 92.7 | 81.9 | 73.9 | 77.1 |
| **H:** $L_t$**=8,** $d$**=256,** $cat\&spconv$ | 96.5 | 89.8 | 84.5 | 82.4 | 89.5 | 78.2 | 71.2 | 74.0 |
| **SVT-Net:** $L_t$**=8,** $d$**=256,** $add$ | 97.8 | 96.5 | **92.7** | **90.7** | 93.7 | **90.1** | 84.3 | **85.5** |

Table 3: Results of ablation study for our SVT-Net.

1 respectively. For model size, it can be seen that SVT-Net and CSVT-Net save 18.2% and 27.3% parameters respectively compared to the existing smallest model Minkloc3D. As for ASVT-Net, it even only has 36.4% parameters of Minkloc3D, which is a significant reduction. And it is worth noting that all of our three models outperform Minkloc3D for a large margin in terms of accuracy at the baseline setting. The ability of significantly improving accuracy under the condition of drastically reduced parameters further fully demonstrates the superiority of our two SVTs. For speed, compared to the current fastest model Minkloc3D, SVT-Net only add ignorable additional inference time. And both ASVT-Net and CSVT-Net run faster than Minkloc3D. Approximately, voxelization and SP-Conv blocks cost about half of the running time, while ASVT and CSVT cost the another half. We find the speed increase is not as significant as the model size reduction, which is because that the inherent Transformer operation requires multiple matrix multiplications. Summing up the above results, we can conclude that our models are good enough in terms of both model size and running speed. Note, compared to Minkloc3D, our network architectures are much shallower, that's why our models are more light-weight than it. And since the main difference between our models with Minkloc3D is the two SVTs we design, we can contribute all performance gains into the learned long-range contextual features.

We believe that expect recognition accuracy, both storage efficiency and recognition accuracy are also significant factors to make solid and convincing comparisons. In this work, extensive results show that our model outperforms the SOTA in all the three aspects. Besides, we also find our three version show different specialties towards the three different aspects, and so we can accordingly make different utilization choice. Specifically, SVT-Net is larger than ASVT-Net and CSVT-Net, but performs better in most cases. Therefore, if there is enough resource, we recommend to use SVT-Net for the place recognition task. If you only have limited computational resource and can't fine-tune the model on new scenarios, we recommend to use CSVT-Net because its generalization ability towards new scenarios is better than ASVT-Net. Otherwise, ASVT-Net is a better choice because it is the fastest and the smallest one.

**What Transformers have learned**: One may be interested in what ASVT and CSVT have learned that could make our

models so elegant. To explore this question, we show some visualization results in Fig. 6. The first row shows the original point clouds randomly selected from Oxford, U.S., R.A. and B.D respectively.

In the second row, we visualize the features of each non-empty voxel after ASVT using T-SNE (Van der Maaten and Hinton 2008). Different colors represent different distribution of these features in the feature space. It can be seen that by interacting each atom with all the others, the model indeed learns the relationship between atoms. Specifically, it is obvious that nearby atoms share the same color, which means they are attended similarly since they may belong to the same object parts. And it can be seen that far away atoms in the 3D space sharing the same implicit mode have similar colors, which means inter-atoms long-range features like relationship between far way semantic similar atoms (e.g., the "same-category" information) has been discovered by the model. A typical example that can prove the above analysis is: in Oxford, two walls of the same height share the same color, which means atoms of them are attend similarly.

In the third row, we visualize which token that each non-empty voxel belongs to. Different color represents different tokens. It can be seen that voxels belong to the same token always represent the same objects and share some common geometric characteristics. For example, in B.D., all the atoms in the same flowerbed (colored in Crimson) form a cube and are clustered together. This observation means that voxels indeed have been clustered together in the feature space according to their geometric characteristics. And obviously, the interaction between clusters or tokens could enhance model's understanding towards the scene. The inter-clusters long-range context properties like the relative positions between clusters would be encoded through such kind of interaction. In a word, the visualization results have confirmed our intuition of designing ASVT and CSVT and they have all contributed to the performance improvement.

## Ablation Study

We have verified the effectiveness and efficiency of ASVT and CSVT in the **Main Results** section. Next, we experimentally study the effectiveness of other key designs. Specifically, we study the impact of the number of tokens $L_t$, dimension of descriptors $d$, Transformer feature fusion strategy and training stability. We design experiments from A to H for this study. Table 3 shows the results. "SVT-Net" in the last row of Table 3 refers to the model we finally choose.

**Impact of number of tokens**: The number of tokens ($L_t$) decides how many clusters we divide the scene into. We change the value of $L_t$ and compare the results in Table 3. Comparing the experiment A, B, C and SVT-Net, we find that setting $L_t$ as 8 is the best choice. When $L_t$ is too small, interaction between different geometric characteristic (hidden in different clusters) would be limited. When $L_t$ is too large, it is easy to cause over-fitting.

**Impact of descriptor's dimension**: To a certain extent, the dimension $d$ determines the descriptor's capability of describing a scene. From experiment D, E, F and SVT-Net in Table 3, we find that overall larger dimension leads to better performance. However, when it is larger than 256, the accuracy increase is minimal while the model size is significantly increased to 1.8M and 3.0M for $d = 384$ and $d = 512$ respectively. Therefore, for a better trade-off between accuracy and model size, we choose $d = 256$ in our implementation.

**Impact of fusion strategy**: In SVT-Net, we need to fuse features learned by ASVT and CSVT before aggregating voxel features into a global descriptor. In experiment G, we investigate the effectiveness of another fusion method, concatenation. In this way, the output dimension is $512$. However, the performance of concatenating the two features is not as good as simply adding them (the dimension is 256). Then, we suspect if it is the higher dimension that causes the performance drop. Therefore, in experiment $H$, we add an additional SP-Conv layer after concatenation to make the dimension be 256. Unfortunately, the model's performance becomes even worse than before. Therefore, finally, we believe that direct adding together is the best way to fuse the features of the two SVTs.

**Training stability**: We notice that for each training time, there are some small differences on the evaluation results. To avoid conclusion bias, we train each model for multiple times and show the boxplot of each model in **Supp**, which reflects the training stability of each model. Considering the trade off between accuracy, model size, and training stability, we claim that SVT-Net is the best performed model.

## Conclusions

In this paper, we proposed a super light-weight network for large scale place recognition named SVT-Net. In SVT-Net, two Sparse Voxel Transformers: Atom-based Sparse Voxel Transformer (ASVT) and Cluster-based Sparse Voxel Transformer (CSVT) are proposed to learn long-range contextual properties. Extensive experiments have demonstrated that SVT-Net as well as its two simplified versions ASVT-Net and CSVT-Net can all achieve state-of-the-art performance with an extremely light-weight network architecture.

## Acknowledgements

This work was supported in part by National Key Research and Development Program of China under Grant No. 2020YFB2104101 and National Natural Science Foundation of China (NSFC) under Grant Nos. 62172421, 71771131, U1711262 and 62072459.

## References

Arandjelovic, R.; Gronat, P.; Torii, A.; Pajdla, T.; and Sivic, J. 2016. NetVLAD: CNN architecture for weakly supervised place recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 5297–5307.

Chen, C.-F.; Fan, Q.; and Panda, R. 2021. CrossViT: Cross-Attention Multi-Scale Vision Transformer for Image Classification. *arXiv preprint arXiv:2103.14899*.

Chen, X.; Läbe, T.; Milioto, A.; Röhling, T.; Vysotska, O.; Haag, A.; Behley, J.; Stachniss, C.; and Fraunhofer, F. 2020.

OverlapNet: Loop closing for LiDAR-based SLAM. In *Proc. of Robotics: Science and Systems (RSS)*.

Chen, Y.; Kalantidis, Y.; Li, J.; Yan, S.; and Feng, J. 2018. A2-Nets: Double Attention Networks. *arXiv preprint arXiv:1810.11579*.

Choy, C.; Gwak, J.; and Savarese, S. 2019. 4d spatio-temporal convnets: Minkowski convolutional neural networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 3075–3084.

Devlin, J.; Chang, M.-W.; Lee, K.; and Toutanova, K. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Dosovitskiy, A.; Beyer, L.; Kolesnikov, A.; Weissenborn, D.; Zhai, X.; Unterthiner, T.; Dehghani, M.; Minderer, M.; Heigold, G.; Gelly, S.; et al. 2020. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*.

Fan, Z.; Liu, H.; He, J.; Sun, Q.; and Du, X. 2020. SRNet: A 3D Scene Recognition Network using Static Graph and Dense Semantic Fusion. *Computer Graphics Forum*, 39(7): 301–311.

Fernández-Moral, E.; Mayol-Cuevas, W.; Arevalo, V.; and Gonzalez-Jimenez, J. 2013. Fast place recognition with plane-based maps. In *2013 IEEE International Conference on Robotics and Automation*, 2719–2724. IEEE.

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

Guo, M.-H.; Cai, J.-X.; Liu, Z.-N.; Mu, T.-J.; Martin, R. R.; and Hu, S.-M. 2020. PCT: Point Cloud Transformer. *arXiv preprint arXiv:2012.09688*.

Han, F.; Yang, X.; Deng, Y.; Rentschler, M.; Yang, D.; and Zhang, H. 2017. SRAL: Shared representative appearance learning for long-term visual place recognition. *IEEE Robotics and Automation Letters*, 2(2): 1172–1179.

Hausler, S.; Garg, S.; Xu, M.; Milford, M.; and Fischer, T. 2021. Patch-NetVLAD: Multi-Scale Fusion of Locally-Global Descriptors for Place Recognition. *arXiv preprint arXiv:2103.01486*.

Hu, J.; Shen, L.; and Sun, G. 2018. Squeeze-and-excitation networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 7132–7141.

Jiang, Y.; Chang, S.; and Wang, Z. 2021. Transgan: Two transformers can make one strong gan. *arXiv preprint arXiv:2102.07074*.

Johns, E.; and Yang, G.-Z. 2011. From images to scenes: Compressing an image cluster into a single scene model for place recognition. In *2011 International Conference on Computer Vision*, 874–881. IEEE.

Kim, W.; Son, B.; and Kim, I. 2021. ViLT: Vision-and-Language Transformer Without Convolution or Region Supervision. *arXiv preprint arXiv:2102.03334*.

Komorowski, J. 2021. MinkLoc3D: Point Cloud Based Large-Scale Place Recognition. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 1790–1799.

Levinson, J.; Askeland, J.; Becker, J.; Dolson, J.; Held, D.; Kammel, S.; Kolter, J. Z.; Langer, D.; Pink, O.; Pratt, V.; et al. 2011. Towards fully autonomous driving: Systems and algorithms. In *2011 IEEE Intelligent Vehicles Symposium (IV)*, 163–168. IEEE.

Li, Y.; Snavely, N.; and Huttenlocher, D. P. 2010. Location recognition using prioritized feature matching. In *European conference on computer vision*, 791–804. Springer.

Liu, Z.; Lin, Y.; Cao, Y.; Hu, H.; Wei, Y.; Zhang, Z.; Lin, S.; and Guo, B. 2021. Swin transformer: Hierarchical vision transformer using shifted windows. *arXiv preprint arXiv:2103.14030*.

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 *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2831–2840.

Maddern, W.; Pascoe, G.; Linegar, C.; and Newman, P. 2017. 1 year, 1000 km: The Oxford RobotCar dataset. *The International Journal of Robotics Research*, 36(1): 3–15.

Mur-Artal, R.; and Tardós, J. D. 2017. Orb-slam2: An open-source slam system for monocular, stereo, and rgb-d cameras. *IEEE Transactions on Robotics*, 33(5): 1255–1262.

Qi, C. R.; Su, H.; Mo, K.; and Guibas, L. J. 2017. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 652–660.

Radenović, F.; Tolias, G.; and Chum, O. 2018. Fine-tuning CNN image retrieval with no human annotation. *IEEE transactions on pattern analysis and machine intelligence*, 41(7): 1655–1668.

Ravankar, A.; Ravankar, A. A.; Kobayashi, Y.; Hoshino, Y.; and Peng, C.-C. 2018. Path smoothing techniques in robot navigation: State-of-the-art, current and future challenges. *Sensors*, 18(9): 3170.

Sun, Q.; Liu, H.; He, J.; Fan, Z.; and Du, X. 2020. Dagc: Employing dual attention and graph convolution for point cloud based place recognition. In *Proceedings of the 2020 International Conference on Multimedia Retrieval*, 224–232.

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

Van der Maaten, L.; and Hinton, G. 2008. Visualizing data using t-SNE. *Journal of machine learning research*, 9(11).

Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, L.; and Polosukhin, I. 2017. Attention is all you need. *arXiv preprint arXiv:1706.03762*.

Wang, W.; Xie, E.; Li, X.; Fan, D.-P.; Song, K.; Liang, D.; Lu, T.; Luo, P.; and Shao, L. 2021. Pyramid vision transformer: A versatile backbone for dense prediction without convolutions. *arXiv preprint arXiv:2102.12122*.

Wu, B.; Xu, C.; Dai, X.; Wan, A.; Zhang, P.; Tomizuka, M.; Keutzer, K.; and Vajda, P. 2020. Visual transformers: Token-based image representation and processing for computer vision. *arXiv preprint arXiv:2006.03677*.

Xia, Y.; Xu, Y.; Li, S.; Wang, R.; Du, J.; Cremers, D.; and Stilla, U. 2021. Soe-net: A self-attention and orientation encoding network for point cloud based place recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 11348–11357.

Yang, Z.; Dai, Z.; Yang, Y.; Carbonell, J.; Salakhutdinov, R.; and Le, Q. V. 2019. Xlnet: Generalized autoregressive pretraining for language understanding. *arXiv preprint arXiv:1906.08237*.

Yu, J.; Zhu, C.; Zhang, J.; Huang, Q.; and Tao, D. 2019. Spatial pyramid-enhanced NetVLAD with weighted triplet loss for place recognition. *IEEE transactions on neural networks and learning systems*, 31(2): 661–674.

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

Zhao, H.; Jiang, L.; Jia, J.; Torr, P.; and Koltun, V. 2020. Point transformer. *arXiv preprint arXiv:2012.09164*.