SARNet: Semantic Augmented Registration of Large-Scale Urban Point Clouds

Haobo Qin[2,1], Yinchang Zhou[1], Chao Liu[3⋆], Xiaopeng Zhang[1,2], Zhanglin Cheng[4], and Jianwei Guo[1,2⋆[0000−0002−3376−1725]]

[1] MAIS, Institute of Automation, Chinese Academy of Sciences
[2] School of Artificial Intelligence, University of Chinese Academy of Sciences
[3] Singapore University of Technology and Design
[4] SIAT, Chinese Academy of Sciences

Abstract. Registering urban point clouds is a pretty challenging task due to the large-scale, noise and data incompleteness of LiDAR scanning data. In this paper, we propose *SARNet*, a novel semantic augmented registration network aimed at achieving efficient registration of urban point clouds at city scale. Different from previous methods that construct correspondences only in the point-level space, our approach fully exploits semantic features as assistance to improve registration accuracy. Specifically, we extract per-point semantic labels with advanced semantic segmentation networks and build a prior semantic part-to-part correspondence. Then we incorporate the semantic information into a learning-based registration pipeline, consisting of three core modules: a *semantic-based farthest point sampling module* to efficiently filter out outliers and dynamic objects; a *semantic-augmented feature extraction module* for learning more discriminative point descriptors; a *semantic-refined transformation estimation module* that utilizes prior semantic matching as a mask to refine point correspondences by reducing false matching for better convergence. We evaluate the proposed SARNet extensively by using real-world data from large regions of urban scenes and comparing it with alternative methods. The code is available at *https://github.com/WinterCodeForEverything/SARNet*.

Keywords: 3D Registration · Semantic Segmentation · Large-scale Point Cloud.

1 Introduction

Point cloud registration aims to estimate an optimal rigid transformation to align two partially overlapping 3D point clouds. It is a fundamental task in computer graphics and 3D vision with numerous downstream applications, including 3D scene reconstruction, autonomous driving, robotics, and augmented reality.

A typical pipeline of traditional registration is first building correspondences between point clouds using local similar features, then estimating the transformation based on matched point pairs [24]. The registration result can be further refined by using

⋆ Corresponding Authors: Chao Liu (liuchao20200202@gmail.com), Jianwei Guo (jianwei.guo@nlpr.ia.ac.cn)

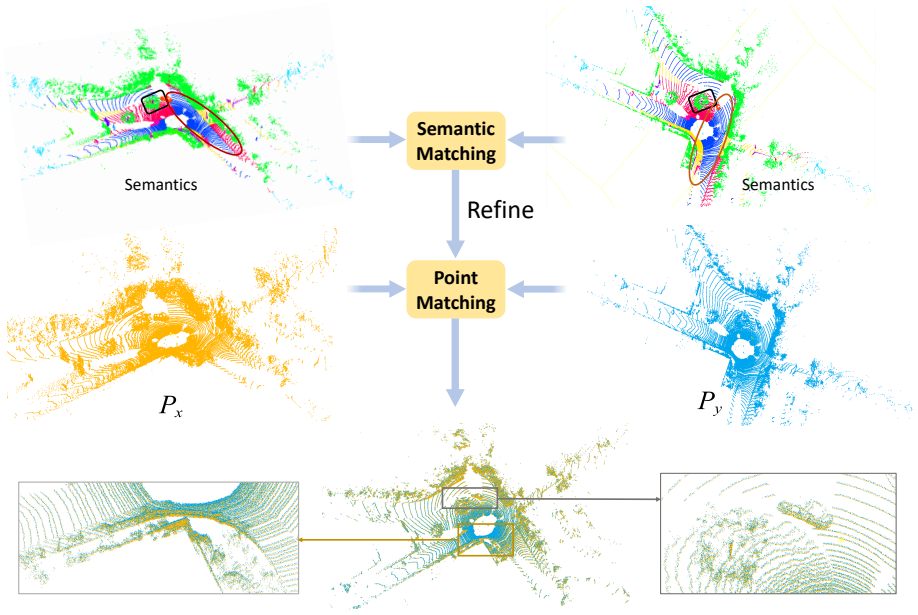

Fig. 1. A human-intuitive experience is that the matching points between the source and target point clouds should be in the same semantic category. Motivated by this human vision, we proposed a novel registration neural network with the assistance of semantic segmentation.

an *Iterative Closest Point* (ICP) algorithm [6]. In order to increase the robustness and efficiency of correspondence searching between point clouds, researchers have proposed a variety of approaches based on hand-crafted 3D feature descriptors and pose optimization algorithms [12, 33, 61, 47, 48]. However, due to their locality, those feature-based methods can not generalize well to highly noisy and non-uniform point clouds.

In the last decade, deep neural network has achieved remarkable success in many 3D vision tasks, such as point cloud classification and segmentation [18]. Thanks to the advances in 3D representation learning and the availability of a large number of annotated 3D datasets, deep point cloud registration has also drawn a lot of attention recently. Such methods either estimate an accurate correspondence search by robustly feature learning [55, 49, 10, 15, 50], or directly learn the final transformation matrix by proposing end-to-end neural networks [3, 9, 23, 35, 53, 30]. Although they achieve encouraging results on object-level and indoor scenes, it is still a challenging problem for those learning-based methods when registering large-scale point clouds. In real-world outdoor scenes, the LiDAR scanning range is usually large (*e.g.,* up to several hundred meters), and the point cloud containing millions of points will incur a high computational cost. The higher sparsity and data incompleteness, the presence of noise and outliers, and the limited overlap also pose great challenges for urban point cloud registration. Moreover, existing methods still struggle to handle a large proportion of erroneous matches, due to many similar and repeated features present in large-scale datasets.

Essentially, most existing methods perform registration just in the point-level space by utilizing the relationship between neighboring points, but do not consider higher-level information. Recently, there have been some methods that exploit semantic information to improve the registration robustness for LiDAR scans [34, 52, 54, 42, 4, 51]. Although these methods reduce the algorithm complexity and speed up computing point correspondences, they just simply integrate semantic information into existing traditional frameworks (*e.g.,* ICP, *Normal Distributions Transform* (NDT)). Therefore, they do not make full use of the semantic prior, *i.e.,* the semantic information is only used for classification and filtering inaccurate matches, but is not fully embedded with geometric features in the core registration stage.

In this paper, we present a novel and efficient neural network that estimates 3D rigid transformations by taking full advantage of semantic knowledge. Our framework is built on the observation that large-scale semantic segmentation of 3D urban scenes has reached high precision with deep learning technologies. As a result, utilizing the semantic segmentation information to establish part-to-part correspondences is reasonable and feasible. In our approach, we exploit SphereFormer [29] to segment point clouds due to its lightweight and efficiency. To effectively incorporate semantic labels into the learning-based registration pipeline, we design a semantic-based farthest point sampling (FPS) approach to obtain a subset of key points and also remove outliers (*e.g.,* dynamic objects). Then a more robust and discriminative point feature is learned in which we not only encode local and global geometric information, but also perform feature exchange between the source and target point clouds by using a cross-attention mechanism. Furthermore, we concatenate the semantic feature with the geometric feature to obtain a semantic-augmented hybrid feature descriptor. Finally, after computing a point-wise correspondence matrix, we propose to utilize a semantic similarity matrix as a mask to reject unreliable point matches.

In summary, the main contributions of this work include the following:

- A novel and reliable neural network, called SARNet, which fully exploits semantic consistency and geometric features to achieve almost state-of-the-art registration performance for urban point clouds.
- A semantic-augmented feature extraction module for learning rich and representative point descriptors. This module is built on a novel semantic-based FPS scheme that efficiently filters out dynamic objects and other outliers to improve registration accuracy.
- A semantic-refined transformation estimation module that utilizes prior semantic part-to-part matching as a mask to refine point correspondences, thus reducing the searching space of pose hypotheses for better convergence.

2 Related work

2.1 Traditional feature-based registration

Traditional point cloud registration methods mainly transfer the registration into an optimization problem, whose critical idea is to develop a sophisticated optimization strategy to find the optimal transformation. They can be further divided into global

registration and local registration methods. Global registration methods usually design handcraft local features, and search the feature correspondence, finally estimate the optimal transformation. The commonly used local features include Spin image [26], FPFH [40], SHOT [41], PPF [11, 16, 46], RoPS[17] etc. In addition to point-level features, Zhang [58] applies hybrid structural features which are constructed by extracting geometric primitives from point clouds to solve the low overlap even no overlap challenges. After defining the point features, efficient feature correspondence searching strategies are proposed to improve the registration efficiency and accuracy [13, 1, 61]. For example, RANSAC [13] is a common feature-matching algorithm that randomly samples small subsets of correspondences and finds optimal correspondences for transformation estimation. Using FPFH features, Zhou *et al.* [61] present an optimization algorithm for fast global registration with partially overlapping 3D surfaces. Zhang *et al.* [59] recently proposes a geometric-only 3D registration method by using the maximal clique constraint to produce accurate pose hypotheses from initial correspondences. However, without considering semantic information, this method often fails to find accurate hypotheses for sparse and simple geometric features.

Local registration methods often start from an initial transformation and solve the refined problem after global registration. ICP [6] is the most representative algorithm, which iteratively finds the closest points and updates the transformation by solving the least square problem. There are several variants of ICP to improve its effectiveness or robustness, such as [39] by selecting suitable points, [14] by weighting point correspondences, and IMLP [7] by incorporating the measurement of noise. Classical registration methods don't require large quantities of training data but may not perform well on point clouds with noises and outliers.

2.2 Learning-based registration

Learning-based registration methods can be roughly classified into feature-learning methods and end-to-end learning registration [24]. Feature-learning methods leverage the deep neural network to learn a robust feature correspondence search. PPFNet[10] uses PPF [11] which is rotation invariant to process point cloud patches. 3DSmoothNet [15] designs a rotation-invariant handcraft feature and input it into the network for deep feature learning. SpinNet [2] introduces a spatial point transformer to map the input local surface into the designed cylindrical space, and utilize 3D cylindrical convolutional neural layers to derive a compact feature descriptor. SiamesePointNet [60] produces the descriptor of interest points by a hierarchical encoder-decoder architecture. As for end-to-end learning registration, PointNetLK [3] is a pioneering work that combines PointNet and Lucas&Kanade algorithm into a trainable recurrent deep neural network. Deep Closest Point (DCP) [44] extracts features for each point to compute a soft matching between point clouds, then utilizes a differentiable SVD module to compute rigid transformation parameters which is widely imitated by later work. RPMNet [50] implements a variant of ICP in an end-to-end learnable way. Deep Global Registration (DGR) [25] utilizes fully convolutional geometric features and a weighted Procrustes algorithm for pose estimation. The above methods except DGR are almost designed for object-level point clouds and unsuitable for complex large-scale point clouds. HRegNet[32] is an efficient hierarchical network for large-scale LiDAR

point cloud registration. It hierarchically extracts keypoints and feature descriptors and matches them with bilateral consensus and neighborhood consensus, which could balance the efficiency and registration accuracy. To solve the low-overlap challenge in registration, PREDATOR [22] introduces an overlap attention block which could concentrate on the points in the overlap region and predict the saliency of overlap points. Following PREDATOR's [22], GoeTransformer [38] also tries to solve the low-overlap challenge by introducing a geometric transformer to learn geometric features for robust superpoint matching.

At present, point cloud registration work based on semantic information is relatively rare. A few methods have incorporated deep semantic priors to estimate 6D poses for robot localization and mapping [20, 57], or point cloud registration [31, 51]. These methods establish instance-to-instance correspondence between semantic clusters, where the instances are usually obtained by applying Euclidean point cloud clustering to separate different instances in the same semantic category. However, this method may fail for point cloud scenarios containing few instances. Besides, 3D instance segmentation is still a very challenging task, and the accuracy of instance segmentation is much lower than semantic segmentation [19].

2.3 3D point feature learning

To solve the disorder and unstructured nature of point clouds, it is important to extract discriminative features in most 3D learning tasks, such as classification, segmentation and registration [18]. The pioneering work PointNet [36] and its extension PointNet++ [37] are general frameworks for mapping unorganized points into high-dimensional spaces through feature transformation and aggregation. DGCNN [45] is another milestone as it introduces k-nearest neighbors searching to construct the graph structure in the feature space and dynamically update after each layer of the network. SOCNN [56] is proposed to focus on the representation learning of the underlying shape formed by neighboring points, which combines the global and local context information to get representative features. Different from these methods focusing on geometric or context feature mapping, we aim to design a novel feature extraction module that not only encodes geometric information, but also fully utilizes semantic knowledge.

3 Problem Statement and Overview

The input to SARNet includes a source and a target point cloud $\mathcal{P}_x = \left\{\mathbf{p}_i \in \mathbb{R}^3\right\}_{i=1}^{m}$ and $\mathcal{P}_y = \left\{\mathbf{q}_j \in \mathbb{R}^3\right\}_{j=1}^{n}$, which may be noisy, incomplete, and with non-uniform density distribution. We aim to compute an optimal rigid transformation $\mathbf{T} = [\mathbf{R}, \mathbf{t}; \mathbf{0}, 1]$ to align these two point clouds, *i.e.,* minimize the following objective function:

$$\underset{(\mathbf{M},\mathbf{R},\mathbf{t})}{\operatorname{argmin}} \sum_{i=1}^{m}\sum_{j=1}^{n} \mathbf{M}(i,j) \cdot \left\|\mathbf{R}\mathbf{p}_i + \mathbf{t} - \mathbf{q}_j\right\|_2 \tag{1}$$

where $\mathbf{R} \in SO(3)$ is a rotation matrix and $\mathbf{t} \in \mathbb{R}^3$ is a translation vector. $\mathbf{M}$ is a permutation matrix which maps points in $\mathcal{P}_x$ to $\mathcal{P}_y$.

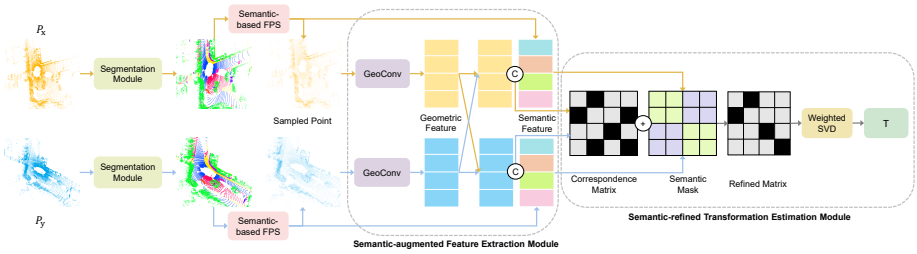

Fig. 2. Network architecture of our proposed SARNet. Given the source and target point clouds taken from the urban scene, we first apply an efficient semantic segmentation to predict per-point semantic labels. Then a *semantic-based FPS module* is proposed to adaptively downsample the clouds as well as remove outliers (*e.g.,* dynamic objects). After that, the *semantic-augmented feature extraction module* extracts high-dimensional features combining semantic and geometric information, where Ⓒ means feature concatenation. Finally, the *semantic-refined transformation estimation module* optimizes the initial point-wise correspondences by filtering semantic-mismatched pairs (the refining operation is represented by ⊕). In the correspondence matrix, the black color indicates corresponding point relationships and the gray color means an incorrect match, while in the semantic mask we use green color to indicate the points belonging to the same semantic class and otherwise use the purple color.

The core idea of our SARNet is that for large-scale point clouds, especially collected in autonomous driving, there are rich easily-gained semantic cues that could help to construct a more accurate correspondence between the source and target. Thus we can effectively integrate both semantic and geometric information for robust registration. The major steps of our algorithm are shown in Fig. 2. We first perform a semantic segmentation on both source and target point clouds to predict per-point semantic labels, which are represented as one-shot vectors. Then we downsample the initial points by designing a new *semantic-based farthest points sampling* approach to obtain a set of key points. Those key points are fed into a *feature extraction module* to obtain high-dimension features that encode both global and local geometric information. We concatenate the geometric features with semantic priors to form the final semantic-augmented features. Next, a *point matching module* is conducted by measuring the feature distance between the source and target to estimate a point-wise corresponding matrix, which can be further refined by using semantic masks. At last, the final transformation is recovered from the corresponding matrix through a weighted *Singular Value Decomposition* (SVD).

4 Methodology

In our approach, we propose to introduce semantics into deep point cloud registration. For both source and target point clouds, we apply a deep neural model, SphereFormer [29], to predict point-wise semantic labels and generate a semantic map. Briefly, SphereFormer [29] achieves new state-of-the-art results of LiDAR-based semantic segmentation by directly aggregating information from dense close points to sparse distant ones. However, this may result in the loss of some crucial features from distant points.

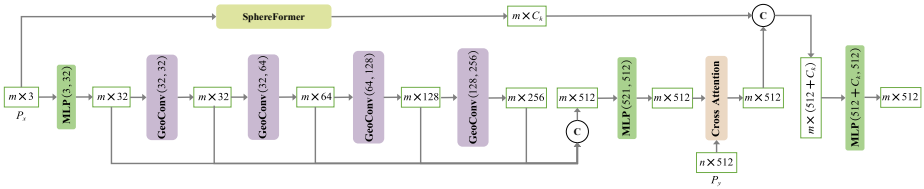

Fig. 3. The structure of our semantic-augmented feature extraction (SAFE) module. Taking the source point cloud $\mathcal{P}_x$ as an example, SAFE first learns the geometric feature. We concatenate all the GeoConv output features to form a 512-dimensional descriptor for each point in $\mathcal{P}_x$, which encodes the local geometry and global contextual information of $\mathcal{P}_x$. To further improve the feature representation ability for registration, SAFE introduces a cross-attention block to learn the structure of the target point cloud $\mathcal{P}_y$. In the last step, we concatenate the crossed-geometric feature with the semantic feature to form the final semantic-augmented feature.

To solve this issue, our network introduces a local feature aggregation module to preserve geometric information. A byproduct of the semantic segmentation is a learned semantic feature for each point, which will be used in later stages. Note that we do not rely on any specific semantic segmentation method. Other advanced segmentors, such as RandLA-Net [21], can also be used.

After segmentation, we could obtain the semantic part-to-part correspondence by measuring the similarity between semantic features. Then the semantic segmentation information is utilized in three ways: (1) In the semantic-based FPS process, we filter moving objects and only sample points in other semantic overlapping regions. We use the semantic proportion defined by the percentage of points number in one semantic category as weight, which means more points will be chosen in larger semantic regions. (2) We concatenate the semantic feature with local and global geometric features to obtain a more robust and representative point descriptor. We will measure the enhanced descriptor distance to generate a correspondence matrix. (3) We utilize the semantic part-to-part correspondence as a mask to refine the correspondence matrix. Inspired by RPM-Net [50], our pipeline is implemented as a recurrent neural network to approximate the ground-truth transformation iteratively. Note that in our approach, the semantic segmentation module and the registration network are trained simultaneously with a weighted combination of the loss terms. In such a way, we do not need to rely on a priori semantic knowledge. Second, we can also use dynamic weights to better balance the relationship between the segmentation and registration networks.

4.1 Semantic-based Farthest Point Sampling

The neural network is unable to handle large-scale point clouds with hundreds of thousands of points. The input is usually downsampled to obtain a spare set of candidate keypoints. Different from previous methods, we design a new weighted farthest point sampling approach by exploiting semantic labels. Our approach is based on the observation that some kinds of objects will be outliers for point cloud registration, especially in realistic outdoor environments. For example, moving cars and humans often appear

in different locations in the source and target point clouds, which will hinder the registration of complex traffic scenes. As a result, if we can filter these dynamic objects, we will build more accurate point mapping. To this end, according to the predicted per-point semantic information, we label the dynamic points as "ignored" by setting their weights to zero in weighted FPS, which means these points will not be chosen for future registration. By checking semantic consistency, we only conduct FPS in semantic overlapping regions between the source and target point clouds except for the moving objects. To further keep the same distribution with initial point clouds, the weight of each point $\mathbf{p}$ belonging to one semantic label corresponds to the number of points in this category. So the final weights of points in weighted FPS can be computed as:

$$W_{\mathbf{p}}^{k} = \begin{cases} N_k/N, \ k \in C_x \cap C_y \text{ and } k \notin C_m \\ 0, \ k \notin C_x \cap C_y \text{ or } k \in C_m \end{cases} \tag{2}$$

where N_k is the number of points belonging to semantic category k, and N is the total number of points in the point cloud. C_x and C_y represent the semantic categories of source and target clouds, respectively. C_m is the semantic category of moving objects.

4.2 Semantic-augmented Feature Extraction

Next, we design a novel *semantic-augmented feature extraction* (SAFE) module. The detailed network architecture of this module is illustrated in Fig. 3, which mainly consists of two components: learning geometry feature $\mathcal{F}^g$ and learning semantic feature $\mathcal{F}^s$. As mentioned above, the semantic feature $\mathcal{F}^s$ can be efficiently learned by SphereFormer when performing semantic segmentation.

Geometric feature embedding. A powerful feature descriptor for point cloud registration should describe the local and global geometric features well. A new *geometric feature embedding* block, called GeoConv, is designed to extract both robust local feature L and global interactive information G. The resulting feature of GeoConv for a point $\mathbf{p}_i$ is thus described as:

$$\mathcal{F}_i^{\text{intra}} = \mathsf{MLP}\left(A_s\left(\mathsf{MLP}\left(L\left(\mathbf{p}_i\right)\right)\right) \oplus \mathsf{MLP}\left(G\left(\mathbf{p}_i\right)\right)\right) \tag{3}$$

where MLP represents a multi-layer perception network, $\oplus$ is the concatenation operator, and A_s is the aggregation function where we choose the maximization operator here.

Fig. 4 shows the structure of our GeoConv module. First, to extract the local feature, we concatenate the feature of $\mathbf{p}_i$ with those of its k-nearest neighbors. We utilize the reciprocal of feature distances between the neighbors and $\mathbf{p}_i$ as weights to measure the contribution of each neighboring point, which means the weight should be higher when the neighbor point is closer to $\mathbf{p}_i$. So the local geometric feature L can be described as:

$$L\left(\mathbf{p}_i\right) = \mathsf{KNN}\left(f_{\mathbf{p}_i}\right) \oplus \mathsf{Rep}\left(f_{\mathbf{p}_i}\right) \oplus \frac{1}{d\left(\mathsf{Rep}\left(f_{\mathbf{p}_i}\right), \mathsf{KNN}\left(f_{\mathbf{p}_i}\right)\right)} \tag{4}$$

where $f_{\mathbf{p}_i}$ means the resultant feature of the prior GeoConv or MLP. $\mathsf{KNN}(f_{\mathbf{p}_i})$ represents the k-nearest neighbors of $f_{\mathbf{p}_i}$ and $\mathsf{Rep}(f_{\mathbf{p}_i})$ means the feature of $\mathbf{p}_i$ is repeated K times. $d(f_a, f_b)$ means the L_2-norm distance between two feature vectors f_a and f_b.

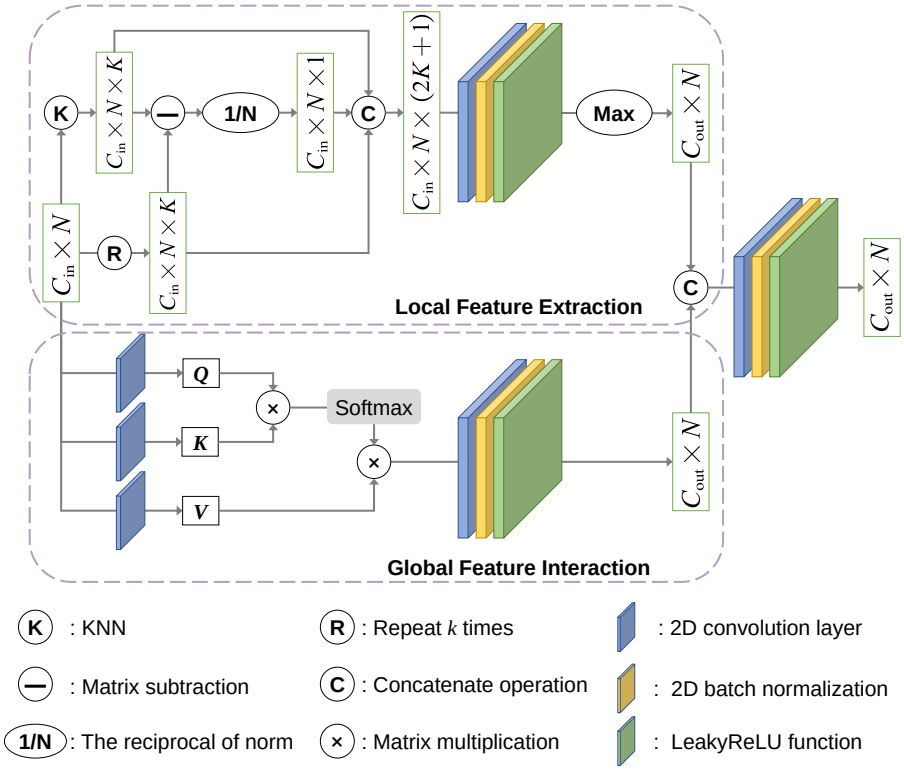

Fig. 4. Illustration of GeoConv. The upper half part mainly extracts local geometric feature for a point $\mathbf{p}_i$ by searching its k-nearest neighbors in the embedded feature space, while the bottom half part utilizes the self-attention mechanism for the interaction of global features extracted from the prior GeoConv.

Second, we consider the interaction between any point pair in one point cloud to describe the global salience of each point $\mathbf{p}_i$. The self-attention mechanism is applied to capture contextual information. In detail, for any two points ($\mathbf{p}_i$ and $\mathbf{p}_j$) in the point cloud, we compute vector-valued keys $k_j \in \mathbb{R}^b$ and queries $q_i \in \mathbb{R}^b$, which are then used to retrieve values $v_j \in \mathbb{R}^b$. The key k_j and value v_j are learned from the feature of $\mathbf{p}_j$ in prior GeoConv or MLP operation, and the query q_i is learned from $\mathbf{p}_i$ in the same way. b is the embedding dimension of the current GeoConv feature. Then the intra-global feature $G(\mathbf{p}_i)$ is a weighted sum of values, where the attention weight a_{ij} assigned to each value is computed by the scaled matrix dot-product of the query with the corresponding key:

$$a_{ij} = \mathsf{Softmax}\left(\frac{q_i^{\mathrm{T}} \cdot k_j}{\sqrt{b}}\right), \quad G\left(\mathbf{p}_i\right) = \sum_{j:(i,j)\in\mathcal{P}} a_{ij} v_j. \tag{5}$$

Inspired by DGCNN [45], we cumulatively utilize GeoConv to embed geometric features into hierarchical feature spaces, and concatenate all the features learned by each GeoConv. Then after applying an MLP function, we can get the feature that encodes both local and global geometric information.

Inter-geometric information perception. The above feature $\mathcal{F}^{\text{intra}}$ only encodes the geometric information in the source or target point cloud independently, but has no knowledge about the structure of the other point cloud. Learning the co-contextual information between the source and target is important for the point cloud registration task. To this end, we introduce a cross-attention block [43] for the inter-geometric information perception which mixes the embedding features of the source and target. Following the work of Deep Closest Point (DCP) [44], the cross-attention block learns a function as a residual term of each embedding feature:

$$\begin{aligned} \mathcal{F}_x^{\text{CA}} &= \mathcal{F}_x^{\text{intra}} + \Theta(\mathcal{F}_x^{\text{intra}}, \mathcal{F}_y^{\text{intra}}), \\ \mathcal{F}_y^{\text{CA}} &= \mathcal{F}_y^{\text{intra}} + \Theta(\mathcal{F}_y^{\text{intra}}, \mathcal{F}_x^{\text{intra}}). \end{aligned} \tag{6}$$

where $\mathcal{F}_x^{\text{intra}}$ and $\mathcal{F}_y^{\text{intra}}$ are the learned high-dimensional feature of source and target, respectively. The asymmetric function $\Theta : \mathbb{R}^{(N\times b)} \times \mathbb{R}^{(N\times b)} \rightarrow \mathbb{R}^{(N\times b)}$ is the cross attention block introduced by the Transformer [43]. Finally, we set the complete geometry feature as $\mathcal{F}_x^g = \mathcal{F}_x^{\text{CA}}, \mathcal{F}_y^g = \mathcal{F}_y^{\text{CA}}$.

Semantic-augmented feature: Since we represent the semantic feature as a one-hot vector, there is no improvement for using cross attention block on the semantic feature. In our approach, the geometric feature after cross attention block will be directly concatenated with the semantic feature to form the final semantic-augmented feature:

$$\mathcal{F} = \mathsf{MLP}(\mathcal{F}^g \oplus \mathcal{F}^s) \tag{7}$$

where $\mathcal{F}^s$ is the semantic feature output by SphereFormer [29].

4.3 Semantic-refined Transformation Estimation

Once obtaining enhanced feature descriptors $\mathcal{F}_x$ and $\mathcal{F}_y$ for the source and target point clouds, we compute an initial point-wise correspondence matrix $\mathbf{M}^{\text{ini}}$ by measuring the similarity between $\mathcal{F}_x$ and $\mathcal{F}_y$. Ideally, $\mathbf{M}^{\text{ini}}(i,j) = 1$ if the point $\mathbf{p}_i \in \mathcal{P}_x$ exactly corresponds to $\mathbf{q}_j \in \mathcal{P}_y$ and $\mathbf{M}^{\text{ini}}(i,j) = 0$ otherwise. However, such a discrete permutation matrix is undifferentiable, causing the gradient to not be backpropagated for network training. Besides, due to the presence of noise in LiDAR scanning, the points after alignment may not coincide exactly. Therefore, instead of the one-to-one correspondence, we assume that one point in the source corresponds to the weighted mean of points in the target. In our approach, the correspondence matrix constraint is relaxed to a doubly stochastic constraint: $\sum_{i=1}^{m} \mathbf{M}^{\text{ini}}(i,j) = 1, \sum_{j=1}^{n} \mathbf{M}^{\text{ini}}(i,j) = 1, \mathbf{M}^{\text{ini}}(i,j) \in [0,1]$.

Since there are a lot of similar local geometric structures (*e.g.,* a plane in the road and a plane in a building) in the scene, the initial correspondence matrix $\mathbf{M}^{\text{ini}}$ may have many incorrect matches. To improve the matching accuracy while reducing the searching space of pose hypotheses, we quickly prune bad hypotheses based on semantics.

We find that if two points are labeled in different semantic categories, we do not need to build a correspondence between them, *i.e.,* we would only attempt to match keypoints that belong to the same semantic category. Therefore, we compute a semantic correspondence matrix $\mathbf{M}^S = \{0,1\}^{I,J}$ as a mask by multiplying the one-hot semantic features, which can be written as:

$$\mathbf{M}^s(i,j) = \begin{cases} 1, & \text{if } C_x(i) = C_y(j) \\ 0, & \text{if } C_x(i) \neq C_y(j) \end{cases} \tag{8}$$

Then we assign the value $\mathbf{M}^{\text{ini}}(i,j)$ to the negative-infinity where $\mathbf{M}^s(i,j) = 0$. After applying a Softmax function, those negative-infinity values in corresponding matrix will be zero:

$$\mathbf{M}^{\text{ini}}(i,j) = \begin{cases} -\infty, & \text{if } \mathbf{M}^s(i,j) = 0 \\ \mathbf{M}^{\text{ini}}(i,j), & \text{if } \mathbf{M}^s(i,j) = 1 \end{cases} \tag{9}$$

So the final correspondence matrix becomes:

$$\mathbf{M}\left(i,j\right) = \frac{\exp\left(\mathbf{M}^{\text{ini}}\left(i,j\right)\right)}{\sum_{j=1}^{N} \exp\left(\mathbf{M}^{\text{ini}}\left(i,j\right)\right)} \tag{10}$$

Finally, we utilize $\mathbf{M}$ to compute the corresponding point $\tilde{y}_i$ in $\mathcal{P}_y$ for each point x_i in $\mathcal{P}_x$ to obtain $\tilde{\mathcal{P}}_y$:

$$\tilde{y}_i = \sum_{j=1}^{N} \mathbf{M}(i,j) \cdot y_j \tag{11}$$

Therefore, we can build the cross-covariance matrix by using $\mathcal{P}_x$ and $\tilde{\mathcal{P}}_y$ [44], which is then decomposed by *weighted singular value decomposition* (WSVD) to estimate the transformation $\mathbf{T}$. Most recent deep learning-based registration methods compute weights of SVD by directly operating on the initial features $\mathcal{F}_x$ and $\mathcal{F}_y$. However, we find that the point-wise order in $\mathcal{F}_x$ doesn't correspond to that in $\mathcal{F}_y$, thus the computed weights will hinder to obtain the optimal transformation T as the learning process goes on. Differently, we first use $\mathbf{M}$ to adjust the point-wise order in $\mathcal{F}_y$, then we concatenate $\mathcal{F}_x$ with the adjusted $\mathcal{F}_y$ and feed them into a MLP to compute the final weights:

$$W = \mathsf{MLP}(\mathcal{F}_x \oplus \mathbf{M} \cdot \mathcal{F}_y) \tag{12}$$

4.4 Loss Functions

Our SARNet model integrates the above-described modules in a unified end-to-end network architecture as shown in Fig. 2, which is trained in a supervised fashion by an efficient joint loss function. We use a transformation loss for registration and a cross-entropy loss function for semantic segmentation. Different from previous registration methods, we use the uncertainty weighting method proposed in [27] to combine the transformation loss and the semantic loss which could automatically balance the two tasks instead of complicated manual attempts during end-to-end training.

Transformation loss. our transformation loss is the L_2-norm distance between the source point cloud $\mathcal{P}_x$ transformed using the ground-truth transformation $\{\mathbf{R}_{\text{gt}}, \mathbf{t}_{\text{gt}}\}$ and the predicted transformation $\{\mathbf{R}', \mathbf{t}'\}$. Since we design the registration network as a recurrent neural network, we need to compute the loss for every iteration k. Following RPMNet [50], we weigh the losses by $(1/2)^{K-k}$ to give later iterations higher weights, where K is the total number of iterations:

$$\mathcal{L}_k^{\text{trans}} = \frac{1}{N} \sum_{i=1}^{N} \sqrt{\left[(\mathbf{R}_{\text{gt}}\mathbf{p}_i + \mathbf{t}_{\text{gt}}) - (\mathbf{R}'_k\mathbf{p}_i + \mathbf{t}'_k)\right]^2} \tag{13}$$

$$\mathcal{L}^{\text{trans}} = \sum_{k=1}^{K} \frac{1}{2^{K-k}} \cdot \mathcal{L}_k^{\text{trans}} \tag{14}$$

where N is the number of points in the source point cloud, $\{\mathbf{R}'_k, \mathbf{t}'_k\}$ is the predicted transformation in the k-th iteration.

Semantic loss. we choose the cross-entropy loss function to supervise the learning of semantic segmentation module. Given the ground-truth semantic labels $C_x,\ C_y \in R$ with the learned source features $\mathcal{F}_x \in R^b$ and target features $\mathcal{F}_y \in R^b$, the semantic loss can be described as:

$$\mathcal{L}^{\text{sem}} = -\log\left(\frac{e^{\mathcal{F}_x[C_x]}}{\sum_{i=0}^{c-1} e^{\mathcal{F}_x[i]}}\right) - \log\left(\frac{e^{\mathcal{F}_y[C_y]}}{\sum_{j=0}^{c-1} e^{\mathcal{F}_y[j]}}\right) \tag{15}$$

where c is the total number of semantic classes. Since the semantic segmentation module is conducted before the semantic-based FPS and doesn't need an iteration process, the semantic loss will be computed only once.

Total loss. We find that transformation loss and semantic loss have different convergence speed during the training process. It is difficult for us to find suitable fixed parameters, especially the convergence speed of cross-entropy loss is changing. To solve this problem, we use the learnable parameters σ_t and σ_s to weigh the transformation loss and semantic loss respectively for optimal trade-off. To maintain their stability, the uncertainty can be reduced as two $\log$ terms:

$$\mathcal{L}^{\text{total}} = \frac{1}{2\sigma_t^2} \cdot \mathcal{L}^{\text{trans}} + \frac{1}{2\sigma_s^2} \cdot \mathcal{L}^{\text{sem}} + \log(\sigma_t) + \log(\sigma_s) \tag{16}$$

4.5 Implementation Details

Our SARNet is implemented in PyTorch on a server equipped with an Intel Xeon Gold 6226R CPU and NVIDIA RTX 2080 Ti (11 GB memory) graphics cards. We train the entire neural network end-to-end on our training dataset for 60 epochs with Adam optimizer [28]. The initial learning rate is set to 10^{-4} and reduced with an attenuation coefficient of 0.5 every 10 epochs. The initial learnable parameters σ_t and σ_s are randomly sampled from the continuous uniform distribution in the range of $[0.2, 1.0]$. The training task is completed in about 40 hours.

5 Experimental Results

In this section, we demonstrate the effectiveness of our approach on two real-world outdoor LiDAR datasets. We also evaluate the proposed algorithm qualitatively and quantitatively through the visual inspection of our results and a comparison with traditional registration approaches and state-of-the-art deep learning-based methods.

5.1 Experimental Setup

Datasets. We carry out experiments on two large-scale outdoor LiDAR point cloud datasets, namely SemanticKITTI dataset [5] and NuScenes dataset [8], which are most commonly used for 3D point cloud recognition. The SemanticKITTI dataset consists of 43552 densely annotated LiDAR scans. We select 11 sequences with ground-truth annotation for our experiments. To keep the semantic consistency between the training and testing datasets, we choose the first 70% of point clouds in each sequence as training data, the next 10% for validation, and the final 20% for testing. In total, we obtain 16233 frames for training, 2319 frames for validation, and 4649 frames for testing. The NuScenes dataset has 34149 scans with semantic annotation which belongs to 850 scenes (on average there are 40 or 41 scans in each scene). We separate the point clouds in every scene into training set (75%), validation set (10%), and testing set (15%). Finally, we get 25500 scans for training, 3400 scans for validation and 5200 scans for testing.

Pre-processing. To maintain the consistency of semantic annotation, we generate the source and target point clouds by adding random rotations and translations on each point cloud during the training and testing stages. Specifically, for an original point cloud $\mathcal{P}$ in the dataset, along each coordinate axis, we randomly apply rotations $(\mathbf{R}_x, \mathbf{R}_y, \mathbf{R}_z)$ with angles in the range of $[0, 45^\circ]$, and provide translations $(\mathbf{t}_x, \mathbf{t}_y, \mathbf{t}_z)$ within the range $[-5, 5]$ meters. As a result, the rigid transformation applied to $\mathcal{P}$ can be computed as rotation matrix $\mathbf{R} = \mathbf{R}_x \cdot \mathbf{R}_y \cdot \mathbf{R}_z$ with a translation vector $\mathbf{t} = \mathbf{t}_x + \mathbf{t}_y + \mathbf{t}_z$. We record $\mathbf{R}$ and $\mathbf{t}$ as the ground truth for subsequent evaluations. To simulate the moving objects in the real world which should appear in different positions in the source and target point clouds, we specially add another random offset to the points belonging to the class of moving objects. To this end, we randomly sample Euler angle rotations in the range $[0, 3^\circ]$ along each axis, and apply random translations in the range $[-5, 5]$ meters along the x-axis, $[-1, 1]$ meters along the y-axis and $[-0.1, 0.1]$ meters along the z-axis.

Furthermore, we shuffle the points order and add random Gaussian noise to both source and target in training data, where the noise is relative to the size of point clouds. For SemanticKITTI, the average diagonal of the bounding box for each point cloud is about 102 meters, so the noise for each point is sampled independently from the distribution $\mathcal{N}(0, 0.01 * 102)$ and will be clipped to $[-0.05 * 102, 0.05 * 102]$ on each axis. Similarly, the average diagonal of NuScenes is 80 meters, so the distribution of sampled noises is $\mathcal{N}(0, 0.01 * 80)$ and clipped to $[-0.05 * 80, 0.05 * 80]$.

Table 1. Quantitative performance comparison of different registration methods on SemanticKITTI dataset and NuScenes dataset. The best results are marked in red color, and the second best results are in blue color. The symbol '-' means the results are unavailable.

Methods	SemanticKITTI				Nuscenes			
	RRE (deg) ↓	RTE (m) ↓	Recall ↑	Time (ms)	RRE (deg) ↓	RTE (m) ↓	Recall ↑	Time (ms)
FGR	0.77 ± 0.44	0.26 ± 0.12	36.7%	506.1	1.14 ± 0.46	0.27 ± 0.12	19.6%	284.6
RANSAC	0.80 ± 0.45	0.22 ± 0.10	67.3%	549.6	0.78 ± 0.47	0.21 ± 0.10	74.2%	268.2
DCP	0.84 ± 0.46	0.30 ± 0.11	40.0%	**46.4**	1.04 ± 0.48	0.29 ± 0.12	38.9%	**45.5**
FMR	0.60 ± 0.34	0.22 ± 0.15	78.9%	85.5	0.64 ± 0.39	0.23 ± 0.11	75.3%	61.1
DGR	0.32 ± 0.25	0.17 ± 0.11	95.2%	1496.6	0.24 ± 0.22	0.15 ± 0.11	94.9%	523.0
HRegNet	**0.23 ± 0.21**	**0.12 ± 0.10**	96.9%	106.2	**0.14 ± 0.08**	**0.07 ± 0.04**	**98.5%**	87.3
Segregator	0.27 ± 0.18	0.12 ± 0.11	**99.0%**	150	-	-	-	-
Ours	**0.11 ± 0.10**	**0.09 ± 0.06**	**99.7%**	**83.9**	**0.05 ± 0.04**	**0.05 ± 0.03**	**100%**	**58.4**

5.2 Evaluation Metrics

To determine the registration accuracy, we measure the deviation between the predicted values and the ground-truth values by using the metrics of *relative rotation error* (RRE) and *relative translation error* (RTE). RRE is computed as:

$$\text{RRE} = \arccos\left(\frac{1}{2} \cdot \left(\text{Tr}\left(\mathbf{R}_{\text{gt}}^{-1}\hat{\mathbf{R}}\right) - 1\right)\right) \tag{17}$$

where $\mathbf{R}_{\text{gt}}$ and $\hat{\mathbf{R}}$ are the ground-truth and estimated rotation matrices, respectively. Tr is a function for calculating the trace of a matrix and arccos represents the arc cosine function. RTE can be calculated as:

$$\text{RTE} = \left\|\mathbf{t}_{\text{gt}} - \hat{\mathbf{t}}\right\|_2 \tag{18}$$

where $\mathbf{t}_{gt}$ and $\hat{\mathbf{t}}$ are the ground-truth and estimated translation vectors. Obviously, the closer the RRE and RTE are to 0, the more accurate the predicted values.

After defining the error functions, we define the registration recall as the ratio of successful registration, where a transformation is accepted as positive if the RRE and RTE are within the thresholds ξ_r and ξ_t. We set $\xi_r = 2(\text{deg})$ and $\xi_t = 0.5(\text{m})$ as default values in all of our experiments. We find that some failed registrations can cause dramatically large RRE and RTE, which will result in unreliable error metrics. Therefore, we only calculate RRE and RTE for successful registrations. In addition, we also record the mean and the standard deviation of RRE and RTE as the form of $\pm$ in the results.

5.3 Comparisons

We thoroughly compare our method against various registration competitive methods. For traditional competitors, we select two representative global registration methods, *Fast Global Registration* (FGR) [61] and RANSAC [13], because local registration

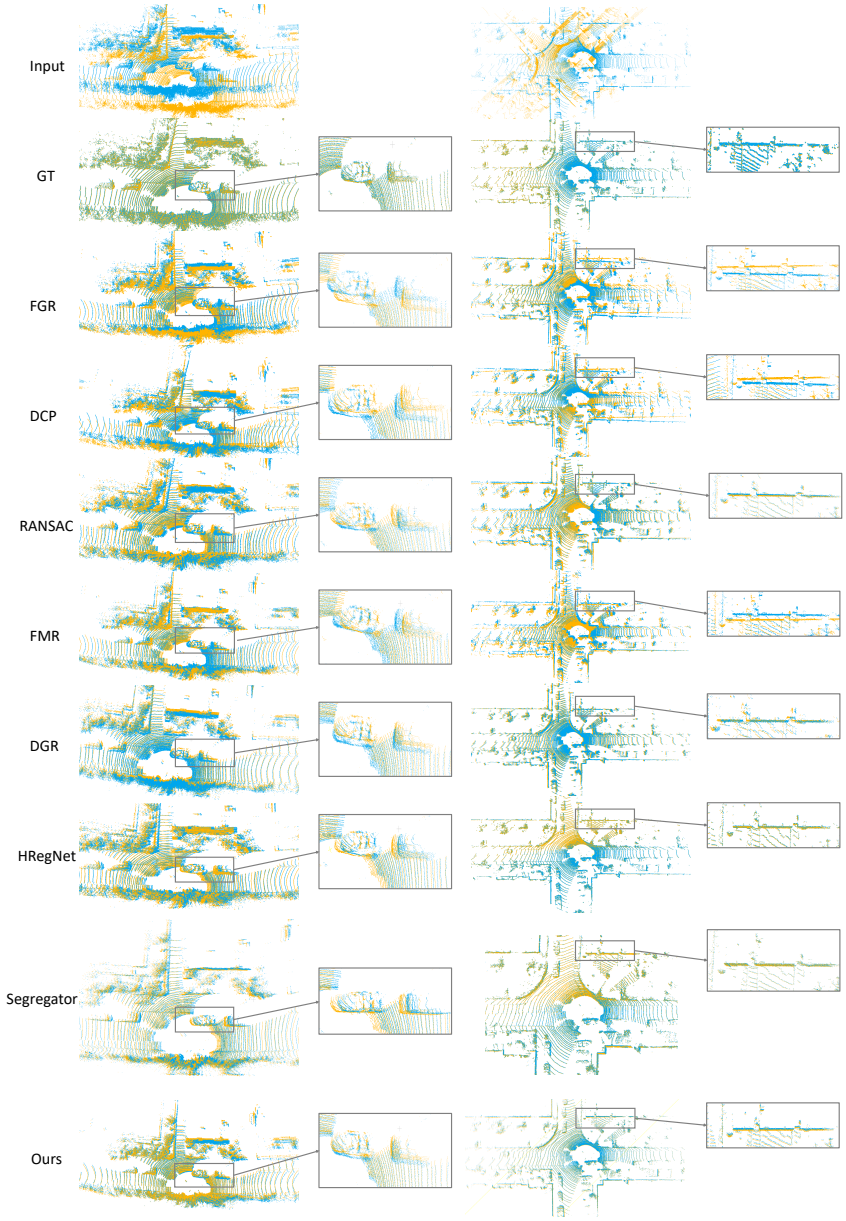

Fig. 5. Visual comparison of different point cloud registration methods using SemanticKITTI dataset. The first row shows the input source and target point clouds, and the second row shows the ground-truth registration result. Then the results of all comparison methods are sorted by the registration recall. The detailed comparisons are shown in the zoomed-in insets.

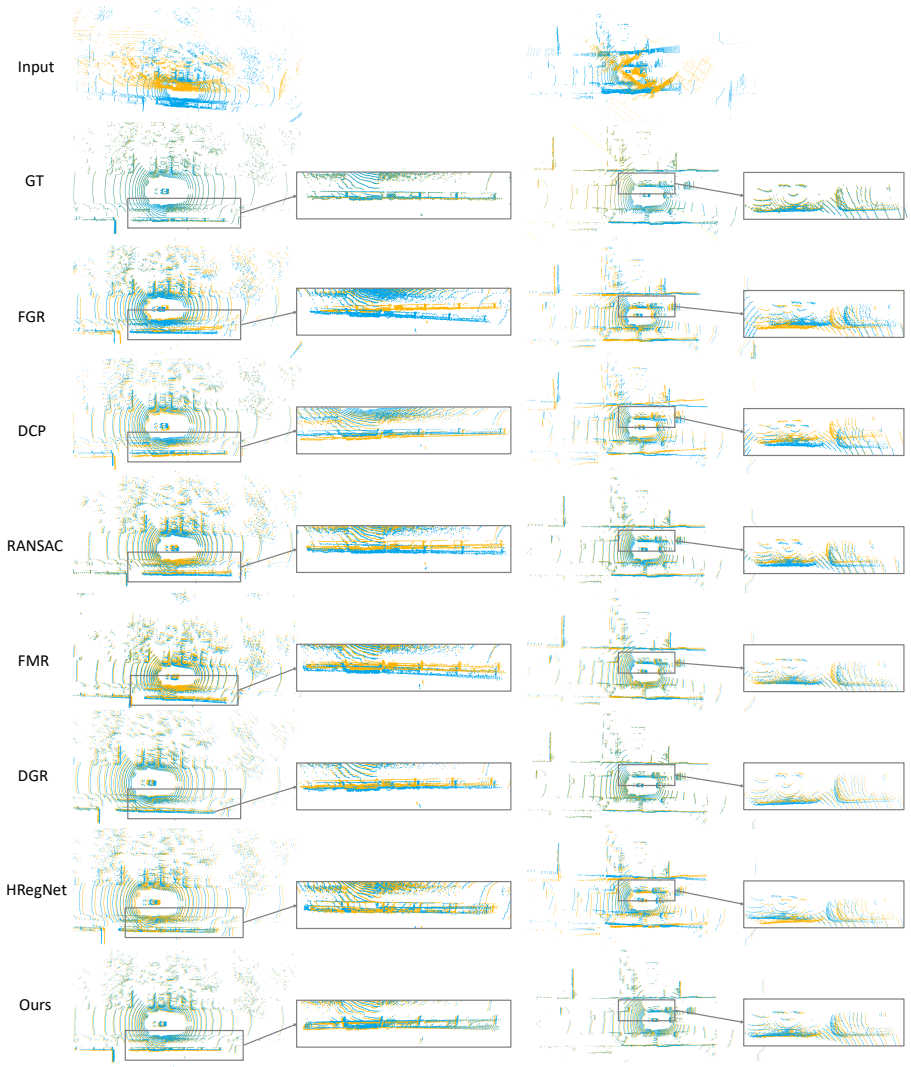

Fig. 6. Visual comparison of different point cloud registration methods using NuScenes dataset. The first row shows the input source and target point clouds, and the second row shows the ground-truth registration result. Then the results of all comparison methods are sorted by the registration recall.

methods like ICP usually fail to generate reasonable transformation if the source and target are not initially aligned. FGR and RANSAC are implemented using Open3D library [62], where we need to compute the *Fast Point Feature Histograms* (FPFH)[40] features in the same voxel size. In addition, we compare our method to four recent deep learning-based methods, including *Deep Closest Point* (DCP) [44], *Feature-metric Reg-*

istration (FMR) [23], *Deep Global Registration* (DGR) [9], and HRegNet [32]. Finally, we compare with Segregator [51], which is a semantic global point cloud registration framework using both semantic and geometric information. These methods provide a plethora of comparisons to other techniques and establish themselves as state-of-the-art methods. Not that Segregator's code compilation, parameter configuration, and data flow processing all focus on the KITTI dataset and do not support other datasets. Therefore we only report the performance of Segregator on SemanticKITTI.

Fig. 5 and Fig. 6 show the qualitative comparison results on several scenes selected from SemanticKITTI and NuScenes datasets, respectively. The numerical statistics of each method on the whole dataset are reported in Table 1. From both qualitative and quantitative comparisons, it can be seen that FGR generates large rotation and translation errors, and the registration recall is below 40% on SemanticKITTI and even worse on the Nuscenes dataset. RANSAC can achieve a relatively good performance thanks to its powerful outlier rejection mechanism, but it is still worse than most learning-based methods except DCP.

As for the learning-based methods, the recall rate of DCP on SemanticKITTI and Nuscenes dataset are both less than 40% and the average of RRE and RTE is quite large. FMR performs well to some extent and its recall is more than 70% on both datasets, however, it gets pretty large RRE and RTE compared to our methods. DGR could achieve a quite good performance in terms of recall rate. But it is still far from our methods in terms of RRE and RTE, because the voxel-based representation of point clouds in DGR limits the accuracy of the registration. HRegNet obtains a similar average RTE to ours on both SemanticKITTI and NuScenes, while its average RRE is more than twice as large as that of our proposed method. In terms of the recall rate, we still outperform HRegNet. In addition, to pursue high registration accuracy, HRegNet designs a quite complex neural network and consumes a large time and memory to train the network. It also has to pre-train its key point detection module and feature extraction module in two periods. We spent four days training HRegNet and it took less than two days to train our network.

Segregator shares a similar idea with ours by exploiting both semantic information and geometric distribution to build up outlier-robust correspondences and search for inliers. It achieves a good recall rate, but the value of RRE is much worse than ours. Besides, Segregator is a non-deep learning method and does not fully utilize the ability of point feature learning. Finally, our method creatively introduces semantic constraints into the deep registration process and achieves state-of-the-art performance compared to previous alternatives.

5.4 Ablation Study

Finally, we conduct experiments on the SemanticKITTI dataset to evaluate the influence of different components of our designed network. To demonstrate the effectiveness of semantic information for registration, we progressively add the proposed modules to a baseline network, and prove the functionality of semantic-based FPS, semantic-augmented feature extraction module and semantic-refined transformation estimation module, respectively. To build the baseline network, we use traditional FPS for

Table 2. Ablation studies on SemanticKITTI dataset. The best result of each measurement is marked in **bold** font.

Module	RRE (deg)	RTE (m)	Recall
baseline network	0.33 ± 0.28	0.30 ± 0.12	37.6%
baseline+SFPS	0.28 ± 0.25	0.28 ± 0.12	52.8%
baseline+SFPS+SAFE	0.19 ± 0.18	0.22 ± 0.11	89.3%
baseline+SFPS+SAFE+SRTE	$\mathbf{0.11 \pm 0.10}$	$\mathbf{0.09 \pm 0.06}$	$\mathbf{99.7\%}$

points down-sampling, DGCNN [45] for feature extraction, and correspondence indicator without semantic mask.

Semantic-based FPS. We validate the effectiveness of the semantic-based FPS (SFPS) by comparing it to the conventional FPS. We divide the points in SemanticKITTI into 20 categories by remapping the original semantic label of points to 20 different kinds. We exclude the moving-object categories because they will hinder the registration. As shown in Table 2, the strategy of using SFPS slightly decreases the mean of RRE and RTE, while improving registration recall by almost 15% than the baseline network.

Semantic-augmented feature extraction. After adding the SFPS module, instead of using the popular DGCNN to embed high-dimensional features, we use our designed semantic-augmented feature extraction module (SAFE) to learn point features. SAFE can generate new features that encode both semantic and geometric information. According to the results, the SAFE module significantly improves registration recall from 52.8% to 89.3%, and makes great progress on RRE from 0.28 ± 0.25 to 0.19 ± 0.18.

Semantic-refined transformation estimation. To demonstrate the ability of the semantic-refined transformation estimation module (SRTE), we compare it to the traditional rigid transformation estimation approach. We find SRTE could not only improve the registration precision greatly but also speed up the training process. Our network can converge within 60 epochs when using the proposed SRTE module and converge until 100 epochs using the traditional transformation estimation module.

Impact of semantic label deterioration. Finally, we investigate the performance of SARNet against semantic label deterioration, *i.e.,* , we explore the impact of different semantic segmentation accuracies on the performance of the entire registration network. To this end, we gradually increase the noise ratio in the semantic label predicted by SphereFormer [29] by randomly setting part of predicted true labels as 'false'. Fig. 7 shows that as semantic segmentation accuracy decreases, the point cloud registration accuracy also decreases, highlighting the effectiveness of semantic information in the point cloud registration task. However, when the mean Intersection over Union (mIoU) is greater than 60%, the proposed method can provide satisfactory registration results, indicating the robustness of our method to semantic segmentation performance.

5.5 Limitations

We successfully applied our method for registering large-scale urban point clouds. However, there still exist several limitations in our SARNet. First, our registration pro-

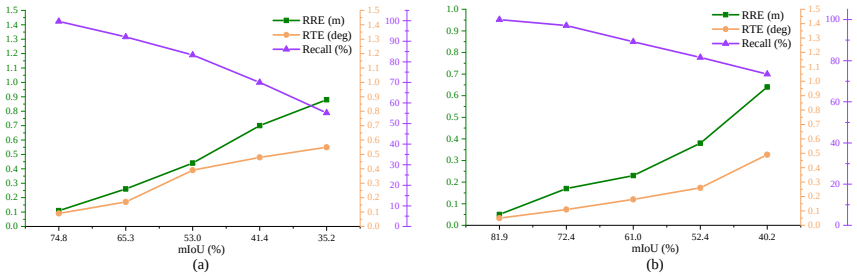

Fig. 7. The impact of varying semantic segmentation accuracy on point cloud registration performance. (a) and (b) represent the evaluation metrics on the SemanticKITTI and the Nuscenes test set, respectively.

cess partially depends on good semantic segmentation. Therefore our registration accuracy drops when we can not obtain a good segmentation, especially for scenarios under very challenging semantic mask deterioration (*e.g.,* , when semantic segmentation accuracy is lower than 40%, see Fig. 7). Therefore, we may have poorer generalization performance compared to other geometric-only methods. Second, using semantic information can improve the registration accuracy for the testing scenes in which the object classes are similar to the training set. Thus, object types that do not exist in the training dataset can not be precisely segmented, leading to unsatisfactory registration results. Fortunately, the semantic categories of common urban objects are limited, and enriching the training dataset can partially solve this problem.

6 Conclusion and Future Work

In this paper, we have proposed a new deep neural network for large-scale outdoor point cloud registration. Our key idea is to introduce semantic information to improve both registration accuracy and efficiency. We design three neural modules for taking full advantage of semantic information in points sampling, feature extraction, and rigid transformation estimation. We demonstrate the effectiveness and advantages of our approach by ablation studies and comparing it to state-of-the-art methods on real-world data.

In future work, instead of just using semantic information, we would like to construct a more general registration framework that utilizes part-to-part structural correspondences, such as planes or cylinders. In addition, we have verified the enhancement of semantic segmentation for registration, and we will explore the possibility that point cloud registration promotes the segmentation in turn. We intend to achieve iterative promotion for both registration and segmentation.

Acknowledgments

This work is partially funded by the National Natural Science Foundation of China (62172416, U22B2034, U21A20515, 61972388, 62262043, 62376271), Shenzhen Sci-

ence and Technology Program (GJHZ20210705141402008), and Beijing Natural Science Foundation (L231013).

References

1. Aiger, D., Mitra, N.J., Cohen-Or, D.: 4-points congruent sets for robust pairwise surface registration. In: ACM Trans. Graph. (Proc. SIGGRAPH), pp. 1–10 (2008)
2. Ao, S., Hu, Q., Yang, B., Markham, A., Guo, Y.: Spinnet: Learning a general surface descriptor for 3d point cloud registration. In: IEEE Computer Vision and Pattern Recognition (CVPR). pp. 11753–11762 (2021)
3. Aoki, Y., Goforth, H., Srivatsan, R.A., Lucey, S.: Pointnetlk: Robust & efficient point cloud registration using pointnet. In: IEEE Computer Vision and Pattern Recognition (CVPR). pp. 7163–7172 (2019)
4. Arvanitis, G., Zacharaki, E.I., Váša, L., Moustakas, K.: Broad-to-narrow registration and identification of 3d objects in partially scanned and cluttered point clouds. IEEE Trans. Multimedia **24**, 2230–2245 (2021)
5. Behley, J., Garbade, M., Milioto, A., Quenzel, J., Behnke, S., Stachniss, C., Gall, J.: Semantickitti: A dataset for semantic scene understanding of lidar sequences. In: IEEE International Conference on Computer Vision (ICCV). pp. 9297–9307 (2019)
6. Besl, P.J., McKay, N.D.: Method for registration of 3-d shapes. In: Sensor fusion IV: control paradigms and data structures. vol. 1611, pp. 586–606. Spie (1992)
7. Billings, S.D., Boctor, E.M., Taylor, R.H.: Iterative most-likely point registration (imlp): A robust algorithm for computing optimal shape alignment. PloS one **10**(3), e0117688 (2015)
8. Caesar, H., Bankiti, V., Lang, A.H., Vora, S., Liong, V.E., Xu, Q., Krishnan, A., Pan, Y., Baldan, G., Beijbom, O.: nuscenes: A multimodal dataset for autonomous driving. In: IEEE Computer Vision and Pattern Recognition (CVPR). pp. 11621–11631 (2020)
9. Choy, C., Dong, W., Koltun, V.: Deep global registration. In: IEEE Computer Vision and Pattern Recognition (CVPR). pp. 2514–2523 (2020)
10. Deng, H., Birdal, T., Ilic, S.: Ppfnet: Global context aware local features for robust 3d point matching. In: IEEE Computer Vision and Pattern Recognition (CVPR). pp. 195–205 (2018)
11. Drost, B., Ulrich, M., Navab, N., Ilic, S.: Model globally, match locally: Efficient and robust 3d object recognition. In: IEEE Computer Vision and Pattern Recognition (CVPR). pp. 998–1005. Ieee (2010)
12. Duchenne, O., Bach, F., Kweon, I.S., Ponce, J.: A tensor-based algorithm for high-order graph matching. IEEE Trans. Pattern Anal. Mach. Intell. **33**(12), 2383–2395 (2011)
13. Fischler, M.A., Bolles, R.C.: Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. Communications of the ACM **24**(6), 381–395 (1981)
14. Godin, G., Rioux, M., Baribeau, R.: Three-dimensional registration using range and intensity information. In: Videometrics III. vol. 2350, pp. 279–290. International Society for Optics and Photonics (1994)
15. Gojcic, Z., Zhou, C., Wegner, J.D., Wieser, A.: The perfect match: 3d point cloud matching with smoothed densities. In: IEEE Computer Vision and Pattern Recognition (CVPR). pp. 5545–5554 (2019)
16. Guo, J., Xing, X., Quan, W., Yan, D.M., Gu, Q., Liu, Y., Zhang, X.: Efficient center voting for object detection and 6d pose estimation in 3d point cloud. IEEE Trans. Image Process. **30**, 5072–5084 (2021)
17. Guo, Y., Sohel, F., Bennamoun, M., Wan, J., Lu, M.: An accurate and robust range image registration algorithm for 3d object modeling. IEEE Trans. Multimedia **16**(5), 1377–1390 (2014)

18. Guo, Y., Wang, H., Hu, Q., Liu, H., Liu, L., Bennamoun, M.: Deep learning for 3d point clouds: A survey. IEEE Trans. Pattern Anal. Mach. Intell. **43**(12), 4338–4364 (2020)
19. He, Y., Yu, H., Liu, X., Yang, Z., Sun, W., Wang, Y., Fu, Q., Zou, Y., Mian, A.: Deep learning based 3d segmentation: A survey. arXiv preprint arXiv:2103.05423 (2021)
20. He, Y., Ma, L., Jiang, Z., Tang, Y., Xing, G.: Vi-eye: Semantic-based 3d point cloud registration for infrastructure-assisted autonomous driving. In: Proceedings of International Conference on Mobile Computing and Networking. pp. 573–586 (2021)
21. Hu, Q., Yang, B., Xie, L., Rosa, S., Guo, Y., Wang, Z., Trigoni, N., Markham, A.: Randlanet: Efficient semantic segmentation of large-scale point clouds. In: IEEE Computer Vision and Pattern Recognition (CVPR). pp. 11108–11117 (2020)
22. Huang, S., Gojcic, Z., Usvyatsov, M., Wieser, A., Schindler, K.: Predator: Registration of 3d point clouds with low overlap. In: IEEE Computer Vision and Pattern Recognition (CVPR). pp. 4267–4276 (2021)
23. Huang, X., Mei, G., Zhang, J.: Feature-metric registration: A fast semi-supervised approach for robust point cloud registration without correspondences. In: IEEE Computer Vision and Pattern Recognition (CVPR). pp. 11366–11374 (2020)
24. Huang, X., Mei, G., Zhang, J., Abbas, R.: A comprehensive survey on point cloud registration. arXiv preprint arXiv:2103.02690 (2021)
25. Iglesias, J.P., Olsson, C., Kahl, F.: Global optimality for point set registration using semidefinite programming. In: IEEE Computer Vision and Pattern Recognition (CVPR). pp. 8287–8295 (2020)
26. Johnson, A.E.: Spin-images: a representation for 3-d surface matching (1997)
27. Kendall, A., Gal, Y., Cipolla, R.: Multi-task learning using uncertainty to weigh losses for scene geometry and semantics. In: IEEE Computer Vision and Pattern Recognition (CVPR). pp. 7482–7491 (2018)
28. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980 (2014)
29. Lai, X., Chen, Y., Lu, F., Liu, J., Jia, J.: Spherical transformer for lidar-based 3d recognition. In: IEEE Computer Vision and Pattern Recognition (CVPR). pp. 17545–17555 (2023)
30. Liu, H.Y., Guo, J.W., Jiang, H.Y., Liu, Y.C., Zhang, X.P., Yan, D.M.: Puzzlenet: Boundary-aware feature matching for non-overlapping 3d point clouds assembly. Journal of Computer Science and Technology **38**(3), 492–509 (2023)
31. Liu, S., Wang, T., Zhang, Y., Zhou, R., Li, L., Dai, C., Zhang, Y., Wang, H.: Deep semantic graph matching for large-scale outdoor point clouds registration. arXiv preprint arXiv:2308.05314 (2023)
32. Lu, F., Chen, G., Liu, Y., Zhang, L., Qu, S., Liu, S., Gu, R.: Hregnet: A hierarchical network for large-scale outdoor lidar point cloud registration. In: IEEE International Conference on Computer Vision (ICCV). pp. 16014–16023 (2021)
33. Mellado, N., Aiger, D., Mitra, N.J.: Super 4pcs fast global pointcloud registration via smart indexing. In: Comp. Graph. Forum. vol. 33, pp. 205–215. Wiley Online Library (2014)
34. Nüchter, A., Wulf, O., Lingemann, K., Hertzberg, J., Wagner, B., Surmann, H.: 3d mapping with semantic knowledge. In: Robot Soccer World Cup. pp. 335–346. Springer (2005)
35. Pais, G.D., Ramalingam, S., Govindu, V.M., Nascimento, J.C., Chellappa, R., Miraldo, P.: 3dregnet: A deep neural network for 3d point registration. In: IEEE Computer Vision and Pattern Recognition (CVPR). pp. 7193–7203 (2020)
36. Qi, C.R., Su, H., Mo, K., Guibas, L.J.: Pointnet: Deep learning on point sets for 3d classification and segmentation. In: IEEE Computer Vision and Pattern Recognition (CVPR). pp. 652–660 (2017)
37. Qi, C.R., Yi, L., Su, H., Guibas, L.J.: Pointnet++: Deep hierarchical feature learning on point sets in a metric space. Advances in neural information processing systems **30** (2017)

38. Qin, Z., Yu, H., Wang, C., Guo, Y., Peng, Y., Xu, K.: Geometric transformer for fast and robust point cloud registration. arXiv preprint arXiv:2202.06688 (2022)
39. Rusinkiewicz, S., Levoy, M.: Efficient variants of the icp algorithm. In: Proceedings third international conference on 3-D digital imaging and modeling. pp. 145–152. IEEE (2001)
40. Rusu, R.B., Blodow, N., Beetz, M.: Fast point feature histograms (fpfh) for 3d registration. In: IEEE international conference on robotics and automation. pp. 3212–3217 (2009)
41. Tombari, F., Salti, S., Stefano, L.D.: Unique signatures of histograms for local surface description. In: European Conference on Computer Vision (ECCV). pp. 356–369. Springer
42. Truong, G., Gilani, S.Z., Islam, S.M.S., Suter, D.: Fast point cloud registration using semantic segmentation. In: 2019 Digital Image Computing: Techniques and Applications (DICTA). pp. 1–8. IEEE (2019)
43. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł., Polosukhin, I.: Attention is all you need. In: Advances in neural information processing systems. vol. 30 (2017)
44. Wang, Y., Solomon, J.M.: Deep closest point: Learning representations for point cloud registration. In: IEEE International Conference on Computer Vision (ICCV). pp. 3523–3532 (2019)
45. Wang, Y., Sun, Y., Liu, Z., Sarma, S.E., Bronstein, M.M., Solomon, J.M.: Dynamic graph cnn for learning on point clouds. ACM Trans. Graph. **38**(5), 1–12 (2019)
46. Xing, X., Guo, J., Nan, L., Gu, Q., Zhang, X., Yan, D.M.: Efficient mspso sampling for object detection and 6-d pose estimation in 3-d scenes. IEEE Trans. Ind. Electron. **69**(10), 10281–10291 (2021)
47. Yang, J., Quan, S., Wang, P., Zhang, Y.: Evaluating local geometric feature representations for 3d rigid data matching. IEEE Trans. Image Process. **29**, 2522–2535 (2020)
48. Yang, J., Zhang, J., Cai, Z., Fang, D.: Novel 3d local feature descriptor of point clouds based on spatial voxel homogenization for feature matching. Visual Computing for Industry, Biomedicine, and Art **6**(1), 18 (2023)
49. Yew, Z.J., Lee, G.H.: 3dfeat-net: Weakly supervised local 3d features for point cloud registration. In: European Conference on Computer Vision (ECCV). pp. 607–623 (2018)
50. Yew, Z.J., Lee, G.H.: Rpm-net: Robust point matching using learned features. In: IEEE Computer Vision and Pattern Recognition (CVPR). pp. 11824–11833 (2020)
51. Yin, P., Yuan, S., Cao, H., Ji, X., Zhang, S., Xie, L.: Segregator: Global point cloud registration with semantic and geometric cues. In: IEEE International Conference on Robotics and Automation (ICRA). pp. 2848–2854 (2023)
52. Yu, F., Xiao, J., Funkhouser, T.: Semantic alignment of lidar data at city scale. In: IEEE Computer Vision and Pattern Recognition (CVPR). pp. 1722–1731 (2015)
53. Yuan, W., Eckart, B., Kim, K., Jampani, V., Fox, D., Kautz, J.: Deepgmr: Learning latent gaussian mixture models for registration. In: European Conference on Computer Vision (ECCV). pp. 733–750. Springer (2020)
54. Zaganidis, A., Sun, L., Duckett, T., Cielniak, G.: Integrating deep semantic segmentation into 3-d point cloud registration. IEEE Robotics and Automation Letters **3**(4), 2942–2949 (2018). https://doi.org/10.1109/LRA.2018.2848308
55. Zeng, A., Song, S., Nießner, M., Fisher, M., Xiao, J., Funkhouser, T.: 3dmatch: Learning local geometric descriptors from rgb-d reconstructions. In: IEEE Computer Vision and Pattern Recognition (CVPR). pp. 1802–1811 (2017)
56. Zhang, C., Song, Y., Yao, L., Cai, W.: Shape-oriented convolution neural network for point cloud analysis. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 34, pp. 12773–12780 (2020)
57. Zhang, C., Zhao, H., Wang, C., Tang, X., Yang, M.: Cross-modal monocular localization in prior lidar maps utilizing semantic consistency. In: IEEE International Conference on Robotics and Automation (ICRA). pp. 4004–4010 (2023)

58. Zhang, L., Guo, J., Cheng, Z., Xiao, J., Zhang, X.: Efficient pairwise 3-d registration of urban scenes via hybrid structural descriptors. IEEE Trans. Geosci. Remote Sens. **60**, 1–17 (2022)
59. Zhang, X., Yang, J., Zhang, S., Zhang, Y.: 3d registration with maximal cliques. In: IEEE Computer Vision and Pattern Recognition (CVPR). pp. 17745–17754 (2023)
60. Zhou, J., Wang, M., Mao, W., Gong, M., Liu, X.: Siamesepointnet: A siamese point network architecture for learning 3d shape descriptor. In: Comp. Graph. Forum. vol. 39, pp. 309–321. Wiley Online Library (2020)
61. Zhou, Q.Y., Park, J., Koltun, V.: Fast global registration. In: European Conference on Computer Vision (ECCV). pp. 766–782. Springer (2016)
62. Zhou, Q.Y., Park, J., Koltun, V.: Open3d: A modern library for 3d data processing. arXiv preprint arXiv:1801.09847 (2018)