https://doi.org/10.5755/j02.eie.35996

Sparse Point Cloud Registration Network with Semantic Supervision in Wilderness Scenes

Zhichao Zhang[1], Feng Lu[1], Youchun Xu[1], Jinsheng Chen[1], Yulin Ma[2,*]
[1]Institute of Military Transportation, Army Military Transportation University,
Tianjin 300161, China
[2]School of Mechanical Engineering, Anhui Polytechnic University,
Wuhu 241000, China
*zzcjake6688@163.com; 1849048346@qq.com; xu56419@126.com; jjuv_cjs@163.com; *mayulin@mail.ahpu.edu.cn*

***Abstract*—The registration of laser point clouds in complex conditions in wilderness scenes is an important aspect in the research field of autonomous vehicle navigation. It serves as the foundation for solving problems such as environment reconstruction, map construction, navigation and positioning, and pose estimation during the motion process of autonomous vehicles using laser radar sensors. Due to the sparse structured features, uneven point cloud density, and high noise levels in wilderness scenes, achieving reliable and accurate point cloud registration is challenging. In this paper, we propose a semantic-supervised sparse point cloud registration network (S3PCRNet) aiming to achieve effective registration of laser point clouds in wilderness large-scale scenes. Firstly, a local feature aggregation module is designed to extract the local structural features of the point cloud. Then, based on rotation position encoding, a randomly grouped self-attention mechanism is proposed to obtain the global features of the point cloud through learning. A semantic information weight matrix is calculated to filter out negligible points. Subsequently, a semantic fusion feature module is utilised to find reliable correspondences between point clouds. Finally, the proposed method is trained and evaluated on both the RELLIS-3D dataset and a self-made Off-road-3D dataset.**

***Index Terms*—Wilderness scenes; Laser point clouds; Point cloud registration; Semantic supervision.**

I. INTRODUCTION

With the gradual application of unmanned driving technology in various fields, unmanned vehicles are playing an increasingly important role in tasks such as mining operations, disaster relief, and battlefield transportation. In these harsh natural conditions and complex terrain outdoor scenes, unmanned vehicles need to have more stable environmental perception and autonomous navigation capabilities. However, based on image 2D perception and information processing, there are limitations that cannot fully satisfy the environmental perception requirements of unmanned vehicles in complex and varied environments. Compared with cameras, LiDAR has stronger environmental adaptation capabilities as a sensing sensor, and the processing and application of LiDAR 3D point cloud information in the field of unmanned vehicle technology is receiving increasing attention. Point cloud registration is a crucial link in the processing and application of 3D point cloud information, and it is also a basic work for tasks such as 3D reconstruction [1], perception positioning [2], and pose estimation for unmanned vehicles. Currently, research on point cloud registration for unmanned vehicles is mainly focused on indoor and urban environments. In a wide range of research-oriented point cloud data sets, numerous researchers have achieved outstanding research results and demonstrated advanced and robust performance in practical applications.

In-depth research on point cloud registration methods for unmanned vehicles is mainly focussed on laser point cloud simultaneous localization and mapping (SLAM), localisation and navigation, etc. and mainly includes four methods: iterative closest point (ICP) registration, normal distribution transformation (NDT) registration, feature-based registration, and deep learning-based registration. Represented by ICP [3], generalized ICP [4], point-to-line ICP [5], normal ICP [6], point cloud registration methods generally reduce registration error by iterative optimisation under known correspondence relationships. The NDT [7] algorithm is a statistical registration method based on the Gaussian distribution, which has high efficiency and accuracy in processing large-scale point cloud data. A series of SLAM algorithms, represented by lidar odometry and mapping (LOAM) [8]–[11], introduce line features and plane features in the odometry stage for interframe feature registration. Feature descriptor-based methods, such as point feature histograms (PFH) [12], fast point feature histograms (FPFH) [13], signature of histograms of orientations (SHOT) [14], rotational projection statistics (RoPS) [15], and fast global registration (FGR) [16], construct local structures of point clouds and achieve good results in registration tasks with significant pose differences. Deep learning-based methods have been a hot topic in recent years. Deep learning methods can generally learn strong feature representations of point clouds and have a stronger generalisation ability. By matching in high-dimensional feature space, they can find correspondence relationships and improve inlier rates, thereby achieving pose estimation. Since PointNet [17] and PointNet++ [18] proposed a deep learning network that can solve the problem of unordered representation of point clouds, classic point cloud registration

Manuscript received 9 August, 2023; accepted 10 January, 2024.
This research work is supported in part by the Natural Science Research Project of Anhui Educational Committee under Grant No. 2023AH020015; and in part by the Applied Basic Research Project of Wuhu under Grant No. 2023jc10.

networks that directly apply to raw point clouds, such as SpinNet [19], D3Feat [20], DeepVCP [21], HRegNet [22], Predator [23], PointDSC [24], and GeoTransformer [25], have also been continuously proposed. However, currently these methods [19] – [25] are mostly studied for structured scenes such as cities and campuses, and there is less research on point cloud registration in wilderness scenes such as mines, mountain roads, and forests.

In urban and campus scenes, artificial infrastructure such as paved roads, road shoulders, walls, buildings, and railings are the main elements of the scene, while there are also a large number of vehicles and pedestrians. However, in wilderness scenes such as forests and grasslands, there are fewer structured objects, and the number of vehicles and personnel is relatively low. Elements in the scene, such as grass, trees, rocks, and poles, are randomly distributed, with few structured features. Therefore, from the perspective of point cloud features, registration in these scenes is more challenging. In comparison to urban and campus scenes, natural conditions in wilderness scenes, such as fog, weeds, and dust, introduce more noise to the laser point cloud information, which in turn affects the extraction of point cloud features. Additionally, the denser the point cloud, the richer the expression of the scene features. To obtain more abundant features in originally sparse wilderness scenes, it is often desirable to increase the point cloud density. For example, high-density scanning LiDARs may be installed in unmanned vehicles or additional radars deployed to achieve high-density point cloud acquisition. However, this increases the computational complexity of point cloud data processing, and wilderness scenes with large-scale point clouds also face the challenge of managing the size of the point cloud data.

The process of feature extraction in point cloud registration networks is essentially a process of semantic aggregation. Whether it is through multilayer perceptron (MLP), convolution, or graph model network to hierarchically extract local or global features of point clouds, the aim is to discover features that have meaningful distinctions in a reasonable spatial dimension for each point and gather points of the same type into a cluster. Although networks such as DeepVCP, HRegNet, Predator, GeoTransformer, NgeNet [26], etc. have different methods, they all adopt multilayer network structures in the point cloud feature extraction stage and perform coarse matching when semantic features are clearly represented. The only difference is that they complete coarse matching without semantic label supervision. Due to the stability of features derived from line segment labelling, the authors in [27] believe that high-level semantic features are more suitable for point cloud registration of real LiDAR scans. They proposed self-supervised line labels to segment point clouds and extract features. SARNet is a point cloud registration network that explicitly proposes semantic enhancement. Compared to urban and highway environments, wilderness environments have unstructured boundaries, uneven terrain, strong textures, and irregular features.

To apply in wilderness scenes with less prominent structural features and more noise in harsh weather conditions, we propose a semantic-supervised point cloud registration network. Large-scale point clouds in large scenes, such as 128-line LiDAR single-frame point clouds reaching more than 230,000 points, have redundant information in the feature expression. Deep compressed point cloud registration (DCPCR) [28] uses deep learning methods to extract feature characteristics of the point cloud, generate compressed feature maps of the point cloud, and achieve a balance between reducing computational complexity and improving the accuracy of the point cloud registration. To adapt to the practical application of point cloud registration in wilderness scenes, we design a semantic-supervised sparse point cloud registration network (S3PCRNet) to enhance the registration accuracy and recall rate.

The main contributions of this paper are as follows.

1. Based on KPConv, a residual block is designed to extract strong structural features from local spatial features to address the problem of sparse structured features in wilderness scenes.
2. To utilise limited computational resources and increase the density of point clouds involved in self-attention calculations, a randomly grouped self-attention module is proposed. A rotation position encoding method is also introduced to enhance the model’s ability to aggregate contextual information from point clouds.
3. By introducing an explicitly expressed semantic-mixed feature matching module, unstable points in point cloud matching are removed, and stable features are strengthened. This allows one to obtain reliable correspondences from the matched point clouds.

II. MODELS AND METHODS

A. Backbone Network

Unlike point cloud classification and object detection tasks, for the registration network of low-overlap-rate outdoor large-scale point clouds, excessive attention to global features is not necessary during the feature descriptor extraction stage. This is because aggregating global features from nonoverlapping parts can be detrimental to subsequent feature matching tasks.

The backbone network is designed as shown in Fig. 1. Inspired by the local spatial encoding in RandLA-Net [29], we focus on designing a residual block for aggregation of local features based on local spatial encoding and KPconv kernel points in the point cloud feature extraction stage. This aims to better extract robust geometric structural features of point clouds in online and surface-feature-scarce outdoor scenes. In the global context aggregation step, the self-attention mechanism of the Transformer model is employed. However, since the self-attention mechanism requires high computational power and memory consumption, to reduce computational complexity, firstly we apply voxel filtering after obtaining the initial point cloud. In the local feature extraction stage, the point cloud is downsampled to less than 20,000 points and then the point cloud is randomly grouped. The same self-attention calculation is performed for each group. After obtaining the global contextual features, they are merged. Then, cross-attention calculation is introduced to establish prominent correspondences between the source point cloud and the target point cloud, followed by pose estimation.

First, the source point cloud P and the target point cloud Q undergo feature extraction through the local feature aggregation residual block and then they are downsampled to

P' and Q', respectively. During the downsampling process of multilayer feature extraction, the point cloud quantities of P' and Q' are controlled to be around 2000, according to practical considerations. The point cloud decoder is then applied to obtain the superpoint semantic confidence matrix. In the global context aggregation stage, irrelevant items in the scene are first removed based on semantic features. This process yields superpoints P and Q from the point cloud. Then, the superpoints are randomly grouped for self-attention operations, resulting in salient features of the superpoint point clouds.

In the feature matching stage, a semantic hybrid feature is constructed. Overlapping point cloud prediction is performed through cross-attention, and a correspondences weight matrix is computed. This enables the establishment of reliable point correspondences. Finally, singular value decomposition (SVD) is performed to obtain the spatial transformation of the point cloud.

B. Point Cloud Feature Encoder

Initial Feature Acquisition: Considering the influence of outdoor environments and weather conditions, we believe that the intensity and colour information of point clouds in real scenes are unstable, while extracting structural information from the three-dimensional spatial coordinates of point clouds is more robust. Taking the source point cloud P as an example, we first process it through a simple MLP module to obtain the high-level feature $F = \{f_i \in R^d | i = 1,2 \cdots, n\}$ of the point cloud. This initial MLP module consists of a linear layer, a batch normalisation layer (BatchNorm), and an activation layer (ReLU)

$$F = MLP\left(p_1, p_2, p_3, ..., p_n\right), \quad F \in R^{N\times d}. \tag{1}$$

The point cloud feature encoder consists of multiple layers of local spatial-convolutional feature residual blocks and downsampling layers. To improve the computational efficiency of large-scale point clouds, inspired by the work in [29], we adopt random sampling for the first two downsampling layers. In the last two layers, where the point cloud density is relatively sparse, weighted farthest point downsampling is employed. The introduction of weights ensures a better distribution of samples in different regions, and selecting samples with higher weights makes the sampling results more representative. Each downsampling layer performs pooling operations to maintain the number of output points.

Local Spatial Feature Aggregate Residual Blocks (LSFA_RB): Compared to multilayer perceptron (MLP) networks, KPConv demonstrates superior performance in extracting local structural features, making it highly favoured in the design of point cloud feature extraction networks.

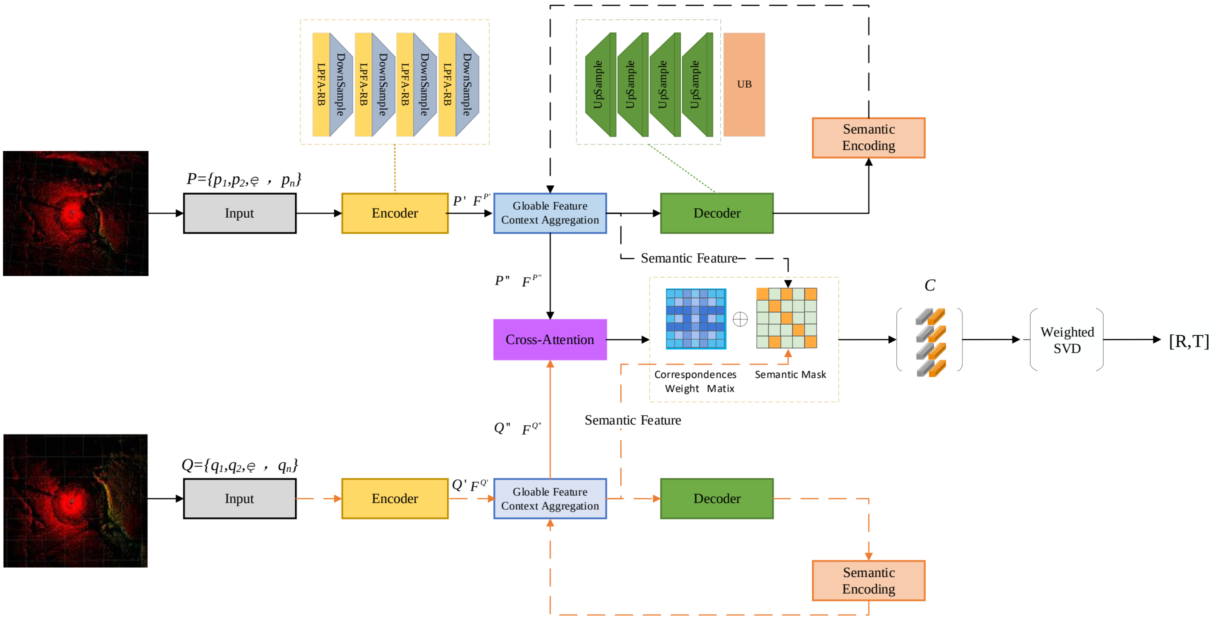

Fig. 1. Backbone network.

When handling unevenly dense point cloud data, the KPConv feature extraction based on spherical neighbourhood exhibits advantages over the K-nearest neighbour (KNN) based local feature extraction methods. Particularly for practical point cloud data in outdoor scenes, starting from the second layer of point cloud feature extraction, we design residual blocks based on KPConv to aggregate local features of the point cloud. Unlike the KPConv-FCNN network, we introduce the features extracted by KPConv into the local spatial feature encoding, aiming to enhance the expressive power of point cloud structural features. As shown in Fig. 2, we also design residual blocks for point cloud feature aggregation additionally, and local spatial feature encoding (LSFE) is shown in Fig. 3.

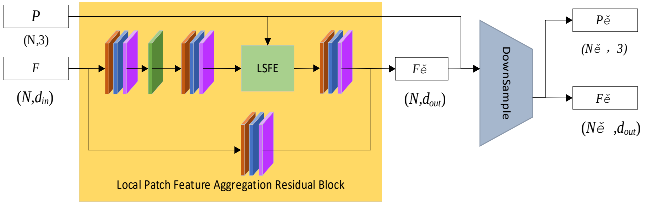

Fig. 2. Local spatial feature aggregate residual blocks.

Local Spatial Feature Encoding: To construct a local neighbourhood $\{p_i{}^k, f_i{}^k\}$ for point p_i using KNN search, p_i^k represents the points in the local neighbourhood of p_i, and f_i^k

represents their corresponding features. In this paper, spatial position encoding is applied to the p_i points. Unlike the work in [64], we consider p_i^k as redundant information. The local spatial feature encoding designed in this paper mainly consists of three parts: p_i, the relative position between p_i^k and p_i, the Euclidean distance between p_i^k and p_i, and the method is as follows

$$RPE = p_i \oplus \left(p_i - p_i^k\right) \oplus \left\| p_i - p_i^k \right\|. \tag{2}$$

Through the processing of the MLP module, the initial feature F is aligned with the feature space, resulting in the local spatial encoding feature r_i^k as follows

$$r_i^k = MLP\left(RPE\right). \tag{3}$$

At this point, the dimension of the output local spatial feature encoding feature r_i^k is the same as the dimension d of the initial feature F. Subsequently, a $2d$ feature is obtained by combining the local spatial feature with the initial feature, and then the neighbourhood feature of the point is derived through a shared MLP

$$f_i^{'} = MLP\left(\frac{1}{k}\sum_{j=1}^{k}\left(f_i^{\,j} \oplus r_i^{\,j}\right)\right). \tag{4}$$

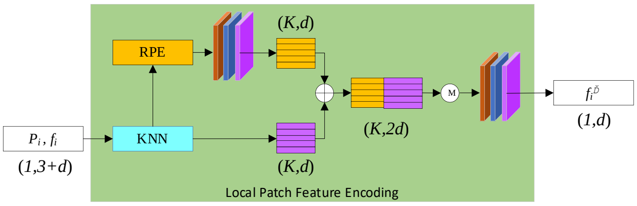

Fig. 3. Local spatial feature encoding (LSFE).

C. Point Cloud Feature Decoder

After completing the feature encoding of the point cloud, we adopt a self-attention mechanism based on rotational positional encoding in the global context information aggregation module. This mechanism effectively aggregates local and global information in point cloud data to obtain high-dimensional point cloud features with significance. To better extract semantic features from the point cloud, four upsampling layers are designed in the decoder section. These upsampling layers progressively increase the dimensionality of the point cloud data to better preserve the structural information of the original point cloud.

During the process of point cloud upsampling, it is more reasonable to maintain the invariance of the original point cloud structure. Similarly to the work in [29], we use the nearest neighbour method to reconstruct the upsampled points. This reconstruction method effectively preserves the structural information of the original point cloud and avoids data distortion caused by sampling. Meanwhile, a shared MLP module is utilised to reduce the dimensionality of the features. This module effectively reduces the dimensionality of high-dimensional point cloud features, facilitating subsequent semantic information extraction.

In the encoding phase, we preserve the indices of all down-sampled points in each encoding layer for subsequent correspondence of semantic information to the down-sampled layers of the encoder. This design effectively corresponds semantic information to the down-sampled layers of the encoder, thereby facilitating the subsequent decoding process. Therefore, for each point in the decoding layer, we can use the KNN algorithm to find the nearest neighbouring points from the points of the previous layer. This algorithm effectively matches the points in the decoding layer with their neighbouring points, thus better preserving the structural information of the original point cloud.

In the process of upsampling, each point's feature undergoes dimension reduction and is duplicated to its neighbouring points. This approach effectively spreads out the features of the points, ensuring that the semantic information of the original point cloud is preserved more accurately. Subsequently, a fully connected network is utilised to acquire the semantic features of each point. When the maximum value of these features is calculated, semantic labels can be determined. This design effectively extracts the semantic features of each point, thereby facilitating subsequent classification or recognition tasks.

D. Global Feature Context Aggregation Module

The global feature context aggregation module is primarily composed of two components: semantic-agnostic point cloud filtering and a random grouping self-attention mechanism. In outdoor scenes, semantic-agnostic point cloud filtering plays a crucial role in effectively removing unstable point cloud data, laying a solid foundation for subsequent processing. The random grouping self-attention mechanism aims to comprehensively aggregate the global context information of the point cloud while maintaining a relatively stable scale. This mechanism not only significantly improves the computational efficiency of the information aggregation process, but also greatly reduces memory consumption, providing strong support for large-scale point cloud data processing.

Point cloud filtering. Through point cloud decoding, we successfully extract the semantic features of the point cloud. These semantic features are obtained by aggregating the structural features of the point cloud, reflecting the unique properties of the environment. In urban environments, where the structural features are more prominent, the robustness of point cloud feature extraction is relatively high. However, in outdoor environments, the robustness of point cloud feature extraction is greatly reduced due to less distinct structural features.

In the representation of higher-level features, we can easily select points suitable for scene point cloud registration based on semantic features. For example, fences, poles, large rocks, curbs, trees, pits, mounds, etc. are stable points suitable for point cloud registration. However, moving objects such as vehicles, pedestrians, grass, smoke, or points with unstable structural features are not suitable for point cloud registration.

In the semantic representation stage, we adopt the following measures: first, unstable points are labelled with an ignore tag; second, a semantic mask matrix M^S is constructed. In this way, the points marked with the ignore tag will not participate in the subsequent computation of the network. This measure helps to improve the accuracy and stability of the computation. In this process, we use semantic features $S \in R^{N \times C}$ as input and set the number of points in the point cloud to N and the number of categories of semantic label to C.

Meanwhile, we assign different weights W^S based on the features of different categories. For prominent structural features, such as poles, large rocks, and trees, higher weights are assigned; while for other categories, general weights are assigned. This weight allocation method can better reflect the characteristics of different categories, thus improving the accuracy and effectiveness of point cloud registration

$$M^S = S \otimes W^S. \tag{5}$$

Design of self-attention module for randomisation groups. In outdoor environments, the presence of numerous irregular objects such as rocks, mounds of earth, and vegetation leads to relatively sparse structural features in point cloud data. In this context, the self-attention mechanism in the Transformer architecture becomes particularly important. It effectively captures the correlation information among sparse structural features in the point cloud, accurately extracting the salient parts and providing strong support for subsequent downstream tasks.

To better aggregate local features of the point cloud and ensure the participation of more points in the next stage of computation, we apply downsampling to the point cloud. However, it is worth noting that the self-attention module in the Transformer has a high computational complexity, consuming a large amount of memory during runtime. Therefore, measures were taken to strictly control the number of points after downsampling before feeding the point cloud into the Transformer architecture.

Taking the point cloud registration experiment on the Kitti dataset as an example, SARNet downsamples the point cloud to 1024 points. In the superpoint matching experiment with GeoTransform, the number of points in the point cloud remains within 1000 points. These measures aim to optimise computational efficiency and reduce memory consumption, ensuring stable operation of the algorithm.

In the field of point cloud processing, various attention mechanisms have been proposed, including sparse self-attention [30], linear self-attention [31], and local self-attention [32]. When applying global self-attention to large-scale point clouds, it results in a high memory consumption of O（N^2）, where N is the number of input points. Previous research [33] used a window-based self-attention mechanism to reduce computational complexity and memory consumption. This approach divides the three-dimensional space into nonoverlapping cubic windows, and each query point only considers the neighbouring points within the same window. Multiple heads of self-attention operate independently within each window, reducing computational complexity. However, this approach presents a limitation, as points within each window cannot establish associations with distant points, leading to information loss, particularly in large-scale point clouds. To address this issue, a hierarchical sampling strategy is proposed to expand the receptive field of self-attention and capture contextual dependencies for distant objects. In this paper, we introduce a random grouping self-attention module. Each attention head operates within its respective point cloud group, ensuring that the distribution of points within each group remains similar to that of the entire point cloud, albeit sparser. The unordered nature of point clouds ensures that random grouping does not alter the global features of the entire point cloud, and the computational complexity can be reduced to O（N^2/k), where k represents the number of groups.

As Fig. 4 clearly illustrates, the point cloud of a single frame from an outdoor scene was randomly divided into four separate groups, each assigned a unique colour. In Fig. 4(a), the original point cloud can be seen, while Figs. 4(b), 4(c), 4(d), and 4(e) represent the resulting subclouds following the random grouping. It is evident that, despite the local spatial sparsity within each subcloud, the overall distribution characteristics of the original point cloud are preserved. In this study, self-attention calculations are performed independently on each grouped subcloud, followed by feature aggregation using a multilayer perceptron (MLP). This approach offers a more effective means of emphasising salient local features within the global point cloud compared to grouping the points into nonoverlapping windows, as previously reported in [33] and [34].

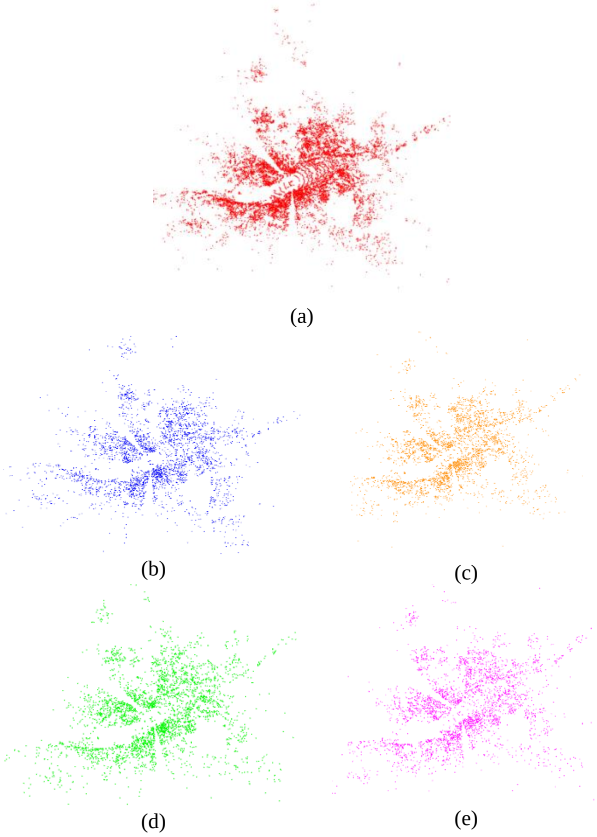

Fig. 4. Random grouping of point clouds.

Rotational position encoding. In the Transformer model for 3D object detection tasks, the authors in [35] argue that the use of three-dimensional coordinates as input features eliminates the need for position encoding in the Transformer network. Although the input to the Transformer already includes the three-dimensional coordinates of the point cloud in real-world scenes, the positional information may be lost in deeper layers of the network, where higher-level features are extracted. Explicit position encoding is an essential component of the Transformer network. Unlike in NLP, where word embeddings are used for position encoding, in point clouds, points and their features exist in a continuous three-dimensional space. Therefore, the transformation of three-dimensional coordinate information into sequential information presents a challenge.

The stratified transformer [33] utilises context-relative position encoding to adaptively capture element position information. It employs three learnable lookup tables to map relative coordinates to corresponding position encodings, treating the summation of three-dimensional coordinates as the sequential position of elements. This approach achieves position encoding for deep camera point cloud three-dimensional coordinates. The team further investigated the sparsity distribution of laser radar points in the spherical transformer [36]. They designed a radial window self-attention mechanism to capture long-range information and established a spherical coordinate system, particularly suitable for sparse distant points. By combining the exponentially segmented position encoding method developed in the spherical transformer, it achieved good results in point cloud segmentation and dynamic object feature capture in real-world scenes.

Inspired by this, our work proposes rotation position encoding based on X-Sin indexing, which avoids the computations required for transforming to a spherical coordinate system. In addition, the spherical transformer directly adds the logarithm of distance and two angles when calculating the position of feature elements, without considering weight parameters for the three variables. This may lead to the collapse of positional information, meaning that if the distances in one direction are the same, regardless of the differences in the distances and two angles, as long as the sum of the three is the same, this method will encode different relative positions into the same embedding. In real-world scenes, point cloud distributions occur in three-dimensional space, and for static point clouds that constitute the main body of the point cloud, they possess rotational and translational invariance. Therefore, we propose X-Sin indexing rotation position encoding, which can also better balance the distribution of laser point clouds with varying densities, ranging from sparse to dense. Compared to the indexing functions in [33] and [37], the indexing function in [36] can better aggregate sparsely distributed point elements at long distances. However, all of these methods involve relatively complex exponential operations, and specific formulas can be found in the references cited.

Given a point $p_i = (x_i, y_i, z_i)$, index encoding should be performed separately on x_i, y_i, z_i. Taking x_i as an example, the X-Sin index formula is proposed as follows

$$idx = \left\lfloor \frac{\beta}{\gamma}\left(x+\frac{\beta}{2}\right) - \frac{\beta}{2\pi} sin\left(\frac{2\pi}{\gamma}\left(x+\frac{\beta}{2}\right)\right) - \frac{\beta}{2} \right\rfloor, \tag{6}$$

where γ is the range of element x_i and β is the range expected in this paper.

Figure 5 shows the representations of the clip function [38], exponential function [39], sigmoid function, and the X-Sin indexing proposed in this paper. It can be observed that the X-Sin indexing enables the encoding of sparse points at distant locations to relatively closer positions, thereby partially balancing the uneven distribution of point cloud density. The setting of parameters γ and β allows for flexible indexing encoding in this work. For example, a smaller value range can be applied to the range of values for the vehicle-mounted laser point cloud z_i. Based on the actual distribution of point clouds, we perform indexing encoding separately for x_i, y_i, and z_i, which better preserves the three-dimensional positional information.

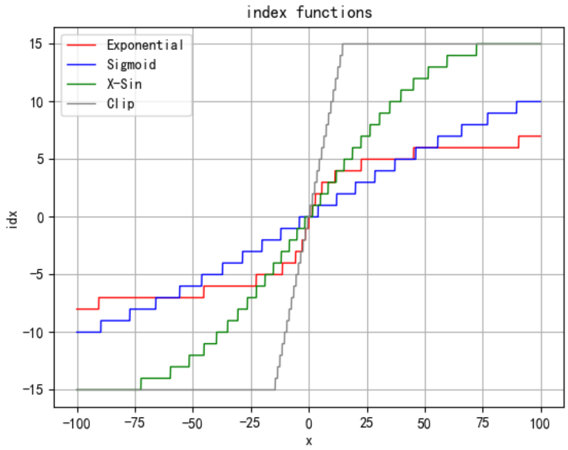

Fig. 5. Comparison of the different index functions.

Inspired by Roformer [40] and RDMNet [41], we adopt a rotation position encoding method with better extrapolation to enhance the applicability of point cloud position encoding, as is shown in Fig. 6. The fundamental concept of rotation position encoding involves introducing absolute position encoding through a rotation orthogonal matrix, allowing the model to pay close attention to relative positional relationships. RDMNet devised a novel rotation position encoding for point clouds to capture superior contextual and geometric information, and utilised MLP to derive rotation positions. Unlike RDMNet, we utilize the previously proposed X-Sin indexing encoding method to better address the uneven density distribution characteristics of vehicle-mounted point clouds.

When using two feature variables x_m and x_n as the query and key, respectively, with their corresponding indices m and n, the rotation position encoding function is denoted by $f(.)$. The position encodings for x_m and x_n are as follows:

$$q_m = W^Q x_m, \tag{7}$$

$$k_n = W^K x_n. \tag{8}$$

The rotation position coding matrices R_m and R_n are introduced for q_m and k_n, respectively, and the rotation position embedding formula is as follows:

$$f_k(k_n, n) = R_n k_n, \tag{9}$$

$$f_q(q_m, m) = R_m q_m. \tag{10}$$

Introduce the value v_n in the attention mechanism, and the attention enhancement feature z_m is calculated using the following formula

$$z_m = \frac{\sum_{N}^{n=1} (R_m q_m)^{\mathrm{T}} (R_n k_n) v_n}{\sum_{N}^{n=1} (W^Q x_m)^{\mathrm{T}} (W^K x_n)}. \tag{11}$$

Taking the query variable q_m as an example, where its dimension is d, and taking θ as the rotation angle, θ is calculated as follows

$$\theta_i = 10000^{-2(i-1)/d}, i \in [1, 2, ..., d/2]. \tag{12}$$

Then the d dimensional rotational position encoding matrix R_m of q_m is

$$R_m = \\ = \begin{bmatrix} cos\ m\ \theta_1 & -sin\ m\ \theta_1 & \cdots & 0 & 0 \\ sin\ m\ \theta_1 & cos\ m\ \theta_1 & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & cos\ m\ \theta_{\frac{d}{2}} & -sin\ m\ \theta_{\frac{d}{2}} \\ 0 & 0 & \cdots & sin\ m\ \theta_{\frac{d}{2}} & cos\ m\ \theta_{\frac{d}{2}} \end{bmatrix}. \tag{13}$$

To improve the efficiency of the calculation, taking the embedding formula of the rotation position $f_q(q_m, m)$ as an example, the detailed calculation is as follows

$$R_m \cdot q_m = \begin{bmatrix} q_1 \\ q_2 \\ \vdots \\ q_{d-1} \\ q_d \end{bmatrix} \otimes \begin{bmatrix} cos\ m\ \theta_1 \\ cos\ m\ \theta_1 \\ \vdots \\ cos\ m\ \theta_{d/2} \\ cos\ m\ \theta_{d/2} \end{bmatrix} + \begin{bmatrix} -q_2 \\ q_1 \\ \vdots \\ -q_d \\ q_{d-1} \end{bmatrix} \otimes \\ \otimes \begin{bmatrix} sin\ m\ \theta_1 \\ sin\ m\ \theta_1 \\ \vdots \\ sin\ m\ \theta_{d/2} \\ sin\ m\ \theta_{d/2} \end{bmatrix}. \tag{14}$$

In the previous section, we described how a single large-scale point cloud is randomly partitioned into four groups, with N representing the number of point clouds outputted from a selected group. To implement the self-attention module in this paper, we have designed the following diagram.

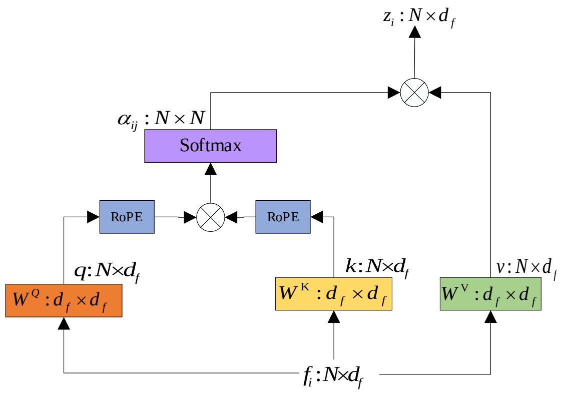

Fig. 6. The self-attention module for rotational po4sition encoding.

After grouping the point clouds, the feature vectors are denoted as F^l. The attention weights Z^l for the randomly divided point clouds are calculated using the rotation self-attention module, where l depends on the number of random groups. First, the feature attention residual block (12) is established, followed by merging and normalising the feature vectors from each group to obtain F'

$$F^l = F^l + Z^l, l = \{1, 2, 3, 4\}. \tag{15}$$

Taking the source point cloud P'' as an example, the rotation position encodes the self-attention features as follows

$$F^{P''} = MLP\left(F^1 \oplus F^2 \oplus F^3 \oplus F^4\right). \tag{16}$$

E. Characteristic-Based Cross-Over Attention

To improve the performance of point cloud registration and facilitate the exchange of feature information between the source and target point clouds, this study introduces a feature-based cross-attention mechanism. The main idea is to perform cross-attention calculations on point cloud features to capture more comprehensive interaction information between two frames of point clouds. To be specific, the rotation position encoded self-attention features obtained from the previous section are utilised as inputs, and the cross-attention mechanism is utilised to connect different subspaces of the two-point clouds. Fp" and FQ" denote the self-attention features of P" and Q", respectively. The cross-attention feature Zp" of P" is calculated as follows:

$$z_i^{P''} = \overset{j=1}{\overset{|Q''|}{\Sigma}} a_{i,j}\left(f_j^{Q''} w^V\right), \tag{17}$$

$$a_{ij} = \frac{exp\left(e_{ij}\right)}{\overset{k=1}{\overset{N}{\Sigma}} exp\left(e_{ik}\right)}, \tag{18}$$

$$e_{ij} = \frac{\left(f_i^{P''} W^Q\right)\left(f_j^{Q''} W^K\right)^T}{\sqrt{d_t}}, \tag{19}$$

where W^{Q}, W^{K}, and W^{V} are the weights of point cloud queries, keys, and values, which are learnable parameters, and the cross-attention feature Q'' of Z$^{Q''}$ is calculated in the same way as Z$^{P''}$.

Taking $F^{P''}$ as an example, we adopt multihead attention weight aggregation and residual operation. The calculation method is as follows

$$F^{p''}{}_{cross} = F^{p''} + \text{MLP}\left(\text{cat}\left(Z^Q{}_1'', Z^Q{}_2'', Z^Q{}_3'' Z^Q{}_4''\right)\right). \tag{20}$$

F. Selection of Correspondence Relationships

Semantic features $F_S^{P''}$ and $F_S^{Q''}$ are extracted from the decoder of the point cloud in this study, and similar to [15], they are represented by one-hot encoding based on semantic categories. However, unlike [42], this paper explicitly aims to incorporate semantic features into mixed high-order features. By aggregating the source point cloud P'' and target point cloud Q'' with cross-attention feature and semantic feature, $H^{P''}$ and $H^{Q''}$ are obtained:

$$H^{P''} = Cat\left(MLP\left(F_{cross}^{P''}\right), F_S^{P''}\right), \tag{21}$$

$$H^{Q''} = Cat\left(MLP\left(F_{cross}^{Q''}\right), F_S^{Q''}\right). \tag{22}$$

When processing point cloud data, if it is known that two points belong to different semantic categories, it can be determined that there is no correspondence between these two points. On the basis of this understanding, we can quickly exclude points with obviously different semantics by utilising semantic information, thereby avoiding unnecessary correspondence calculations. This strategy is particularly important when dealing with large-scale point cloud data because it can effectively reduce computational complexity, significantly reduce computation and time costs, and improve overall processing efficiency.

By decoding the point cloud, the semantic category confidence levels $s_i^{P"}$ and $s_j^{Q"}$ of the source point cloud P'' and the target point cloud Q'' are obtained, and semantic correlation matrices $S \in R^{|P"|\times|Q"|}$ are established accordingly. Through dual normalisation operations, ambiguous matching is suppressed [24]:

$$s_{i,j} = exp\left(-\left\|s_i^{p''} - s_j^{Q''}\right\|^2\right), \tag{23}$$

$$\hat{s}_{i,j} = \frac{s_{i,j}}{\sum_{k=1}^{|P''|} s_{i,k}} \times \frac{s_{i,j}}{\sum_{k=1}^{|Q''|} s_{k,j}}. \tag{24}$$

Inspired by the superpoint matching modules in [24] and [41], we utilise Gaussian correlation matrices $C \in R^{|P"|\times|Q"|}$ to measure the similarity between different subspaces of point clouds in the process of establishing correspondences between two frames. C is computed based on $H^{P''}$ and $H^{Q''}$:

$$c_{i,j} = exp\left(-\left\|h_i^{p''} - h_j^{Q''}\right\|^2\right), \tag{25}$$

$$\hat{c}_{i,j} = \frac{c_{i,j}}{\sum_{k=1}^{|P''|} c_{i,k}} \times \frac{c_{i,j}}{\sum_{k=1}^{|Q''|} c_{k,j}}. \tag{26}$$

By incorporating explicit semantic confidence and constructing a correlation matrix, concurrently with calculating Gaussian correlation matrices based on self-attention features, the correspondence matrix between the source point cloud P'' and the target point cloud Q'' is obtained by elementwise multiplication of the two matrices. The K elements with the highest values are selected as corresponding points between the two frames:

$$\tilde{C} = \left\{\tilde{c}_{ij} \middle| \tilde{c}_{ij} = \hat{c}_{ij} \times \hat{s}_{ij}, \hat{c}_{ij} \in C, \hat{s}_{ij} \in S\right\}, \tag{27}$$

$$\bar{C} = \left\{\left(\bar{p}_i, \bar{q}_j\right) \middle| (i, j) \in topk\left(\tilde{c}_{ij}\right)\right\}. \tag{28}$$

G. *Position Estimate*

By integrating global contextual features and employing correspondence point selection based on semantic fused features, we successfully achieved highly robust correspondences between two frames of point clouds. Furthermore, by utilising a random grouping self-attention mechanism, the computational complexity is significantly reduced, enabling a substantial increase in the scale of matching when dealing with sparse point cloud feature matching, thus eliminating the need for dense point cloud registration. During the pose estimation process, we directly perform SVD decomposition on sparse point cloud data to calculate the relative spatial transformation between the two frames of point clouds. The confidence of the correspondence relationship between the two frames of point clouds in C is used as the weight matrix W for the SVD decomposition, and the final pose transformation is computed using the following formula

$$(R,t) = \underset{R,t}{argmin} \sum_{i=1}^{n} w_i \left\|(Rp_i + t) - q_i\right\|^2. \tag{29}$$

The circle loss function is utilised to supervise the feature matching of point clouds in this study. For features $H^{P''}$ and $H^{Q''}$, the circle loss function is formulated as follows

$$L_{feature} = \frac{1}{2}\left(L_{P''}(H) + L_{Q''}(H)\right). \tag{30}$$

For the calculation of L, we consider a point q_i in Q''. Let M_i denote the set of positive samples of q_i in P'' and N_i denote the set of negative samples of q_i in P''. We set m and n as the upper limit for the number of positive and negative samples, respectively. The detailed calculation process is as follows:

$$L_{Q''}(H) = \frac{1}{L}\sum_{i=1}^{L} log\left[1 + \sum_{p''_j \in M_i} exp\left(\gamma\alpha_{ij}^{m}\right) \times \sum_{p''_k \in N_i} exp\left(\gamma\alpha_{ik}^{n}\right)\right], \tag{31}$$

$$d_{ij} = \left\|h_{q''_i} - h_{p''_j}\right\|_2, \tag{32}$$

$$\alpha_{ij}^{m} = \left\lfloor d_{ij} - m \right\rfloor_+, \tag{33}$$

$$\alpha_{ij}^{n} = \left\lfloor n - d_{ij} \right\rfloor_+. \tag{34}$$

The method of $L_{P"}(H)$ is calculated in the same way as above.

The computation method for pose estimation loss in this research involves evaluating the L_2 norm distance between the source point cloud transformed by the ground truth transformation and the predicted transformation. R_{est} and t_{est} correspond to the estimated rotation matrix and translation matrix of the model, respectively,

$$L_{transformation} = \frac{1}{M}\sum_{i=1}^{M}\left\|\mathrm{R}_{est}\ p''_i + test - q''_i\right\|_2. \tag{35}$$

A balanced cross-entropy loss function is utilised, which assigns different weights to samples from various classes. In this study, the inverse frequency of each class in the training set is used as its weight. By doing so, classes with a smaller sample count are given greater importance in the loss function, enabling better handling of semantic segmentation challenges in large-scale point clouds. The mathematical formula for the adopted balanced cross-entropy loss function is as follows

$$L_{semantic} = -\frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{C}\left\{w_j\ p_{ij}^{lable} log\left(s_{ij}^{P''}\right) + \left(1 - w_j\right)\left(1 - p_{ij}^{lable}\right) log\left(1 - s_{ij}^{P''}\right)\right\}. \tag{36}$$

For the semantic feature $S \in R^{N \times C}$, N is the number of points in the point cloud, and C is the number of semantic label categories.

We introduce three hyperparameters, β_f, β_t, and β_s and the learnable parameters of σ_f, σ_t, and σ_s to adjust the optimisation training process

$$L=\frac{\beta_f}{2\sigma_f^2}L_{feature}+\frac{\beta_t}{2\sigma_t^2}L_{transformation}+\frac{\beta_s}{2\sigma_s^2}L_{semantic}+ \\ +log\sigma_f+log\sigma_t+log\sigma_s. \qquad (37)$$

III. Experimental Evaluation

We designed experiments in different field scenarios, and analysed the registration accuracy and recall rate of S3PCRNet model, mainly including a self-made forest field (Off-road-3D) data set and RELLIS 3D field scene data set. A sample of self-made forest field (Off-road-3D) data set can be shown as in Fig. 7. Different colours of points indicate different categories.

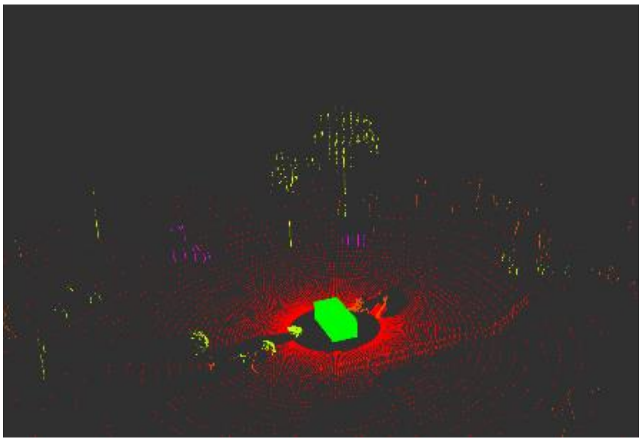

Fig. 7. Off-road-3D point cloud data.

The Off-road-3D dataset utilised in this study was generated by constructing a wilderness scene using UE4. A simulation of an suburban utility vehicle, equipped with a 128-line LiDAR sensor, was implemented through vehicle dynamics modeling. By operating the simulated vehicle in diverse off-road conditions, point cloud data of the wilderness scenes were collected and subsequently annotated with semantic labels, resulting in the creation of the Off-road-3D dataset. This dataset encompasses various key terrain categories, including forest trails, dirt roads, trees, grass, rocks, fences, vehicles, and pedestrians. These categories not only possess broad representativeness, but also have significant importance in the navigation and decision-making processes of autonomous vehicles in authentic off-road environments.

The RELLIS 3D dataset [43] is specifically designed for outdoor scenes, utilising annotated point cloud data acquired from a 64-line Ouster OS1 LiDAR and a 32-line Velodyne LiDAR. The data set encompasses key terrains such as rural roads, forests, and shrubbery, with categories including trees, grass, vehicles, pedestrians, artificial barriers, and debris piles. It serves as an essential resource platform for investigating autonomous navigation and scene understanding tasks in nonroad environments.

S3PCRNet, developed in this paper using PyTorch, utilised two NVIDIA Tesla V100 (16 GB memory) graphic processing units for computational purposes. The training process involved the utilisation of the Adam optimiser on the experimental dataset. Through examination and analysis of the experimental results, the effectiveness and robustness of the proposed algorithm were evaluated.

A. Data Enhancement Design

Data augmentation is a technique that can help models better adapt to various scenarios and fluctuations, while mitigating the risks of overfitting, thus improving a model's robustness and generalisability. Such techniques encompass an array of methods, including but not limited to random rotation, translation, scaling, and noise injection, all of which serve to increase the diversity of training samples. To ensure comparability with SARNet experiments, we adopt data augmentation methods similar to those used in SARNet, involving random rotation ($[0°, 45°]$) and translation ($[-5, 5]$) of point clouds within a given range, as well as adding random offsets to vehicles and pedestrians to represent unstable moving points. Furthermore, Gaussian noise was added to the point clouds during the training process.

B. Evaluation Indicators

The evaluation criteria used in this study are based on SARNet, consisting of the relative translation error (RTE), relative rotation error (RRE), and registration recall (RR). Successful registration is achieved when both RRE and RTE are within the predetermined threshold. Specifically, RRE is set to be less than 2 , and RTE is set to be less than 0.5 meters to meet the requirements of this research.

C. Comparative Experiment

HRgNet, GeoTransformer, and SARNet have all achieved impressive results in registering large-scale point clouds. HRgNet is a typical point cloud registration network that combines both deep and shallow features. The point cloud registration network structure proposed by GeoTransformer [25], which utilises geometric attention mechanisms, has been widely adopted. On the other hand, SARNet presents an innovative semantic-enhanced point cloud registration network. This paper considers these three-point cloud registration networks to be highly representative, and their effectiveness has been validated through experiments conducted on the SemanticKITTI dataset of urban scenes.

The application of models trained on SemanticKITTI for point cloud registration in wild environments resulted in unsatisfactory outcomes. Consequently, we retrained the aforementioned methods on the data set used in this paper and compared their performance against our proposed method. In the data preprocessing stage, we initially applied voxel filtering to the raw point clouds with a voxel size of 0.3. In wild environments, vehicles traverse not only asphalt and dirt roads but also relatively flat terrain. In such scenarios, drivable areas are prioritised due to the presence of obstacles like bumps in nonroad areas. LiDAR scans can detect these regions and present them in the point cloud. In addition, uneven and rough road surfaces exhibit unique features. Identifying ground features is crucial in sparsely populated outdoor environments. Unlike DeepVCP, we did not eliminate ground points from the point cloud.

The experiments were carried out on the Off-road-3D dataset and the RELLIS-3D dataset. The main focus was on evaluating three metrics: RTE, RRE, and RR.

Based on the experimental data presented in Table I and Table II and visualisation results show in Fig. 8, it can be observed that the registration performance of the algorithms compared in this study is generally better on the Off-road-3D dataset than on the RELLIS-3D dataset. This is mainly due to the fact that the Off-road-3D dataset contains a higher proportion of tree trunks, stones, and other features in the scene compared to the RELLIS-3D dataset, which has more grassland scenes. The richer structural features in the Off-road-3D dataset are more conducive to meeting the conditions for effective matching. Although SARNet relies heavily on confidence in semantic features, its semantic segmentation performance drops significantly in off-road environments, resulting in poorer point cloud registration accuracy on the off-road dataset. Therefore, further optimisation is required. On the other hand, HRgNet considers both bilateral consistency and neighbourhood consistency in point cloud registration, which has certain advantages in obtaining corresponding relationships. When tested on the outdoor data set proposed in this paper, HRgNet achieved good results. GeoTransformer has a unique design in geometric feature encoding, which results in relatively good performance. In terms of translation accuracy, it performs well. Similarly, the proposed S3PCRNet also uses a similar method for spatial encoding of geometric structures and exhibits good registration performance in outdoor environments.

TABLE I. COMPARISON OF DIFFERENT REGISTRATION METHODS (OFF-ROAD-3D DATA SET).

Method	**Off-road-3D**		
	RRE (deg)	**RTE (m)**	**Recall (%)**
SARNet	0.27	0.22	81.2
HRgNet	0.24	0.14	82.1
GeoTransformer	0.22	**0.09**	84.7
S3PCRNet	**0.21**	0.12	**85.6**

TABLE II. COMPARISON OF DIFFERENT REGISTRATION METHODS (RELLIS-3D DATA SET).

Method	**RELLIS-3D**		
	RRE (deg)	**RTE (m)**	**Recall (%)**
SARNet	0.31	0.27	71.3
HRgNet	0.26	0.23	76.5
GeoTransformer	0.25	**0.15**	78.6
S3PCRNet	**0.25**	0.21	**79.1**

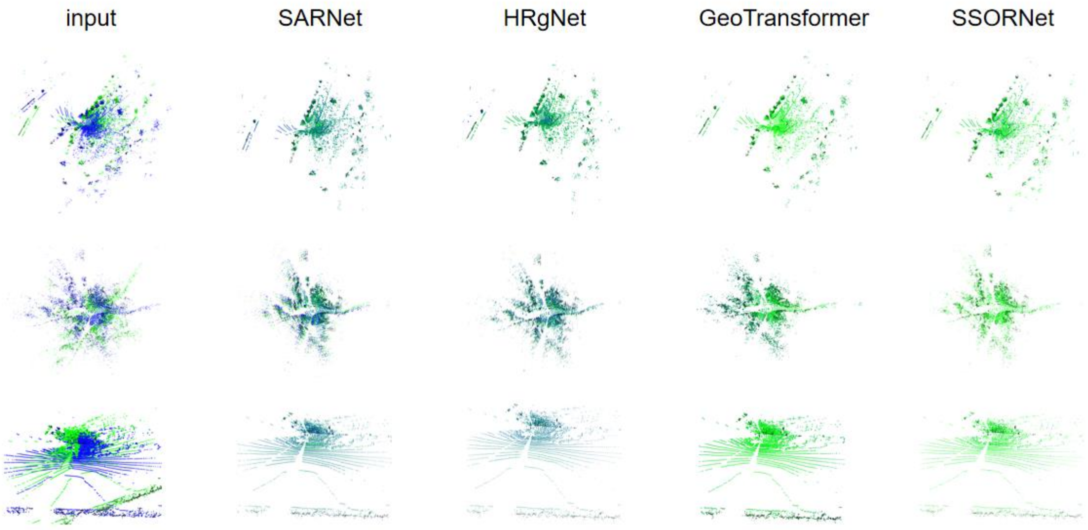

Fig. 8. Comparative experiment results.

D. Ablation Experiment

In this study, extensive ablation studies were conducted to gain a better understanding of the various modules of our method.

Local feature aggregation module: To validate the effectiveness of the local spatial feature aggregate residual blocks (LSFA-RB) module, it was replaced with the KPEncoder [44] module in GeoTransformer. The LPFA-RB module has a spatial position encoding function, which provides a greater advantage in extracting local structural features compared to the KPEncoder module, which is shown in Table III.

TABLE III. ABLATION STUDIES ON RELLIS-3D DATA SET.

Model	**RRE (deg)**	**RTE (m)**	**Recall (%)**
KPEncoder	0.37	0.23	77.9
LSFA-RB	**0.25**	**0.21**	**79.1**

The random grouping self-attention mechanism, a significant concept presented in this study, was contrasted with the conventional multihead self-attention mechanism (MSA). When the MSA was applied under the same computational complexity, the maximum number of points for point cloud matching was limited to 1024. However, with the random grouping self-attention mechanism, the number of points in point clouds increased to 4096 at a similar level of memory consumption, allowing the self-attention calculation in denser point clouds and enhancing the expression of features. In this study, comparative experiments were conducted by downsampling point clouds to 1024, 2048, and 4096 points, respectively. The comparative experimental results are shown in Table IV.

TABLE IV. ABLATION STUDIES ON RELLIS-3D DATA SET.

Model	**RRE (deg)**	**RTE (m)**	**Recall (%)**
MSA（1024）	0.35	0.33	75.7
MSA（2048）	0.29	0.28	76.3
RGSA（1024）	0.36	0.35	72.7

Model	RRE (deg)	RTE (m)	Recall (%)
SGSA（2048）	0.27	0.25	77.4
SGSA（4096）	**0.25**	**0.21**	**79.1**

Rotation position encoding based on the X-Sin index: The X-Sin index is a key method for balancing the uneven density of laser point clouds and enhancing model robustness. It greatly reduces the computational cost and parameter quantity of large-scale point cloud sequences. Compared to the exponential piecewise indexing function (PIF) mentioned in [37], the X-Sin index offers more flexibility in adjustment and provides finer granularity for sparser point clouds in distant areas. The comparative experimental results are shown in Table V.

TABLE V. ABLATION STUDIES ON RELLIS-3D DATA SET.

Model	RRE (deg)	RTE (m)	Recall (%)
PIF	0.25	0.24	78.3
X-Sin	**0.25**	**0.21**	**79.1**

Semantic supervision: In the stage of semantic fusion feature matching, this study begins by aggregating semantic features and introduces the confidence of semantic features when calculating the correspondence matrix. To validate the effectiveness of semantic supervision, we performed comparative experiments with ablation. The experimental results, which are shown in Table VI, indicate that aggregating semantic features and incorporating semantic feature confidence can significantly improve the performance of the model in point cloud registration tasks.

TABLE VI. ABLATION STUDIES ON RELLIS-3D DATA SET.

Model	RRE (deg)	RTE (m)	Recall (%)
Without SS	0.29	0.19	70.7
With SS	**0.25**	**0.21**	**79.1**

E. *Limitations*

To mitigate the impact of diverse weather conditions in real-life settings on point cloud data, this paper disregards the intensity information of the laser point cloud and instead extracts semantic information from the structural features of the point cloud. Therefore, the accuracy of semantic information determines the effectiveness of semantic supervision. Accurately segmenting point clouds in outdoor environments poses a significant challenge and demands a more diverse point cloud segmentation data set. To address this issue, this study developed the Off-road-3D data set and incorporated semantic confidence information into the semantic mask matrix, assigning higher weights to reliably segmented tree trunks, vehicles, personnel, and rocks while categorising unreliable segments such as road surfaces, grass, and low-lying objects into other classes.

IV. CONCLUSIONS

In this paper, we propose a sparse point cloud registration network tailored for large outdoor scenes. The key concept of this network is to extract semantic features from the point cloud through enhanced structural information extraction and to explicitly guide correspondence selection in point cloud registration. To achieve this objective, we innovatively designed the local spatial feature aggregation module, the random grouping self-attention mechanism module, and the semantic fusion feature matching module. Training and optimisation were performed on the RELLIS-3D data set and a custom Off-road-3D data set, and the effectiveness of the proposed method was validated through ablation experiments. However, it is important to note that point cloud registration in large outdoor scenes presents greater challenges compared to urban environments. Classical learning-based point cloud registration methods exhibit significant decreases in registration performance and fail to achieve the registration recall rate and precision observed in urban environment data sets such as KITTI and Nuscenes. Therefore, in future work, we will continue to enrich outdoor environment data sets and optimise network designs to enhance point cloud registration performance and meet the requirements of more downstream tasks.

CONFLICTS OF INTEREST

The authors declare that they have no conflicts of interest.

REFERENCES

[1] G. Wang *et al.*, “What matters for 3D scene flow network”, in *Computer Vision - ECCV 2022. ECCV 2022. Lecture Notes in Computer Science*, vol. 13693. Springer, Cham, 2022, pp. 38–55. DOI: 10.1007/978-3-031-19827-4_3.

[2] Y. Zheng, Y. Li, S. Yang, and H. Lu, “Global-PBNet: A novel point cloud registration for autonomous driving”, *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 11, pp. 22312–22319, 2022. DOI: 10.1109/TITS.2022.3153133.

[3] S. Rusinkiewicz and M. Levoy, “Efficient variants of the ICP algorithm”, in *Proc. of Third International Conference on 3-D Digital Imaging and Modeling*, 2021, pp. 145–152. DOI: 10.1109/IM.2001.924423.

[4] R. Kuramachi, A. Ohsato, Y. Sasaki, and H. Mizoguchi, “G-ICP SLAM: An odometry-free 3D mapping system with robust 6DoF pose estimation”, in *Proc. of 2015 IEEE International Conference on Robotics and Biomimetics (ROBIO)*, 2015, pp. 176–181. DOI: 10.1109/ROBIO.2015.7418763.

[5] Y. Cao, Z. Zhang, X. Chen, H. Zhu, and P. Zhao, “Indoor SLAM algorithm based on PL-ICP and map matching”, in *Proc. of 2021 1st International Conference on Control and Intelligent Robotics*, 2021, pp. 295–299. DOI: 10.1145/3473714.3473765.

[6] J. Serafin and G. Grisetti, “NICP: Dense normal based point cloud registration”, in *Proc. of 2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2015, pp. 742–749. DOI: 10.1109/IROS.2015.7353455.

[7] S. Chen, H. Ma, C. Jiang, B. Zhou, W. Xue, Z. Xiao, and Q. Li, “NDT-LOAM: A real-time lidar odometry and mapping with weighted NDT and LFA”, *IEEE Sensors Journal*, vol. 22, no. 4, pp. 3660–3671, 2022. DOI: 10.1109/JSEN.2021.3135055.

[8] J. Zhang and S. Singh, “LOAM: Lidar odometry and mapping in real-time”, in *Proc. of Robotics: Science and Systems (RSS’14)*, 2014.

[9] H. Wang, C. Wang, C.-L. Chen, and L. Xie, “F-LOAM: Fast LiDAR odometry and mapping”, in *Proc. of 2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2021, pp. 4390–4396. DOI: 10.1109/IROS51168.2021.9636655.

[10] T. Shan and B. Englot, “LeGO-LOAM: Lightweight and ground optimized lidar odometry and mapping on variable terrain”, in *Proc. of 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2018, pp. 4758–4765. DOI: 10.1109/IROS.2018.8594299.

[11] L. Liao, C. Fu, B. Feng, and T. Su, “Optimized SC-F-LOAM: Optimized fast LiDAR odometry and mapping using scan context”, 2022. arXiv: 2204.04932. DOI: 10.1109/CVCI56766.2022.9964574.

[12] R. B. Rusu, N. Blodow, Z. C. Marton, and M. Beetz, “Aligning point cloud views using persistent feature histograms”, in *Proc. of 2008 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2008, pp. 3384–3391. DOI: 10.1109/IROS.2008.4650967.

[13] R. B. Rusu, N. Blodow, and M. Beetz, “Fast Point Feature Histograms (FPFH) for 3D registration”, in *Proc. of 2009 IEEE International Conference on Robotics & Automation*, 2009, pp. 3212–3217. DOI: 10.1109/ROBOT.2009.5152473.

[14] F. Tombari, S. Salti, and L. Di Stefano, “Unique signatures of histograms for local surface description”, in *Computer Vision - ECCV 2010. ECCV 2010. Lecture Notes in Computer Science*, vol. 6313.

Springer, Berlin, Heidelberg, 2010, pp. 356–369. DOI: 10.1007/978-3-642-15558-1_26.

[15] Y. Guo, F. Sohel, M. Bennamoun, M. Lu, and J. Wan, "Rotational projection statistics for 3D local surface description and object recognition", *International Journal of Computer Vision*, vol. 105, pp. 63–86, 2013. DOI: 10.1007/s11263-013-0627-y.

[16] Q.-Y. Zhou, J. Park, and V. Koltun, "Fast global registration", *Computer Vision - ECCV 2016. Lecture Notes in Computer Science()*, vol. 9906. Springer, Cham, 2016, pp. 766–782. DOI: 10.1007/978-3-319-46475-6_47.

[17] C. R. Qi, H. Su, K. Mo, and L. J. Guibas, "Pointnet: Deep learning on point sets for 3D classification and segmentation", in *Proc. of 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 77–85. DOI: 10.1109/CVPR.2017.16.

[18] C. R. Qi, L. Yi, H. Su, and L. J. Guibas, "Pointnet++: Deep hierarchical feature learning on point sets in a metric space", in *Proc. of 31st Conference on Neural Information Processing Systems (NIPS2017)*, 2017, pp. 1–10.

[19] S. Ao, Q. Hu, B. Yang, A. Markham, and Y. Guo, "Spinnet: Learning a general surface descriptor for 3D point cloud registration", in *Proc. of 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021, pp. 11748–11757. DOI: 10.1109/CVPR46437.2021.01158.

[20] X. Bai, Z. Luo, L. Zhou, H. Fu, L. Quan, and C.-L. Tai, "D3Feat: Joint learning of dense detection and description of 3D local features", in *Proc. of 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020, pp. 6358–6366.

[21] W. Lu, G. Wan, Y. Zhou, X. Fu, P. Yuan, and S. Song, "DeepVCP: An end-to-end deep neural network for point cloud registration", in *Proc. of 2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, 2019, pp. 12–21. DOI: 10.1109/ICCV.2019.00010.

[22] F. Lu *et al.*, "HRegNet: A hierarchical network for large-scale outdoor lidar point cloud registration", in *Proc. of 2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021, pp. 15994–16003. DOI: 10.1109/ICCV48922.2021.01571.

[23] S. Huang, Z. Gojcic, M. Usvyatsov, A. Wieser, and K. Schindler, "PREDATOR: Registration of 3D point clouds with low overlap", in *Proc. of 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021, pp. 4265–4274. DOI: 10.1109/CVPR46437.2021.00425.

[24] X. Bai *et al.*, "PointDSC: Robust point cloud registration using deep spatial consistency", in *Proc. of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021, pp. 15854–15864. DOI: 10.1109/CVPR46437.2021.01560.

[25] Z. Qin, H. Yu, C. Wang, Y. Guo, Y. Peng, and K. Xu, "Geometric transformer for fast and robust point cloud registration", in *Proc. of 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022, pp. 11133–11142. DOI: 10.1109/CVPR52688.2022.01086.

[26] L. Zhu, H. Guan, C. Lin, and R. Han, "Neighborhood-aware geometric encoding network for point cloud registration", 2022. arXiv: 2201.12094.

[27] X. Zhao *et al.*, "SuperLine3D: Self-supervised line segmentation and description for LiDAR point cloud", in *Computer Vision - ECCV 2022. ECCV 2022. Lecture Notes in Computer Science*, vol. 13669. Springer, Cham, 2022, pp. 263–279. DOI: 10.1007/978-3-031-20077-9_16.

[28] L. Wiesmann, T. Guadagnino, I. Vizzo, G. Grisetti, J. Behley, and C. Stachniss, "DCPCR: Deep compressed point cloud registration in large-scale outdoor environments", *IEEE Robotics and Automation Letters*, vol. 7, no. 3, pp. 6327–6334, 2022. DOI: 10.1109/LRA.2022.3171068.

[29] Q. Hu *et al.*, "RandLA-Net: Efficient semantic segmentation of large-scale point clouds", in *Proc. of 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020, pp. 11105–11114. DOI: 10.1109/CVPR42600.2020.01112.

[30] R. Child, S. Gray, A. Radford, and I. Sutskever, "Generating long sequences with sparse transformers", 2019. arXiv: 1904.10509.

[31] F. Babiloni *et al.*, "Linear complexity self-attention with 3rd order polynomials", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 45, no. 11, pp. 12726–12737, 2023. DOI: 10.1109/tpami.2022.3231971.

[32] X. Shen, D. Han, Z. Guo, C. Chen, J. Hua, and G. Luo, "Local self-attention in transformer for visual question answering", *Appl. Intell.*, vol. 53, no. 13, pp. 16706–16723, 2023. DOI: 10.1007/s10489-022-04355-w.

[33] X. Lai *et al.*, "Stratified transformer for 3D point cloud segmentation", in *Proc. of 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022, pp. 8490–8499. DOI: 10.1109/CVPR52688.2022.00831.

[34] X. Ma *et al.*, "Luna: Linear unified nested attention", 2021. arXiv: 2106.01540v2.

[35] I. Misra, R. Girdhar, and A. Joulin, "An end-to-end transformer model for 3D object detection", in *Proc. of the 2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021, pp. 2886–2897. DOI: 10.1109/ICCV48922.2021.00290.

[36] X. Lai, Y. Chen, F. Lu, J. Lu, and J. Jia, "Spherical transformer for LiDAR-based 3D recognition", in *Proc. of 2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023, pp. 17545–17555. DOI: 10.1109/CVPR52729.2023.01683.

[37] K. Wu, H. Peng, M. Chen, J. Fu, and H. Chao, "Rethinking and improving relative position encoding for vision transformer", in *Proc. of 2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021, pp. 10013–10021. DOI: 10.1109/ICCV48922.2021.00988.

[38] P. Shaw, J. Uszkoreit, and A. Vaswani, "Self-attention with relative position representations", in *Proc. of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 2018, pp. 464–468, vol. 2. DOI: 10.18653/v1/N18-2074.

[39] G. Ke, D. He, and T.-Y. Liu, "Rethinking positional encoding in language pre-training", 2021. arXiv: 2006.15595.

[40] J. Su, Y. Lu, S. Pan, B. Wen, and Y. Liu. RoFormer: Enhanced transformer with rotary position embedding", 2021. arXiv: 2104.09864.

[41] C. Shi, X. Chen, H. Lu, W. Deng, J. Xiao, and B. Dai, "RDMNet: Reliable dense matching based point cloud registration for autonomous driving", 2023. arXiv: 2303.18084. DOI: 10.1109/TITS.2023.3286464.

[42] C. Liu, J. Guo, D.-M. Yan, Z. Liang, X. Zhang, and Z. Cheng, "SARNet: Semantic augmented registration of large-scale urban point clouds", 2022. arXiv: 2206.13117.

[43] S. Wang, B. Z. Li, M. Khabsa, H. Fang, and H. Ma, "Linformer: Self-attention with linear complexity", 2020. arXiv: 2006.04768.

[44] H. Thomas, C. R. Qi, J.-E. Deschaud, B. Marcotegui, F. Goulette, and L. Guibas, "KPConv: Flexible and deformable convolution for point clouds", in *Proc. of 2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, 2019, pp. 6410–6419. DOI: 10.1109/ICCV.2019.00651.