remote sensing

MDPI

Article

A Registration Method of Overlap Aware Point Clouds Based on Transformer-to-Transformer Regression

Yafei Zhao [1], Lineng Chen [2], Quanchen Zhou [1], Jiabao Zuo [1], Huan Wang [1] and Mingwu Ren [1,*]

[1] School of Computer Science and Engineering, Nanjing University of Science and Technology, Nanjing 210094, China; 218106010160@njust.edu.cn (Y.Z.); quanchen.zhou@njust.edu.cn (Q.Z.); jiabao.zuo@njust.edu.cn (J.Z.); wanghuanphd@njust.edu.cn (H.W.)

[2] School of Electronic and Information Engineering, Guangxi Normal University, Guilin 541004, China; linengchen@gxnu.edu.cn

* Correspondence: renmingwu@njust.edu.cn

Abstract: Transformer has recently become widely adopted in point cloud registration. Nevertheless, Transformer is unsuitable for handling dense point clouds due to resource constraints and the sheer volume of data. We propose a method for directly regressing the rigid relative transformation of dense point cloud pairs. Specifically, we divide the dense point clouds into blocks according to the down-sampled superpoints. During training, we randomly select point cloud blocks with varying overlap ratios, and during testing, we introduce the overlap-aware Rotation-Invariant Geometric Transformer Cross-Encoder (RIG-Transformer), which predicts superpoints situated within the common area of the point cloud pairs. The dense points corresponding to the superpoints are inputted into the Transformer Cross-Encoder to estimate their correspondences. Through the fusion of our RIG-Transformer and Transformer Cross-Encoder, we propose Transformer-to-Transformer Regression (TTReg), which leverages dense point clouds from overlapping regions for both training and testing phases, calculating the relative transformation of the dense points by using the predicted correspondences without random sample consensus (RANSAC). We have evaluated our method on challenging benchmark datasets, including 3DMatch, 3DLoMatch, ModelNet, and ModelLoNet, demonstrating up to a 7.2% improvement in registration recall. The improvements are attributed to our RIG-Transformer module and regression mechanism, which makes the features of superpoints more discriminative.

Keywords: point cloud registration; Transformer-to-Transformer; dense point cloud

check for updates

Citation: Zhao, Y.; Chen, L.; Zhou, Q.; Zuo, J.; Wang, H.; Ren, M. A Registration Method of Overlap Aware Point Clouds Based on Transformer-to-Transformer Regression. *Remote Sens.* **2024**, *16*, 1898. https://doi.org/10.3390/rs16111898

Academic Editor: Jorge Delgado García

Received: 7 March 2024
Revised: 22 May 2024
Accepted: 23 May 2024
Published: 25 May 2024

1. Introduction

Point cloud registration is a critical research area within the realms of computer vision and robotics, serving as pivotal function in diverse applications including 3D object reconstruction, scene comprehension, and robotic manipulation [1,2]. Achieving precise alignment of point clouds enables the amalgamation of data from varied sources, thereby supporting activities such as environmental modeling, object identification, and augmented reality applications. Enhancing the efficiency and precision of point cloud registration algorithms empowers researchers to elevate the performance of autonomous systems, robotic perception, and augmented reality applications, consequently driving progress across sectors spanning industrial automation to immersive virtual reality encounters.

Recently, there has been a notable increase in research within the domain of point cloud registration focusing on deep learning methodologies. These innovative strategies utilize neural networks to directly acquire descriptions from 3D points, eliminating the necessity for manual feature engineering and tackling issues like varying point density and noise. Fully Convolutional Geometric Features (FCGF) [3] is a deep learning method that seeks to extract geometric features directly from point clouds. Through the application of fully convolutional neural networks, FCGF can effectively capture both local and global geometric details, facilitating precise point cloud registration amidst noise and partial overlap. FCGCF [4] incorporates color data from point clouds into the FCGF network structure, merging geometric structural details with color features for enhanced representation. By fusing geometric and color information, the feature descriptors are enhanced in distinguishing points with high similarity in three-dimensional geometric structures. Udpreg [5] proposes a distribution consistency loss function based on a mixture of Gaussian models to supervise the network in learning its posterior distribution probabilities. It combines this approach with the Sinkhorn algorithm [6] to handle partial point cloud registration, aiding the network in extracting discriminative local features. Through unsupervised learning, UDPReg achieves label-free point cloud registration. GeoTransformer [7] introduces a method to extract global geometric features from the position coordinates of superpoints. It presents a geometric Transformer for learning global features and introduces the overlap circle loss function, treating superpoint feature learning as metric learning. By combining this approach with the Sinkhorn method, GeoTransformer achieves point cloud registration without the need for RANSAC [8]. RoITr [9] introduces a network based on the Transformer architecture utilizing channel-shared weights to leverage the global properties of Transformer. Building upon the GeoTransformer framework, it embeds geometric features from self-attention modules into cross-attention modules to achieve rotation invariance in the Transformer structure. RegTR [10] utilizes a superpoint correspondence projection function to directly constrain the features interacting with the Transformer Cross-Encoder and the voxelized superpoint coordinates. This method replaces RANSAC and directly regresses the relative transformation matrix. RORNet [11] divides point clouds into several small blocks and learns the latent features of overlapping regions within these blocks. This approach reduces the feature uncertainty caused by global contrast and subsequently selects highly confident keypoints from the overlapping regions for point cloud registration. HR-Net [12] introduces a dense point matching module to refine the matching relationships of dense points and utilizes a recursive strategy to globally match superpoints of point clouds and locally adjust dense point clouds layer by layer, thereby estimating a more accurate transformation matrix. Roreg [13] addresses the point cloud registration challenge by focusing on directional descriptors and local rotation techniques. The directional descriptors are categorized into rotational equivariance and rotational invariance components. Equivariance mandates that descriptors are invariant to transformations in the relative point positions within the point cloud, whereas invariance ensures that registration outcomes are insensitive to changes in scale, rotations, or translations of the point cloud. A local rotation approach is devised to integrate rough rotations for significant angle adjustments with precise rotations for minor angle variations, aiming to ascertain the optimal rotation amount and improve registration precision.

Combining the 3D coordinates and features of superpoints, RegTR [10] employs Transformer to directly perform global information interaction on superpoints. However, the coordinates of superpoints are sparse, and the computation on superpoints is voxelized around the centers of point cloud blocks, introducing errors in superpoint coordinates, especially for point clouds with small areas of overlap. We seek to leverage the global properties of Transformer to extract and incorporate global information from dense point clouds. Nevertheless, due to the limitations of Transformer in terms of data length and computational resources, direct processing of dense point clouds is not feasible. Through multiple experiments and data analysis, we discovered that the similarity between the neighborhoods of points outside the overlapping region and those inside the overlapping region has a significant influence on point cloud registration. Points within the overlapping region have less significance for point cloud registration due to their uniform structure. Therefore, it is crucial to select the overlapping region and features with higher discriminative power within this region to enhance the registration's effectiveness. Drawing inspiration from previous studies [7,9,10], we segmented the point cloud registration procedure into two distinct stages. Initially, we leverage Transformer's overarching characteristics to differentiate the overlapping and non-overlapping zones, thereby converting the point-to-point matching challenge into a classification task across these areas. Subsequently, we select representative dense keypoints within the overlapping region using a Transformer Cross-Encoder to directly regress the relative transformation.

2. Materials and Methods

2.1. Problem Setting

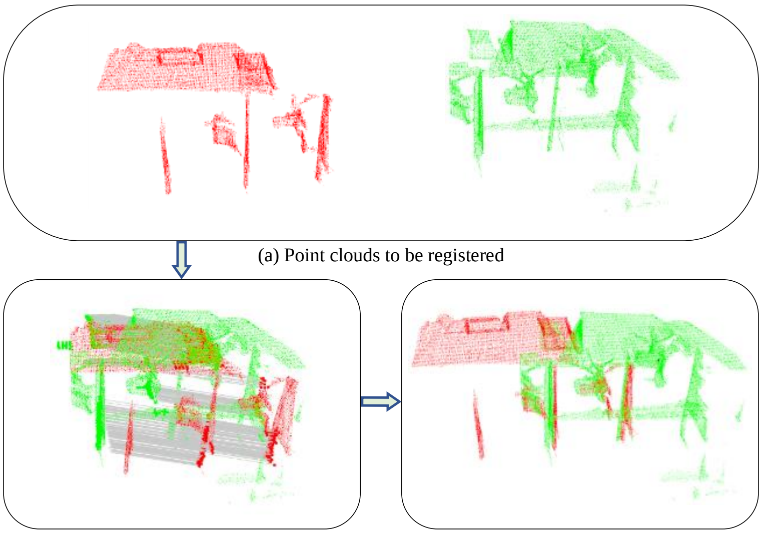
Our objective is to utilize dense point clouds to compute the relative rigid transformation matrix $\mathcal{T}^* \in SE(3)$ between point cloud pairs $\mathbf{P}_0 \in \mathbb{R}^{3\times M}$ and $\mathbf{Q}_0 \in \mathbb{R}^{3\times N}$ by minimizing the Equation (1) defined as follows:

$$\mathcal{F} = \min_{\mathcal{T}^*} \sum_{(\hat{p}_j,\hat{q}_k)\in\hat{C}^{p^d}} \left\|\mathcal{T}^*(\hat{p}_j) - \hat{q}_k\right\|_2, \tag{1}$$

where $\hat{C}^{p^d} = \{(\hat{p}_j,\hat{q}_k) | \hat{p}_j \in p^d \subset Q, \hat{q}_k \in q^d \subset Q\}$, which is the set of predicted dense correspondences; $(\hat{p}_j,\hat{q}_k)$ is a pair of correspondence; and $\|\cdot\|_2$ is $\mathcal{L}_2$ norm.

2.2. Overview of Our Method

Our approach, named TTReg, utilizes a global transformer to select dense correspondences related to sparse superpoints within the common area to estimate the transformation (See Figure 1). TTReg consists of an encoder–decoder feature extraction module, a sparse superpoint matching module, and a dense point matching module (see Figure 2). The encoder–decoder utilizes the KPConv [14] backbone as a feature extraction module and computes downsampling points of different levels (Section 2.3). The sparse superpoint matching module utilizes our RIG-Transformer to select matching superpoints located in overlapping regions to generate dense point clouds (Section 2.4). We partition dense points and superpoints into spatially clustered blocks. During training, we randomly select point cloud blocks with varying overlap ratios, and during testing, we choose dense points corresponding to superpoints selected by RIG-Transformer. Then, the dense point matching module directly regresses the correspondences of input dense point clouds, enabling the computation of relative transformation between dense point cloud pairs (Section 2.5). We introduce different loss functions to supervise superpoint matching module and dense point matching module to learn the correspondences and predict the transformation (Section 2.6). Our contributions are summarized as follows:



(a) Point clouds to be registered

(b) Estimated dense matching points

(c) Registered pairs of point clouds

Figure 1. Our TTReg predicts dense correspondences in the overlap region and estimates the transformation of point clouds with regions of low overlap. Points in red and green represent point clouds $\mathbf{P}_0$ and $\mathbf{Q}_0$, respectively, and gray lines represent the relationship of correspondences.

- We propose a Rotation-Invariant Geometric Transformer Cross-Encoder module (RIG-Transformer) that combines the geometric features and positional encoding of superpoint coordinates to extract more distinctive features for predicting superpoints located in the overlapping region.
- Through the fusion of our RIG-Transformer and Transformer Cross-Encoder, we introduce a Transformer-to-Transformer dense regression (TTReg) that leverages dense point clouds from overlapping regions for both training and testing phases to compute the transformation matrix.
- Through extensive experiments, our method showcases strong matching capabilities on public 3DMatch and ModelNet benchmark, with a notable improvement of 7.2% in matching recall on datasets with small overlap ratios.

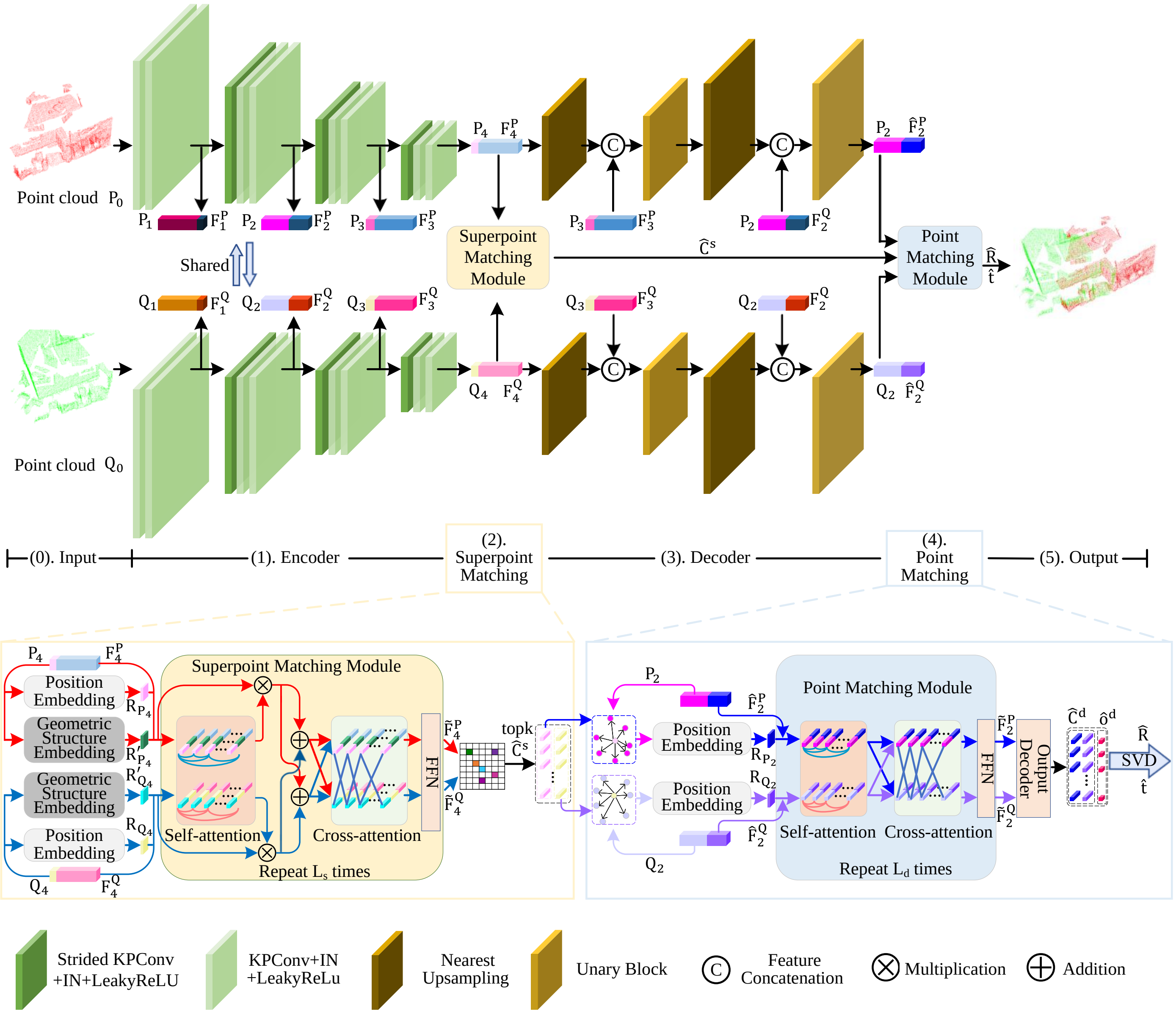


Figure 2. Overview of our TTReg architecture. $\mathbf{F}_4^\mathbf{P}$ and $\mathbf{F}_4^\mathbf{Q}$ are features of superpoints $\mathbf{P}_4$ and $\mathbf{Q}_4$. $\mathbf{F}_2^\mathbf{P}$ and $\mathbf{F}_2^\mathbf{Q}$ represent features of dense points $\mathbf{P}_2$ and $\mathbf{Q}_2$. Our RIG-Transformer serves as the superpoint matching module for selecting the optimal matching superpoint pairs $\hat{\mathbf{C}}^s$ within the overlap area. The point matching module encodes the feature $\mathbf{F}_2^\mathbf{P}$ and $\mathbf{F}_2^\mathbf{Q}$ of dense points $\mathbf{P}_2$ and $\mathbf{Q}_2$ corresponding to $\hat{\mathbf{C}}^s$, and predicts the dense correspondences $\hat{\mathbf{C}}^d$. Finally, the relative transformation matrices $\hat{\mathbf{R}}$ and $\hat{\mathbf{t}}$ are calculated utilizing dense correspondences $\hat{\mathbf{C}}^d$.

2.3. Feature Extraction and Correspondences Sampling

We utilize KPConv [14] as our feature extractor. The original point cloud pairs, represented as $\mathbf{P}_0 \in \mathbb{R}^{3\times M_0}$ and $\mathbf{Q}_0 \in \mathbb{R}^{3\times N_0}$, are voxelized to calculate the downsampled 3D points $\mathbf{P}_j \in \mathbb{R}^{3\times M_j}$ and $\mathbf{Q}_j \in \mathbb{R}^{3\times N_j}$, where M_j and N_j denote the number of 3D points obtained at each convolutional layer. Unlike Farthest Point Sampling (FPS) [15,16], we calculate the centroids of adjacent points within a voxel radius $\mathcal{V}$ to derive the downsampled point clouds $\mathbf{P}_j$ and $\mathbf{Q}_j$. These downsampled point clouds are then utilized for feature extraction in the subsequent KPConv layers, resulting in feature representations $\mathbf{F}_j^{\mathbf{P}} \in \mathbb{R}^{D_j\times M_j}$ and $\mathbf{F}_j^{\mathbf{Q}} \in \mathbb{R}^{D_j\times N_j}$.

The architecture for 3DMatch and 3DLoMatch illustrated in Figure 2 is adopted. We apply a 3-layer stridden KPConv convolutional structure to the original point cloud, involving three downsampling steps. Conversely, we perform two upsampling steps during the upsampling stage, resulting in one less upsampling step compared to the downsampling step. This is because point clouds are dense, requiring uniform voxelization to adapt to our correspondence loss function. The choice of upsampling steps follows the settings in Predator [17]. For the ModelNet and ModelLoNet datasets, a 2-layer downsampling and 1-layer upsampling encoder–decoder structure is employed.

To illustrate the sampling and aggregation method between sparse superpoints and dense point clouds used in our architecture of 3DMatch and 3DLoMatch (as shown in Figure 2), we first perform downsampling and feature extraction using the KPConv network. This process yields the lowest-level sparse 3D point cloud superpoints $\mathbf{P}_4$ and $\mathbf{Q}_4$, along with their corresponding features $\mathbf{F}_4^{\mathbf{P}}$ and $\mathbf{F}_4^{\mathbf{Q}}$. We adopt the data grouping method proposed in [17,18], where each superpoint serves as the center of a circle to divide the dense point clouds $\mathbf{P}_2$ and $\mathbf{Q}_2$ into 3D data blocks. The Euclidean distances between the dense points in $\mathbf{P}_4$ and $\mathbf{P}_2$, as well as $\mathbf{Q}_4$ and $\mathbf{Q}_2$, are computed. The dense points that are closest to the superpoints are assigned to the corresponding data blocks. The grouping method for mapping the dense points $\mathbf{Q}_0$ and $\mathbf{P}_0$ to superpoints is defined by Equation (2):

$$\mathcal{G}_k^{\mathcal{Q}_4} = \{\mathbf{q}_4 \in \mathbf{Q}_4 | k = argmin_l(\|\mathbf{q}_4 - \mathbf{q}_{2,l}\|_2), \mathbf{q}_{2,l} \in \mathbf{Q}_2\}, \tag{2}$$

where $\mathbf{q}_4$ represents a superpoint obtained by downsampling the point cloud $\mathbf{Q}_0$, and $\mathbf{q}_2$ denotes a dense 3D point of $\mathbf{Q}_0$ that needs to be grouped. The symbol $\|\cdot\|$ denotes the Euclidean distance of 3D points. The same grouping strategy is applied to the other point cloud $\mathbf{P}_0$.

After the grouping process of dense 3D point clouds, as shown in Figure 3, we calculate the neighborhood points for each superpoint $\mathbf{P}_4$ and $\mathbf{Q}_4$ by considering the points in $\mathcal{G}_k^{\mathcal{P}_4}$ and $\mathcal{G}_k^{\mathcal{Q}_4}$. For each superpoint, we compute the distance to its farthest neighboring point, which is used to measure the overlap region. By applying the relative transformation, we transform the superpoints $\mathbf{P}_4$ and the dense points $\mathbf{P}_2$ from point clouds $\mathbf{P}_0$ into the $\mathbf{Q}_0$ coordinates, denoted as $\mathbf{P}_2^{'}$, $\mathbf{P}_4^{'}$ from $\mathbf{P}_0^{'}$. We then measure the overlap between superpoint pairs $\mathbf{P}_2^{'}$ and $\mathbf{Q}_2$. The overlap is determined based on whether the dense points $\mathbf{P}_4^{'}$, $\mathbf{Q}_4$ are contained within the overlap region of superpoints $\mathbf{P}_2^{'}$, $\mathbf{Q}_2$ or not. The threshold for the overlapping region is represented as o_{thr}, which is used to select the dense point pairs. The selected superpoints in the overlap region of $\mathbf{P}_4$ and $\mathbf{Q}_4$ are denoted as $\mathbf{P}_4^o$ and $\mathbf{Q}_4^o$, respectively, and the dense points related to $\mathbf{P}_4^o$ and $\mathbf{Q}_4^o$ are denoted as $\mathbf{P}_2^o$ and $\mathbf{Q}_2^o$, respectively. We select dense points around superpoints based on the size of the overlapping region of the aligned point clouds. Dense point block pairs with larger overlapping regions are chosen to train our network architecture. In Figure 3, dense point block pairs of (a) are considered to be located in the overlapping region due to a large overlap area, while those in (b) and (j) are discarded as they either have no overlap or a small overlap region.

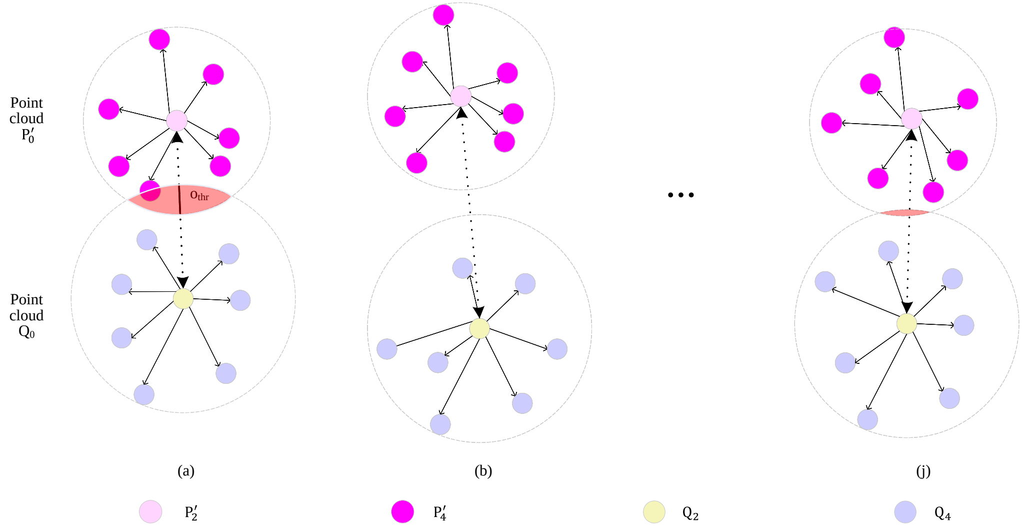


Figure 3. The selection of a superpoint and its corresponding dense points. (**a**) represents the selected dense matching point cloud block with a relatively large overlapping region, while (**b**) and (**j**) represent dense point cloud blocks with no or small overlapping region.

2.4. Superpoint Matching Module

We propose a Rotation-Invariant Geometric Transformer Cross-Encoder module, referred to as RIG-Transformer. Figure 4a depicts the computational flowchart, and Figure 4b,c depicts our RIG-self-attention and RIG-cross-attention, respectively. Through incorporating geometric features and positional encoding, we start by adding and normalizing the input features, a slight deviation from the typical attention mechanism. We compute the geometric features $\mathbf{R}'_{\mathbf{P}_4}$ and $\mathbf{R}'_{\mathbf{Q}_4}$, as well as the positional encoding $\mathbf{R}_{\mathbf{P}_4}$ and $\mathbf{R}_{\mathbf{Q}_4}$ of superpoint $\mathbf{P}_4$ and $\mathbf{Q}_4$. These values are then combined with the feature vectors $\mathbf{F}_4^{\mathbf{P}}$ and $\mathbf{F}_4^{\mathbf{Q}}$ of the superpoints and fed into RIG-Transformer to calculate the interaction features of the point cloud pairs.

For instance, considering the point cloud $\mathbf{P}_0$, we compute the feature vectors for RIG-self-attention, following the process and dimensions depicted in Figure 4. Utilizing the superpoint $\mathbf{P}_4$ and feature vector $\mathbf{F}_4^{\mathbf{P}}$ extracted by KPConv [14] as input, we not only calculate the initial attention [19] components $\mathbf{Q}_s$, $\mathbf{K}_s$, $\mathbf{V}_s$, but also compute weighted geometric feature encodings $\mathbf{G}_s$ and $\mathbf{R}_s$. Subsequently, we derive the geometric branch feature $\mathbf{E}_{\mathbf{P}_4}$ and contextual feature $\mathbf{C}_{\mathbf{P}_4}$ based on the attention weights. The definitions of the individual variables are provided as follows:

$$\mathbf{G}_s = \mathbf{R}'_{\mathbf{P}_4}\mathbf{W}_{\mathbf{G}_s}, \tag{3}$$

$$\mathbf{R}_s = \mathbf{R}'_{\mathbf{P}_4}\mathbf{W}_{\mathbf{R}_s}, \tag{4}$$

$$\mathbf{Q}_s = (\mathbf{F}_4^{\mathbf{P}} + \mathbf{R}_{\mathbf{P}_4})\mathbf{W}_{\mathbf{Q}_s}, \tag{5}$$

$$\mathbf{K}_s = (\mathbf{F}_4^{\mathbf{P}} + \mathbf{R}_{\mathbf{P}_4})\mathbf{W}_{\mathbf{K}_s}, \tag{6}$$

$$\mathbf{V}_s = (\mathbf{F}_4^{\mathbf{P}} + \mathbf{R}_{\mathbf{P}_4})\mathbf{W}_{\mathbf{V}_s}, \tag{7}$$

$$\mathbf{E}_{\mathbf{P}} = Softmax(\frac{\mathbf{Q}_s\mathbf{R}_s + \mathbf{Q}_s\mathbf{K}_s^T}{\sqrt{D_s}})\mathbf{G}_s, \tag{8}$$

$$\mathbf{C}_{\mathbf{P}} = Softmax(\frac{\mathbf{Q}_s\mathbf{R}_s + \mathbf{Q}_s\mathbf{K}_s^T}{\sqrt{D_s}})\mathbf{V}_s, \tag{9}$$

where $\mathbf{W}_{\mathbf{G}_s}$, $\mathbf{W}_{\mathbf{R}_s}$ represent the learnable weights for geometric features; $\mathbf{W}_{\mathbf{Q}_s}$, $\mathbf{W}_{\mathbf{K}_s}$, and $\mathbf{W}_{\mathbf{V}_s}$ denote the learnable self-attention weights for the features of superpoints $\mathbf{P}_4$ and $\mathbf{Q}_4$;

and D_s indicates the dimension of the features $\mathbf{F}_4^{\mathbf{P}}$ and $\mathbf{F}_4^{\mathbf{Q}}$ of $\mathbf{P}_4$ and $\mathbf{Q}_4$. It is noteworthy that $\mathbf{W}_{\mathbf{G}_s}$, $\mathbf{W}_{\mathbf{R}_s}$, $\mathbf{W}_{\mathbf{Q}_s}$, $\mathbf{W}_{\mathbf{K}_s}$, and $\mathbf{W}_{\mathbf{V}_s}$ share weights between point clouds $\mathbf{P}_0$ and $\mathbf{Q}_0$. Similarly, taking $\mathbf{Q_4}$ and $\mathbf{F}_4^{\mathbf{Q}}$ as inputs to the self-attention module RIG-self-attention, we replace the corresponding input variables according to the computation methods in Equations (3)–(9) to calculate $\mathbf{E}_{\mathbf{Q}_4}$ and $\mathbf{C}_{\mathbf{Q}_4}$ for subsequent feature interactions.

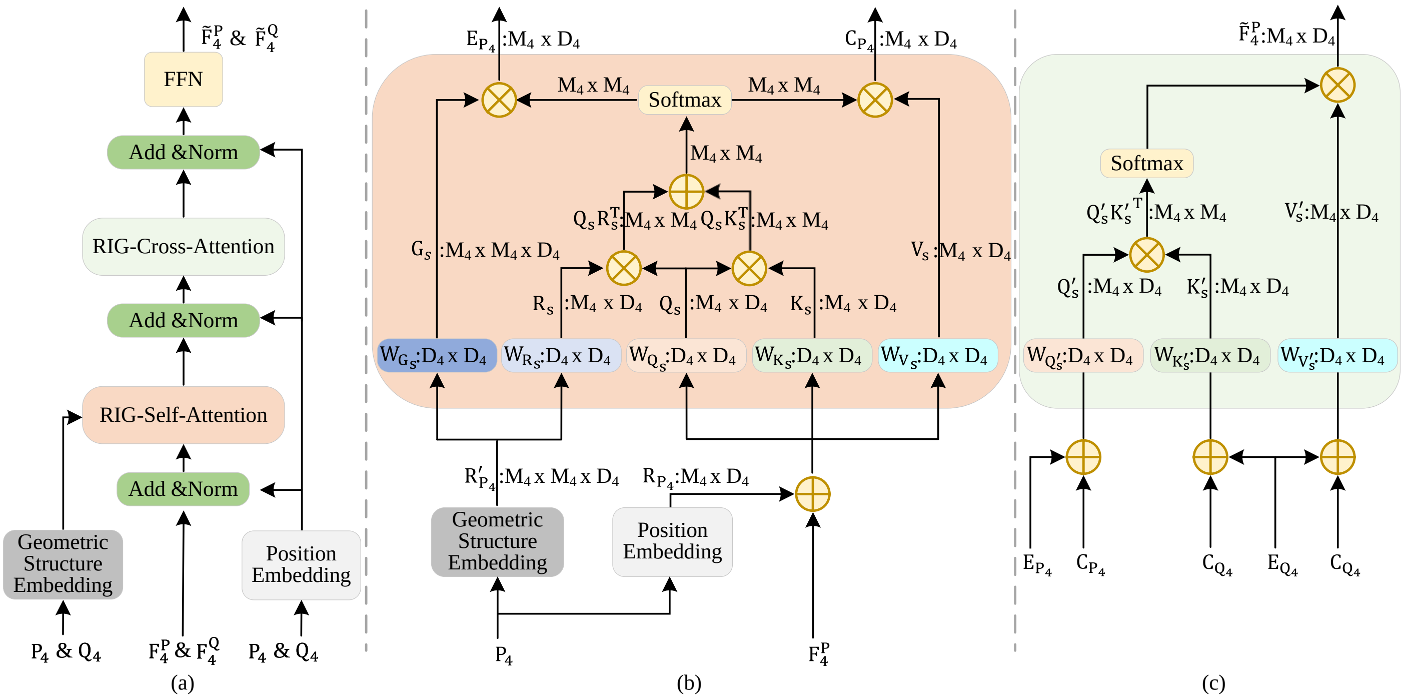


Figure 4. Overview of our RIG-Transformer module: (**a**) depicts an overall computation of the RIG-Transformer module, (**b**,**c**) depict the RIG-self-attention and RIG-cross-attention.

We input the calculated $\mathbf{E}_{\mathbf{P}_4}$, $\mathbf{C}_{\mathbf{P}_4}$, $\mathbf{E}_{\mathbf{Q}_4}$, and $\mathbf{C}_{\mathbf{Q}_4}$ from the above computations into RIG-cross-attention, as illustrated in Figure 4c. We compute the rotation-invariant geometric feature $\widetilde{\mathbf{F}}_4^{\mathbf{P}}$ after the interaction, with the calculation of variables defined as follows:

$$\mathbf{Q}_s^{'} = (\mathbf{E}_{\mathbf{P}} + \mathbf{C}_{\mathbf{P}})\mathbf{W}_{\mathbf{Q}_s^{'}}, \tag{10}$$

$$\mathbf{K}_s^{'} = (\mathbf{E}_{\mathbf{Q}} + \mathbf{C}_{\mathbf{Q}})\mathbf{W}_{\mathbf{K}_s^{'}}, \tag{11}$$

$$\mathbf{V}_s^{'} = (\mathbf{E}_{\mathbf{Q}} + \mathbf{C}_{\mathbf{Q}})\mathbf{W}_{\mathbf{V}_s^{'}}, \tag{12}$$

$$\widetilde{\mathbf{F}}_4^{\mathbf{P}} = Softmax(\frac{\mathbf{Q}_s^{'}\mathbf{K}_s^{'T}}{\sqrt{D_s}})\mathbf{V}_s^{'}, \tag{13}$$

where $\mathbf{W}_{\mathbf{Q}_s^{'}}$, $\mathbf{W}_{\mathbf{K}_s^{'}}$ and $\mathbf{W}_{\mathbf{V}_s^{'}}$ are the learnable shared weights for point clouds $\mathbf{P}_4$ and $\mathbf{Q}_4$. By swapping the input order of $\mathbf{E}_{\mathbf{P}_4}$ and $\mathbf{E}_{\mathbf{Q}_4}$, $\mathbf{C}_{\mathbf{P}_4}$ and $\mathbf{C}_{\mathbf{Q}_4}$ in Figure 4c, we obtain the output features $\widetilde{\mathbf{F}}_4^{\mathbf{Q}}$. As shown in Figure 2, the obtained $\widetilde{\mathbf{F}}_4^{\mathbf{P}}$ and $\widetilde{\mathbf{F}}_4^{\mathbf{Q}}$ are further processed through the FFN module, which consists of two layers of linear transformation units, to facilitate multiple interactions and the fusion of features. The calculated features serve as new $\widetilde{\mathbf{F}}_4^{\mathbf{P}}$ and $\widetilde{\mathbf{F}}_4^{\mathbf{Q}}$. We iterate the computation of RIG-Transformer L_s times to enhance the correlation between intrinsic and interaction features of the point clouds, resulting in the final output feature vectors $\widetilde{\mathbf{F}}_4^{\mathbf{P}}$ and $\widetilde{\mathbf{F}}_4^{\mathbf{Q}}$ with geometric properties and rotation invariance.

To extract the optimal matching dense 3D points $\hat{\mathbf{C}}^d$, during the training phase, we compute neighboring points from $\mathbf{P}_4$ and $\mathbf{Q}_4$ and randomly select matching superpoints $\hat{\mathbf{C}}^s$, followed by selecting densely related points for training our network. During the testing phase, we calculate the optimal matching superpoints and then select the densely related points in the dense point matching module. To select the best matching superpoints, we first normalize the features $\widetilde{\mathbf{F}}_4^{\mathbf{P}}$ and $\widetilde{\mathbf{F}}_4^{\mathbf{Q}}$ mixed with geometric features and positional encoding. Subsequently, we compute the Gaussian correlation matrix $\widetilde{\mathbf{C}}^{\mathbf{s}}$ for the normalized features, with the Gaussian correlation formula for a pair of superpoints defined as follows:

$$\widetilde{\mathbf{C}}^{\mathbf{s}}_{\mathbf{ij}} = exp(\left\|\widetilde{\mathbf{F}}_4^{p_i} - \widetilde{\mathbf{F}}_4^{q_j}\right\|_2^2). \tag{14}$$

To enhance the discriminative feature matching of superpoints, we normalize the correlation matrix $\widetilde{\mathbf{C}}^{\mathbf{s}}_{\mathbf{ij}}$ using bidirectional normalization [7,20,21]. Bidirectional normalization refers to normalizing the feature correlation matrix in rows and columns separately. The formula is defined as follows:

$$\hat{\mathbf{C}}^{\mathbf{s}}_{\mathbf{ij}} = \frac{\widetilde{\mathbf{C}}^{\mathbf{s}}_{\mathbf{ij}}}{\sum_{k=1}^{|\mathbf{P}_4|}\widetilde{\mathbf{C}}^{\mathbf{s}}_{\mathbf{ik}}} \cdot \frac{\widetilde{\mathbf{C}}^{\mathbf{s}}_{\mathbf{ij}}}{\sum_{k=1}^{|\mathbf{Q}_4|}\widetilde{\mathbf{C}}^{\mathbf{s}}_{\mathbf{kj}}}. \tag{15}$$

After bidirectional normalization, we select the top K best-matching pairs of superpoints based on the matching scores and extract the related dense points. These dense matching points not only possess richer geometric features but are also located in overlapping regions. This reduces the impact on point cloud registration performance from points structurally similar but located outside the overlapping regions.

2.5. Point Matching Module

Due to the quadratic relationship between weight matrices and data length in the attention mechanism of Transformer, existing methods are unable to directly process dense point clouds with Transformer. Furthermore, numerous dense point pairs correspond to points in the overlapping areas. Thus, we choose the top K matching point pairs $\hat{\mathbf{C}}^s$ using the predicted superpoints containing global information. Subsequently, we index the dense 3D matching point pairs located in overlapping regions based on the relationship between superpoints and dense points. The predicted dense matching points are denoted as $\hat{\mathbf{C}}^d$, containing the dense three-dimensional points $\mathbf{P_2}$ and $\mathbf{Q_2}$ of point clouds $\mathbf{P_0}$ and $\mathbf{Q_0}$, along with their respective features $\hat{\mathbf{F}}_2^{\mathbf{P}}$ and $\hat{\mathbf{F}}_2^{\mathbf{Q}}$.

Since $\hat{\mathbf{F}}_2^{\mathbf{P}}$ and $\hat{\mathbf{F}}_2^{\mathbf{Q}}$ are dense point cloud features extracted by the KPConv network, they have more accurate coordinates compared to sparse superpoints. We utilize a Transformer Cross-Encoder to interactively exchange information among the dense point features located within the overlapping regions outputted by the RIG-Transformer module. In the interaction, we fuse the positional encoding [10,19] of the respective dense point coordinates. The incorporation of positional encoding not only enhances the robustness of the features but also allows us to use them for subsequent prediction of corresponding point coordinates and determining if the predicted points reside in overlapping areas. Based on the predicted dense corresponding points, we calculate the transformation matrix.

Our dense point cloud matching module is primarily composed of a Transformer Cross-Encoder module and an output decoding module. The Transformer Cross-Encoder module consists of a self-attention module, cross-attention module, and FFN module. In Figure 5a, the overall calculation process of attention is depicted, Figure 5b represents the calculation of self-attention for feature extraction, and Figure 5c illustrates the attention interaction calculation between two point clouds. The FFN module comprises two layers of linear transformation units, serving to transform and integrate feature channels in the interaction of multi-layer attention.

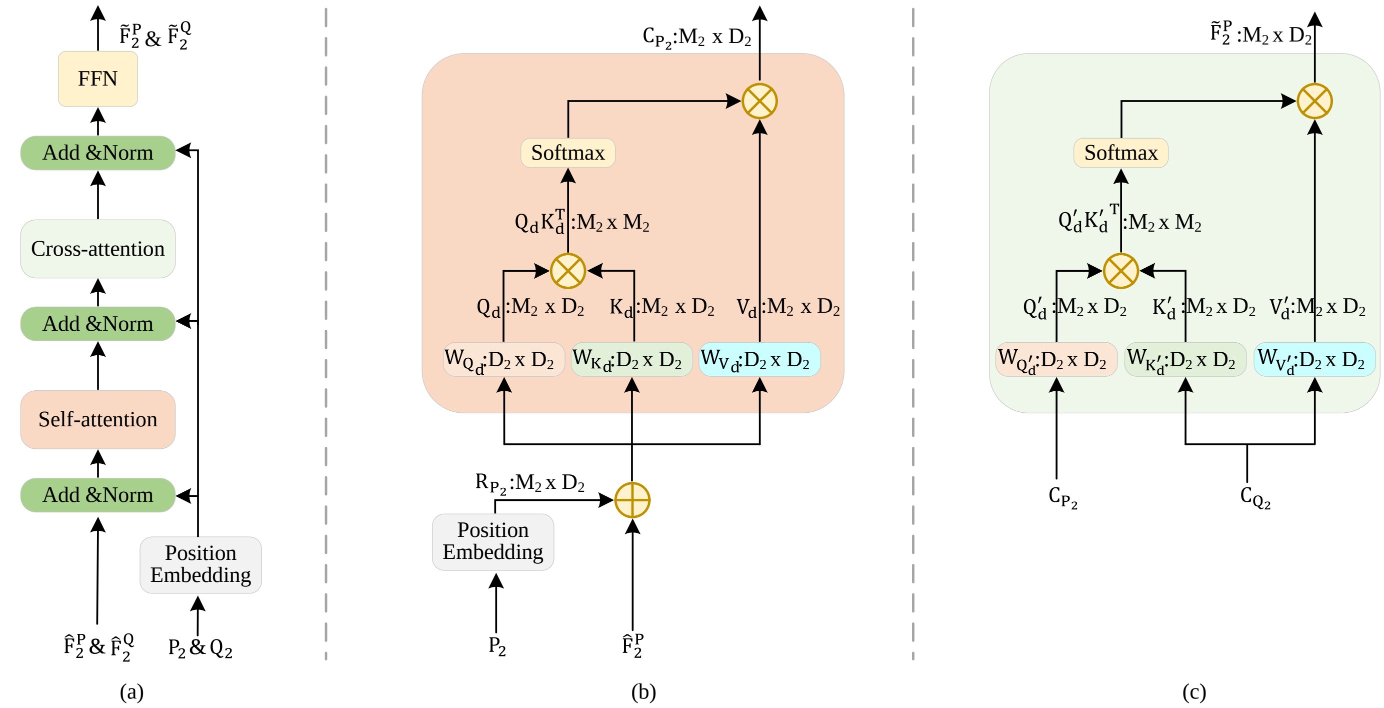


Figure 5. The dense matching module structure: (**a**) depicts the overall computation of the dense matching module, (**b**,**c**) depict the calculation of the self-attention and cross-attention modules.

Taking dense spatial coordinates $\mathbf{P_2}$ and the features $\hat{\mathbf{F}}_2^{\mathbf{P}}$ of point cloud $\mathbf{P}_0$ located in the overlapping region as an example, we illustrate the feature fusion and interaction of dense point feature pairs in Figure 5b,c. Firstly, $\mathbf{P}_2$ are encoded using sine encoding, followed by the fusion with the feature $\hat{\mathbf{F}}_2^{\mathbf{P}}$ to calculate the self-attention context feature $\mathbf{C}_{\mathbf{P}_2}$. The formulas for feature fusion and the self-attention of $\mathbf{Q}_d$, $\mathbf{K}_d$, and $\mathbf{V}_d$ and the context feature $\mathbf{C}_{\mathbf{P}_2}$ are defined as follows:

$$\mathbf{Q}_d = (\hat{\mathbf{F}}_2^{\mathbf{P}} + \mathbf{R}_{\mathbf{P}_2})\mathbf{W}_{\mathbf{Q}_d}, \tag{16}$$

$$\mathbf{K}_d = (\hat{\mathbf{F}}_2^{\mathbf{P}} + \mathbf{R}_{\mathbf{P}_2})\mathbf{W}_{\mathbf{K}_d}, \tag{17}$$

$$\mathbf{V}_d = (\hat{\mathbf{F}}_2^{\mathbf{P}} + \mathbf{R}_{\mathbf{P}_2})\mathbf{W}_{\mathbf{V}_d}, \tag{18}$$

$$\mathbf{C}_2^{\mathbf{P}} = Softmax(\frac{\mathbf{Q}_d{\mathbf{K}_d}^T}{\sqrt{D_d}})\mathbf{V}_s, \tag{19}$$

where the variables $\mathbf{Q}_d$, $\mathbf{K}_d$, and $\mathbf{V}_d$ of self-attention are shared. The corresponding learnable weights $\mathbf{W}_{\mathbf{Q}_d}$, $\mathbf{W}_{\mathbf{K}_d}$, and $\mathbf{W}_{\mathbf{V}_d}$ are shared by the dense point clouds $\mathbf{P}_2$ and $\mathbf{Q}_2$ of point clouds $\mathbf{P}_0$ and $\mathbf{Q}_0$. Similarly, we replace the input variables of the self-attention module (b) in the Transformer Cross-Encoder in Figure 5 with $\mathbf{Q}_2$ and $\hat{\mathbf{F}}_2^{\mathbf{Q}}$ and compute the global context feature vector $\mathbf{C}_{\mathbf{Q}_2}$ of point cloud $\mathbf{Q}_0$ based on Equations (16)–(19). Consequently, we have obtained the context feature vectors $\mathbf{C}_{\mathbf{P}_2}$ and $\mathbf{C}_{\mathbf{Q}_2}$ of the matching point pairs of dense point clouds $\mathbf{P}_0$ and $\mathbf{Q}_0$ within the overlapping region, which serve as inputs to the cross-attention module for further feature fusion.

The computational process of the cross-attention is illustrated in Figure 5c. Subsequently, we input $\mathbf{C}_{\mathbf{P}_2}$ and $\mathbf{C}_{\mathbf{Q}_2}$ into the cross-attention module (c) in sequential order (first $\mathbf{C}_{\mathbf{P}_2}$ and then $\mathbf{C}_{\mathbf{Q}_2}$), and compute the interacted feature $\tilde{\mathbf{F}}_2^{\mathbf{P}}$, which integrates more accurate encoding of dense spatial coordinates and contains global feature interaction information of the point cloud to be matched, thus possessing better discriminative power. The corresponding formula for this computation is defined as follows:

$$\mathbf{Q}_d^{'} = \mathbf{C}_{\mathbf{P}_2}\mathbf{W}_{\mathbf{Q}_d^{'}}, \tag{20}$$

$$\mathbf{K}_s^{'} = \mathbf{C}_{\mathbf{Q}_2}\mathbf{W}_{\mathbf{K}_d^{'}}, \tag{21}$$

$$\mathbf{V}_s^{'} = \mathbf{C}_{\mathbf{Q}_2}\mathbf{W}_{\mathbf{V}_d^{'}}, \tag{22}$$

$$\widetilde{\mathbf{F}}_2^{\mathbf{P}} = Softmax(\frac{\mathbf{Q}_d^{'}\mathbf{K}_d^{'\,T}}{\sqrt{D_d}})\mathbf{V}_d^{'}, \tag{23}$$

where $\mathbf{W}_{\mathbf{Q}_d^{'}}$, $\mathbf{W}_{\mathbf{K}_d^{'}}$, and $\mathbf{W}_{\mathbf{V}_d^{'}}$ are the learnable parameters of the cross-attention variables $\mathbf{Q}_d^{'}$, $\mathbf{Q}_d^{'}$, and $\mathbf{Q}_d^{'}$. Similarly, we interchange the input order of $\mathbf{C}_{\mathbf{P}_2}$ and $\mathbf{C}_{\mathbf{Q}_2}$ in Figure 5c and the corresponding computation Formulas (20)–(23). By first computing in the sequence of $\mathbf{C}_{\mathbf{Q}_2}$ and then $\mathbf{C}_{\mathbf{P}_2}$, we obtain the feature $\widetilde{\mathbf{F}}_2^{\mathbf{Q}}$ for dense point pairs located in the overlapping region. The resulting $\widetilde{\mathbf{F}}_2^{\mathbf{P}}$ and $\widetilde{\mathbf{F}}_2^{\mathbf{Q}}$ contain positional encoding and feature information from the other, representing dense global features with enhanced discriminative power. These features will be directly utilized in the output decoding module to predict corresponding point coordinates and overlap scores.

We feed $\widetilde{\mathbf{F}}_2^{\mathbf{P}}$ and $\widetilde{\mathbf{F}}_2^{\mathbf{Q}}$ as inputs to the output decoding module, which consists of four linear layers. Among these layers, three are utilized for predicting the corresponding coordinates of points, while the fourth linear layer is responsible for predicting scores indicating whether the matched points are within the overlapping area. The network structure is illustrated in Figure 6, and the detailed mathematical expressions are presented as follows:

$$\begin{cases} \hat{\mathbf{P}}_2^o = ReLU(ReLU(\widetilde{\mathbf{F}}_2^{\mathbf{Q}}\mathbf{W}_1 + \mathbf{b}_1)\mathbf{W}_2 + \mathbf{b}_2))\mathbf{W}_3 + \mathbf{b}_3 \\ \hat{\mathbf{Q}}_2^o = ReLU(ReLU(\widetilde{\mathbf{F}}_2^{\mathbf{P}}\mathbf{W}_1 + \mathbf{b}_1)\mathbf{W}_2 + \mathbf{b}_2))\mathbf{W}_3 + \mathbf{b}_3 \end{cases}, \tag{24}$$

$$\begin{cases} \hat{\mathbf{o}}_{\mathbf{P}_2} = \widetilde{\mathbf{F}}_2^{\mathbf{Q}}\mathbf{W}_1^{'} + \mathbf{b}_1^{'} \\ \hat{\mathbf{o}}_{\mathbf{Q}_2} = \widetilde{\mathbf{F}}_2^{\mathbf{P}}\mathbf{W}_1^{'} + \mathbf{b}_1^{'} \end{cases}, \tag{25}$$

where $\mathbf{W}_1$, $\mathbf{W}_2$, $\mathbf{W}_3$, $\mathbf{b}_1$, $\mathbf{b}_2$, $\mathbf{b}_3$, $\mathbf{W}_1^{'}$, and $\mathbf{b}_1^{'}$ are the learnable weights of the corresponding point and overlap prediction linear layers in the output decoding module. We concatenate the dense points located in the common area $\mathbf{P}_2^o$ with the predicted corresponding points $\hat{\mathbf{Q}}_2^o$. Similarly, we concatenate the dense points $\mathbf{Q}_2^o$ within the common area with the network-predicted corresponding points $\hat{\mathbf{Q}}_2^o$. After concatenation, we obtain dense matched points $\hat{\mathbf{P}}_2^d$ and $\hat{\mathbf{Q}}_2^d$ in the overlapping region with a data length of $M_2^o + N_2^o$. The corresponding overlap score is denoted as $\hat{\mathbf{o}}_2^d$, and the formulas are defined as follows:

$$\hat{\mathbf{P}}^d = \begin{bmatrix} \mathbf{P}_2^o \\ \hat{\mathbf{P}}_2^o \end{bmatrix}, \hat{\mathbf{Q}}^d = \begin{bmatrix} \hat{\mathbf{Q}}_2^o \\ \mathbf{Q}_2^o \end{bmatrix}, \hat{\mathbf{o}}^d = \begin{bmatrix} \hat{\mathbf{o}}_{\mathbf{P}_2} \\ \hat{\mathbf{o}}_{\mathbf{Q}_2} \end{bmatrix}, \tag{26}$$

we calculate the relative transformation matrix $\hat{\mathbf{R}}$ and $\hat{\mathbf{t}}$ of the point cloud to be matched by minimizing the loss function of dense matching points:

$$\hat{\mathbf{R}}, \hat{\mathbf{t}} = \arg\min \sum_{j}^{M_2^o + N_2^o} \hat{\mathbf{o}}_j \left\| \mathbf{R}\hat{\mathbf{p}}_j + \mathbf{t} - \hat{\mathbf{q}}_j \right\|^2, \tag{27}$$

where M_2^o and N_2^o represent the data lengths of $\mathbf{P}_2^o$ and $\hat{\mathbf{Q}}_2^o$, $\mathbf{Q}_2^o$, and $\hat{\mathbf{P}}_2^o$, respectively. $\hat{\mathbf{p}}_j$, $\hat{\mathbf{q}}_j$, and $\hat{\mathbf{o}}_j$ are the estimated corresponding points and overlap score weights from Equation (26). We obtain our estimated relative pose transformation matrix by solving Equation (27) using the Kabsch–Umeyama algorithm [22,23] for the optimal solution.

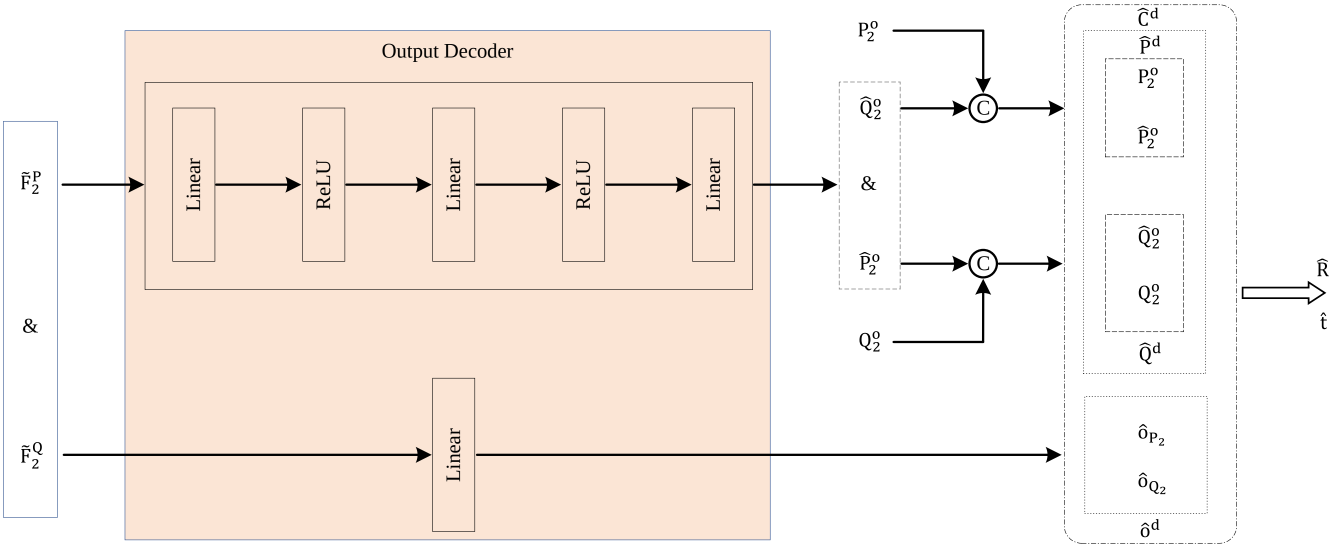


Figure 6. The overview of output decoder structure and transformation calculation.

2.6. Loss Function

Our optimization target loss $\mathcal{L}$ comprises two parts: the superpoint loss $\mathcal{L}_s$ and the dense point loss $\mathcal{L}_d$, i.e., $\mathcal{L} = \mathcal{L}_s + \mathcal{L}_d$. The superpoint loss function constrains and predicts superpoint correspondences located in the overlapping region, while the dense point loss function directly enforces the architecture to predict dense correspondences.

2.6.1. Superpoint Correspondences Loss Function

We utilize the overlapping circle loss from [7] to choose the K best pairs of superpoints with the greatest similarity scores. The function is defined as follows:

$$\mathcal{L}_s^{\mathcal{P}_0} = \frac{1}{|\mathcal{A}|} \sum_{\mathcal{G}_j^{\mathcal{P}} \in \mathcal{A}} \log[1 + \sum_{\mathcal{G}_k^{\mathcal{Q}} \in \varepsilon_p^j} e^{\lambda_j^k \beta_p^{j,k} (d_j^k - \Delta_p)} \cdot \sum_{\mathcal{G}_l^{\mathcal{Q}} \in \varepsilon_n^j} e^{\beta_n^{j,l} (\Delta_n - d_j^k)}], \tag{28}$$

where $d_j^k {=} \|\widetilde{\mathbf{F}}_s^{p_j} - \widetilde{\mathbf{F}}_s^{q_k}\|_2$ represents the distance between feature vectors and $\widetilde{\mathbf{F}}_s^{p_j}$ and $\widetilde{\mathbf{F}}_s^{q_k}$ are the features of superpoints in point clouds $\mathbf{P}_0$ and $\mathbf{Q}_0$, respectively. Following the three-level downsampling of point clouds as shown in Figure 2, we have $\widetilde{\mathbf{F}}_s^{p_j} \in \widetilde{\mathbf{F}}_s^{P_4}$ and $\widetilde{\mathbf{F}}_s^{q_k} \in \widetilde{\mathbf{F}}_s^{Q_4}$ and $\lambda_j^k = (o_j^k)^{\frac{1}{2}}$, where o_j^k denotes the degree of overlap between dense points corresponding to the superpoint pair $\mathcal{G}_j^{\mathcal{P}}$ and $\mathcal{G}_k^{\mathcal{Q}}$. We use $\beta_p^{j,k} = \gamma(d_j^k - \Delta_p)$ and $\beta_n^{j,l} = \gamma(\Delta_n - d_j^l)$ to weight the matching points and non-matching points, respectively, enhancing the discriminative power of the function, where we set the hyperparameters $\Delta_p = 0.1$ and $\Delta_n = 1.4$ as suggested in [7]. Similarly, the loss function $\mathcal{L}_s^{\mathcal{Q}_0}$ for $\mathbf{Q}_0$ is computed using a similar method, resulting in the complete superpoint loss function $\mathcal{L}_s = (\mathcal{L}_s^{\mathcal{P}_0} + \mathcal{L}_s^{\mathcal{Q}_0})/2$.

2.6.2. Point Correspondences Loss Function

The dense point loss function comprises three components: overlap loss, corresponding point loss, and feature loss, i.e., $\mathcal{L}_d = \mathcal{L}_{do} + \mathcal{L}_{dc} + \mathcal{L}_{df}$, which, respectively, constrain the overlapping regions in three-dimensional space, match corresponding points in three-dimensional space, and supervise the learning of the joint network for feature vector space matching.

Overlap Loss

Through the superpoint matching module in Section 2.4, we obtained dense point pairs located in the overlapping region. To facilitate the network in acquiring additional characteristics of corresponding areas, we divide the point cloud match pairs into overlapping and non-overlapping regions. We further constrain the dense points corresponding to matching superpoints with similar structures but low overlap rates using a cross-entropy loss function. This helps reduce matching errors and improve matching accuracy, and the formula for overlap score constraint is defined as follows:

$$\mathcal{L}_{do}^{\mathcal{P}_0} = -\frac{1}{M_d}\sum_{j}^{M_d} o_{p_j}^{gt} \cdot \log \hat{o}_{p_j} + \left(1 - o_{p_j}^{gt}\right) \cdot \log\left(1 - \hat{o}_{p_j}\right), \tag{29}$$

where $\hat{o}_{p_j}$ represents the likelihood score of the network predicting if points are belong the overlapping area, we calculate $o_{p_j}^{gt}$ utilizing the method proposed in [17,24], defined by the formula:

$$o_{p_j}^{gt} = \begin{cases} 1, & \left\|\mathcal{T}^{gt}(p_j) - NN(\mathcal{T}^{gt}(p_j), \mathbf{Q}_0)\right\| < r_d \\ 0, & otherwise \end{cases}, \tag{30}$$

where $\mathcal{T}^{gt}$ represents the rigid transformation matrix of the ground truth relative pose change for the pair of point clouds, r_d is the threshold to determine whether a pair of dense matching points match, and $NN(\cdot)$ denotes the spatial nearest neighbor calculation. Similarly, we can derive the dense overlap loss function $\mathcal{L}_{do}^{\mathcal{Q}_0}$ of $\mathbf{Q}_0$, and the complete dense point overlap loss function is given by $\mathcal{L}_{do} = \mathcal{L}_{do}^{\mathcal{P}_0} + \mathcal{L}_{do}^{\mathcal{Q}_0}$.

Corresponding Point Loss

We constrain the three-dimensional points in the overlapping region by minimizing the Euclidean distance ℓ^1 of the corresponding points in three-dimensional space. For dense points outside the overlapping region, we use the overlap degree as a weight. The formula is defined as follows:

$$\mathcal{L}_{dc}^{\mathcal{P}_0} = \frac{1}{\sum_j o_{\mathbf{p}_j}^{gt}} \sum_{j}^{M_d} o_{\mathbf{p}_j}^{gt} \left|\mathcal{T}^{gt}\left(\mathbf{p}_j^d\right) - \hat{\mathbf{q}}_j^d\right|, \tag{31}$$

where $o_{p_j}^{gt}$ is the ground truth overlap degree, p_j^d and $\hat{q}_k^d$ are a pair of dense matching points as in Equation (26), and M_d is the number of dense three-dimensional points within the overlapped region in point cloud $\mathbf{P}_0$. Similarly, we calculate the corresponding point loss function $\mathcal{L}_{dc}^{\mathcal{Q}_0}$ for the point cloud $\mathbf{Q}_0$. The overall corresponding point loss function is given by $\mathcal{L}_{dc} = \mathcal{L}_{dc}^{\mathcal{P}_0} + \mathcal{L}_{dc}^{\mathcal{Q}_0}$.

Feature Loss

We utilize the infoNCE loss [10] to supervise the network learning of dense point feature vectors, encouraging the network to learn more similar features for matching points. Here, $p^d \in \hat{\mathbf{P}}^d$ and $q^d \in \hat{\mathbf{Q}}^d$ represent a pair of dense matching points as in Equation (26). The definition of the feature loss function infoNCE is as follows:

$$\mathcal{L}_{df}^{\mathcal{P}_0} = -\mathbb{E}_{\mathbf{p}^d \in \hat{\mathbf{P}}^\mathbf{d}}\left[\log \frac{g\left(\mathbf{p}^d, \mathbf{p}_{\mathbf{p}^d}\right)}{g\left(\mathbf{p}^d, \mathbf{p}_{\mathbf{p}^d}\right) + \sum_{\mathbf{n}_{\mathbf{p}^d}} g\left(\mathbf{p}, \mathbf{n}_{\mathbf{p}^d}\right)}\right]. \tag{32}$$

We measure the similarity of features using a logarithmic bilinear function [10], where the function $g(\cdot,\cdot)$ is defined as:

$$g(\mathbf{x}, \mathbf{q}) = exp(\bar{\mathbf{g}}_{\mathbf{p}}^T \mathbf{W}_g \bar{\mathbf{g}}_q^T), \tag{33}$$

where $\mathbf{g}_p \in \widetilde{\mathbf{F}}_d^P$, with $\widetilde{\mathbf{F}}_d^P$ representing the dense feature obtained from the three times downsampled and two times upsampled feature extraction network structure shown in Figure 2. $\bar{\mathbf{g}}_q$ is the feature vector of its corresponding point. $\mathbf{p}_{\mathbf{p}_d}$ and $\mathbf{n}_{\mathbf{p}_d}$ indicate whether the point in $\hat{\mathbf{Q}}^d$ matches $\mathbf{p}^d$, with the matching determined by the positive boundary $\mathcal{V}$

and negative boundary $2\mathcal{V}$, where $\mathcal{V}$ is the radius of the voxel size. $\mathbf{W}_f$ is a learnable weight matrix that is diagonal and symmetric. Similarly, the overall feature loss is given by $\mathcal{L}_{df} = \mathcal{L}_{df}^{\mathcal{P}_0} + \mathcal{L}_{df}^{\mathcal{Q}_0}$.

3. Results

3.1. Datasets

3.1.1. Indoor Benchmarks: 3DMatch and 3DLoMatch

The 3DMatch [25] and 3DLoMatch [17] datasets were introduced to address the challenges of 3D scene understanding and alignment. The datasets consist of RGB-D scans of various indoor scenes, and provide aligned point clouds and RGB images for each scene, along with ground truth transformations that represent the accurate relative poses between pairs of point clouds. One key feature of the datasets is their diversity in terms of scene types, object categories, and sensor noise. The scenes include different indoor environments, such as living rooms, kitchens, and offices, with varying levels of clutter and occlusion. This diversity helps to analyze generalization capabilities for point cloud registration algorithms. The 3DLoMatch dataset contains significant geometric variations, occlusions, and partial overlaps (between 10% and 30%), but the overlaping regions are smaller than those in 3DMatch (>30%), making accurate alignment and pose estimation difficult. This makes the dataset suitable for evaluating the performance of various point cloud registration methods under realistic conditions.

3.1.2. Synthetic Benchmarks: ModelNet and ModelLoNet

We also utilize the ModelNet40 [26] benchmark to further assess our model. We follow the dataset settings proposed by [10,17,27] to obtain ModelNet and ModelLoNet, respectively. These datasets exhibit varying average overlapping regions, with ModelNet at 73.5% overlap and ModelLoNet at 53.6%. The ModelNet40 dataset provides a well-balanced distribution of object categories, including chairs, tables, airplanes, cars, and more, guaranteeing representation from diverse classes. The objects within this dataset are captured from multiple angles and poses, offering a realistic and comprehensive depiction of real-world objects. This diversity presents challenges for algorithms, as they need to handle the different orientations, partial views, and inherent noise present in the data.

3.2. Experiment Details

For 3DMatch and ModelNet40, we set the voxel size $\mathcal{V}$ to 0.025 m and 0.015 m, respectively, with the voxel size doubling at each downsampling step. Training is conducted only on 3DMatch and ModelNet datasets, and evaluation testing is performed not only on 3DMatch and ModelNet, but also on 3DLoMatch and ModelLonet. We select 32 superpoints, with a maximum of 64 dense points associated with each superpoint in the training and testing phases of 3DMatch. For the ModelNet40 dataset, we unify the data length of training superpoint, testing superpoints, as well as the maximum length of dense points associated with each superpoint to 128. We utilize the AdamW optimizer with a consistent initial learning rate of 0.0001. For 3DMatch, the learning rate is decreased by half every 20 epochs, whereas for ModelNet, it is halved every 100 epochs. Training concludes upon reaching 900k iterations. The training and testing processes are carried out on an Nvidia RTX 3090Ti GPU. We set the batch size to 1 for both 3DMatch and ModelNet40.

3.3. Evaluation

3.3.1. Evaluation of 3DMatch and 3DLoMatch

In order to evaluate our approach's effectiveness, we utilize the registration recall (RR) metric configuration proposed in [10,17,28] for measuring the success rate of registration. Moreover, we apply the relative rotation error (RRE) to evaluate the accuracy of rotation matrices and the relative translation error (RTE) to assess discrepancies in translation vector estimations, commonly employed for analyzing transformation matrix errors.

RegTR [10] directly regresses poses using sparse superpoints, which serves as our baseline for assessment (see Table 1).

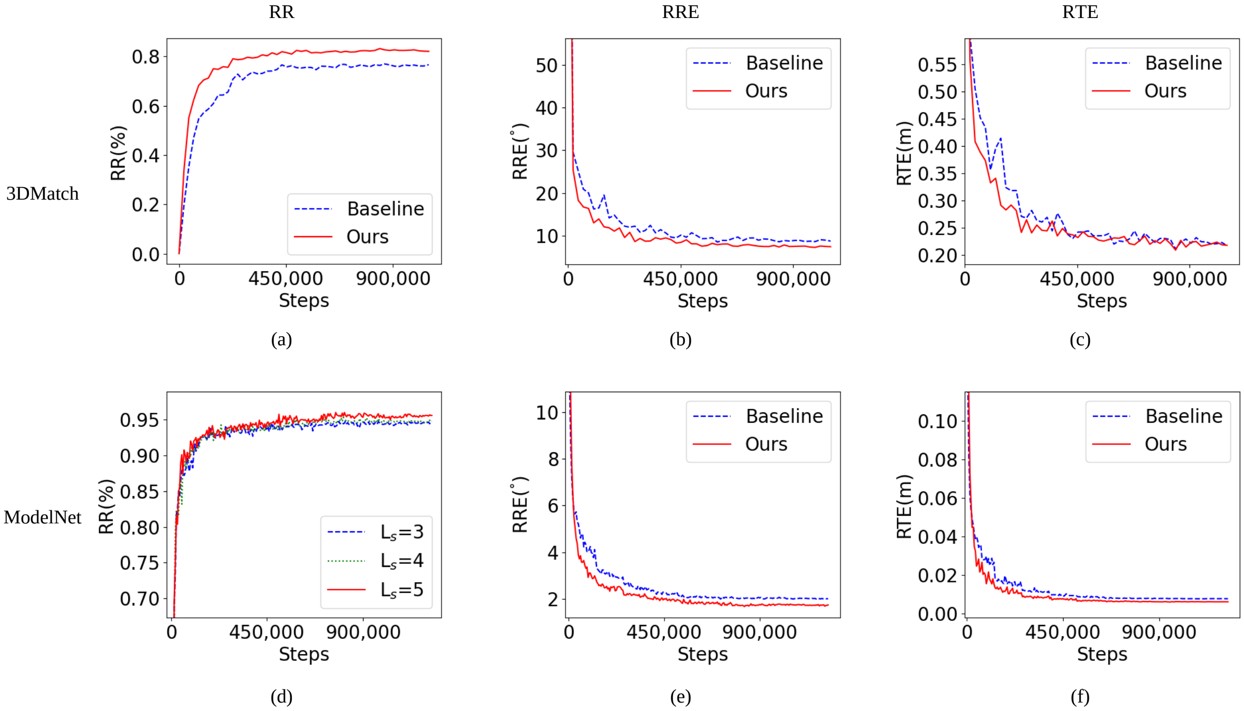
Table 1. The registration performance on 3DMatch and 3DLoMatch datasets.

Model	3DMatch			3DLoMatch		
	RR (%)↑	RRE (°)↓	RTE (m)↓	RR (%)↑	RRE (°)↓	RTE (m)↓
3DSN [29]	78.4	2.199	0.071	33.0	3.528	0.103
FCGF [3]	85.1	1.949	0.066	40.1	3.147	0.100
D3Feat [28]	81.6	2.161	0.067	37.2	3.361	0.103
Predator-5k [17]	89.0	2.029	0.064	59.8	3.048	0.093
Predator-1k [17]	90.5	2.062	0.068	62.5	3.159	0.096
Predator-NR [17]	62.7	2.582	0.075	24.0	5.886	0.148
OMNet [30]	35.9	4.166	0.105	8.4	7.299	0.151
DGR [31]	85.3	2.103	0.067	48.7	3.954	0.113
PCAM [32]	85.5	1.808	0.059	54.9	3.529	0.099
RegTR [10]	92.0	1.567	0.049	64.8	2.827	0.077
HR-Net [12]	93.1	**1.424**	0.044	67.6	2.513	0.073
Ours	**93.8**	1.448	**0.043**	**73.0**	**2.271**	**0.065**

Note: ↑ represents the higher the better, ↓ indicates the lower the better, and bold font represents the best.

In Table 1, the methods above the line are based on RANSAC, and those below are non-RANSAC methods. Our method notably enhances point cloud registration performance within limited overlapping regions, with an increase of over 7 percentage points and lower registration errors. Moreover, it also demonstrates better registration performance for highly overlapping point clouds.

To showcase the exceptional capability of our suggested model, we illustrate the validation set's test performance curves throughout the training phase, as displayed in Figure 7. The initial row illustrates the test curves for the 3DMatch dataset, whereas the subsequent row exhibits the test curves for ModelNet. These graphs depict the RR, RRE, and RTE. Our model converges rapidly, attains superior registration recall rates, and demonstrates reduced matching errors.



(a) (b) (c)

(d) (e) (f)

Figure 7. The evaluation curves during the training process for 3DMatch (**a**–**c**) and ModelNet (**d**–**f**).

In Figure 8, we present the point cloud registration capability on the 3DLoMatch dataset, focusing on areas with small degrees of overlap and high structural similarity outside the overlapping regions. It can be observed that our method generates more feature correspondences, predominantly within the overlap regions, while the baseline method produces more correspondences within the overlap regions but also includes some erroneous matches outside these regions, significantly impacting registration capability.

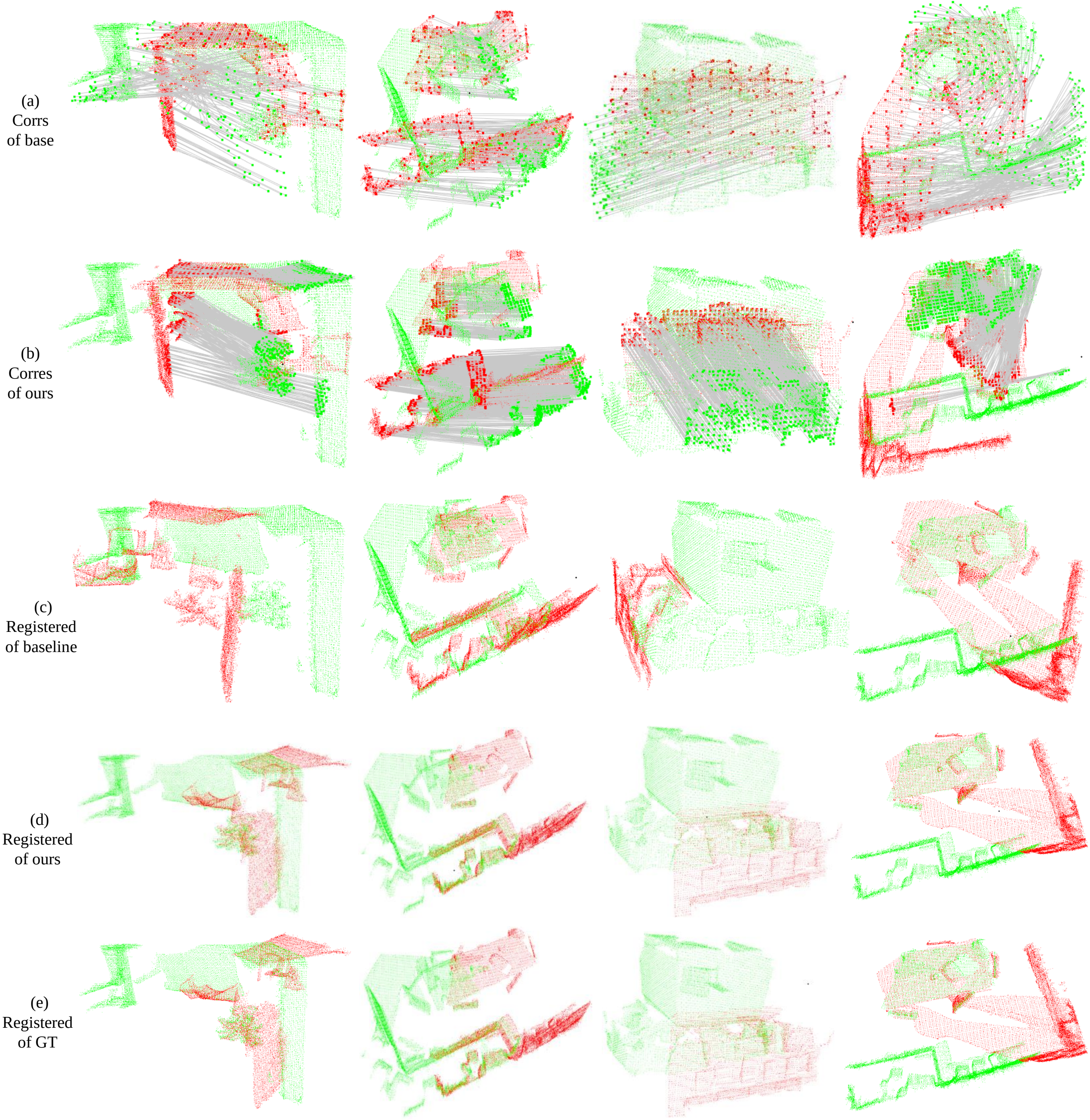


Figure 8. The performance of our method on 3DLoMath. Each column corresponds to different pairs of point clouds. The red and green points signify point clouds $\mathbf{P}_0$ and $\mathbf{Q}_0$. Row (**a**) shows the superpoint correspondences obtained by the baseline, row (**b**) displays the dense point correspondences computed by our method, row (**c**) illustrates the registration of the baseline, row (**d**) depicts the registration of our method, and row (**e**) showcases the registration using ground truth poses.

3.3.2. Evaluation of ModelNet and ModelLoNet

In the case of the ModelNet and ModelLoNet benchmarks, we refer to the relevant method [10,12] to evaluate point cloud registration error using the RRE, RTE, and Chamfer distance (CD). Since RR is a key metric for assessing the success of point cloud registration, we further assess the performance of our method in terms of its RR.

Similarly, we provide detailed demonstrations of the registration performance on ModelNet and ModelLoNet in Table 2 and Figures 7 and 9. The experimental results show that the proposed TTReg not only accomplishes strong performance in registration in real-world scenarios but also achieves significant improvements on synthesized datasets. The dense matching points computed by our method are mainly concentrated within the overlap regions, effectively enhancing the registration performance.

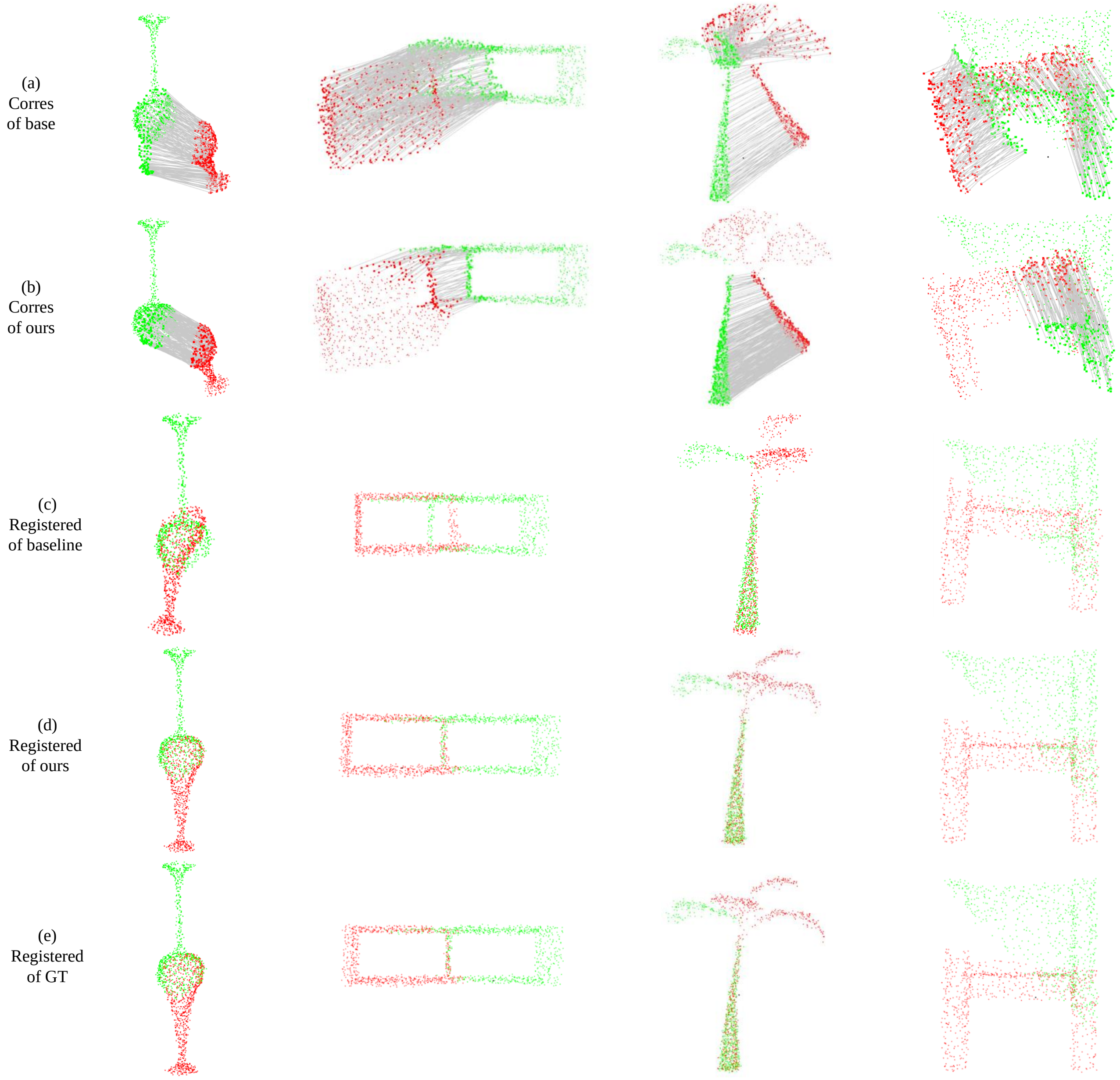


Figure 9. The performance of our method on ModelLoNet. Columns correspond to different point cloud pairs. The red and green points signify point cloud $\mathbf{P}_0$ and $\mathbf{Q}_0$. Row (**a**) shows the superpoint correspondences obtained by the baseline method, row (**b**) displays the dense point correspondences computed by our method, row (**c**) illustrates the registration of the baseline, row (**d**) depicts the registration of our method, and row (**e**) showcases the registration using ground truth poses.

Table 2. The registration performance on ModelNet and ModelLoNet datasets.

Model	**ModelNet**				**ModelLoNet**			
	RR (%)↑	**RRE (°)↓**	**RTE (m)↓**	**CD (m)↓**	**RR (%)↑**	**RRE (°)↓**	**RTE (m)↓**	**CD (m)↓**
PointNetLK [33]	-	29.725	0.297	0.02350	-	48.567	0.507	0.0367
OMNet [30]	-	2.9470	0.032	0.00150	-	6.5170	0.129	0.0074
DCP-v2 [34]	-	11.975	0.171	0.01170	-	16.501	0.300	0.0268
RPM-Net [27]	-	1.7120	0.018	0.00085	-	7.3420	0.124	0.0050
Predator [17]	-	1.7390	0.019	0.00089	-	5.2350	0.132	0.0083
RegTR [10]	96.29 *	1.4730	0.014	0.00078	68.17 *	3.9300	0.087	0.0037
HR-Net [12]	**97.71 ***	**1.1970**	**0.011**	**0.00072**	**74.33 ***	**3.5710**	**0.078**	**0.0034**
Ours	97.24	1.3538	**0.011**	0.00078	72.35	3.9580	0.086	0.0039

Note: ↑ represents the higher the better, ↓ indicates the lower the better. - indicates that the original paper does not provide these data. * represents the results we reproduce, and bold font represents the best.

3.4. Ablation

To corroborate the efficacy of our TTReg, we evaluate the impact of L_s repetitions of the proposed RIG-Transformer on the 3DMatch and ModelNet; the low-overlap 3DLoMatch and ModelLoNet benchmarks will also be evaluated. Following prior works [10,17,27], we assess the RR, RRE, and RTE for 3DMatch and 3DLoMatch, while for the ModelNet and ModelLoNet datasets, we evaluate the CD, RRE, and RTE. The quantitative performance metrics for 3DMatch and 3DLoMatch are presented in Table 3, while those for ModelNet and ModelLoNet are shown in Table 4.

We consider values of $L_s = 3, 4, 5$, with the maximum value limited to 5 due to computational constraints. From Tables 3 and 4, we observed that increasing L_s appropriately leads to improved matching performance, with optimal results achieved at $L_s = 5$. Further increasing L_s may offer additional improvements. However, due to computational limitations, we do not test cases where $L_s > 5$. Notably, our method incurs significantly lower costs when increasing L_s compared to previous methods, as we only match the dense points with the highest correspondence in the overlapping regions, greatly reducing the computational resources required.

Table 3. The ablation performance on 3DMatch and 3DLoMatch datasets.

Model	**3DMatch**			**3DLoMatch**		
	RR (%)↑	**RRE (°)↓**	**RTE (m)↓**	**RR (%)↑**	**RRE (°)↓**	**RTE (m)↓**
Baseline [10]	92.0	1.567	0.049	64.8	2.827	0.077
$L_s = 3$	92.2	1.494	0.044	67.5	2.289	0.070
$L_s = 4$	**93.8**	1.516	0.045	71.4	**2.212**	0.068
$L_s = 5$	**93.8**	**1.448**	**0.043**	**73.0**	2.271	**0.065**

Note: ↑ represents the higher the better, ↓ indicates the lower the better, and bold font represents the best.

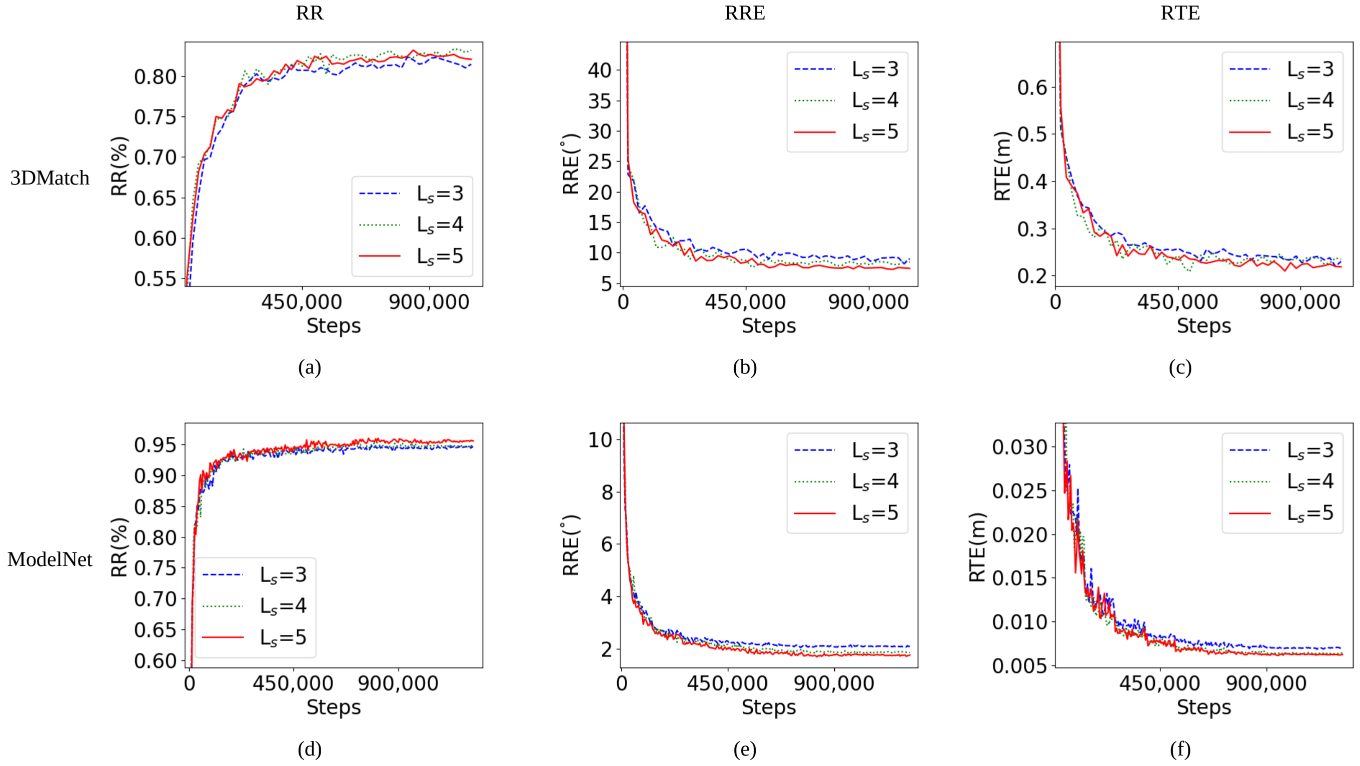
Table 4. The ablation performance on ModelNet and ModelLoNet datasets.

Model	**ModelNet**				**ModelLoNet**			
	RR (%)↑	**RRE (°)↓**	**RTE (m)↓**	**CD (m)↓**	**RR (%)↑**	**RRE (°)↓**	**RTE (m)↓**	**CD (m)↓**
Baseline [10]	96.29 *	1.4730	0.014	**0.00078**	68.17 *	**3.9300**	0.087	**0.0037**
$L_s = 3$	96.05	1.8128	0.015	0.00086	70.14	4.5655	0.089	0.0038
$L_s = 4$	97.08	1.5521	0.013	0.00083	70.77	4.2219	**0.086**	0.0038
$L_s = 5$	**97.24**	**1.3538**	**0.011**	**0.00078**	**72.35**	3.9580	**0.086**	0.0039

Note: ↑ represents the higher the better, ↓ indicates the lower the better. * represents the results we reproduce, and bold font represents the best.

4. Discussion

To further investigate the impact of L_s repetition times of RIG-Transformer during the training process on registration performance, we visualize the evaluation curves for 3DMatch and ModelNet (see Figure 10). The RR improves with increasing L_s, while the RRE and RTE decrease as L_s increases. The evaluation curves during the training process align with the registration performance presented in Tables 3 and 4.



(a) (b) (c)

(d) (e) (f)

Figure 10. The impact of RIG-Transformer layer L_s on registration performance during the training process for 3DMatch (**a**–**c**) and ModelNet (**d**–**f**).

Furthermore, we analyze the distribution of dense points in the overlapping regions predicted by the RIG-Transformer module. The corresponding dense points are illustrated in Figures 11 and 12. We first compute the sparse matching keypoints predicted by the baseline [10] and the dense corresponding points obtained by our RIG-Transformer module. Then, we align the point clouds $\mathbf{P}_0$ and $\mathbf{Q}_0$ using the ground truth point cloud relative pose transformation. The gray connecting lines in the figures link the predicted matching corresponding points. In point cloud pairs with high common ratios, sparse corresponding keypoints are predominantly located in the overlapping regions. Conversely, for tow point cloud with low common ratios, numerous unmatching keypoints appear within non-overlapping areas. On the other hand, in point cloud pairs with low common area ratios between point cloud pairs, numerous unmatched keypoints appear in the non-overlapping areas.

Our model predicts dense points that are primarily clustered within the overlapping regions, particularly in regions with low overlap ratios. This outcome is credited to our model's enhanced capacity to thoroughly investigate the structural characteristics of point cloud sets, leading to improved registration performance by directly predicting the relative pose transformation of point clouds.

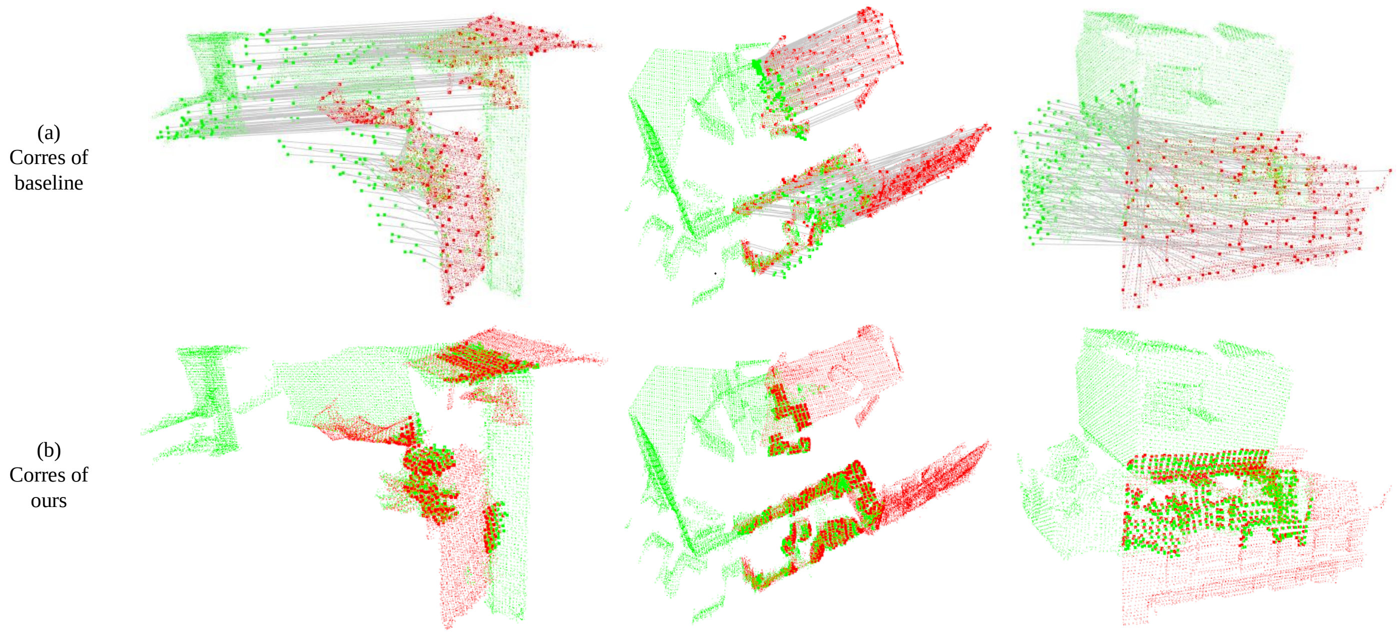


Figure 11. Predicted 3DLoMatch overlap area. Points in red and green represent point clouds $\mathbf{P}_0$ and $\mathbf{Q}_0$, respectively; gray lines represent the connection relationship between corresponding points. The first row (**a**) shows the correspondence of sparse matching keypoints from the baseline, and the second row (**b**) displays the correspondence of dense points predicted by our model located in the overlapping area, with each row representing a pair of point clouds to be matched.

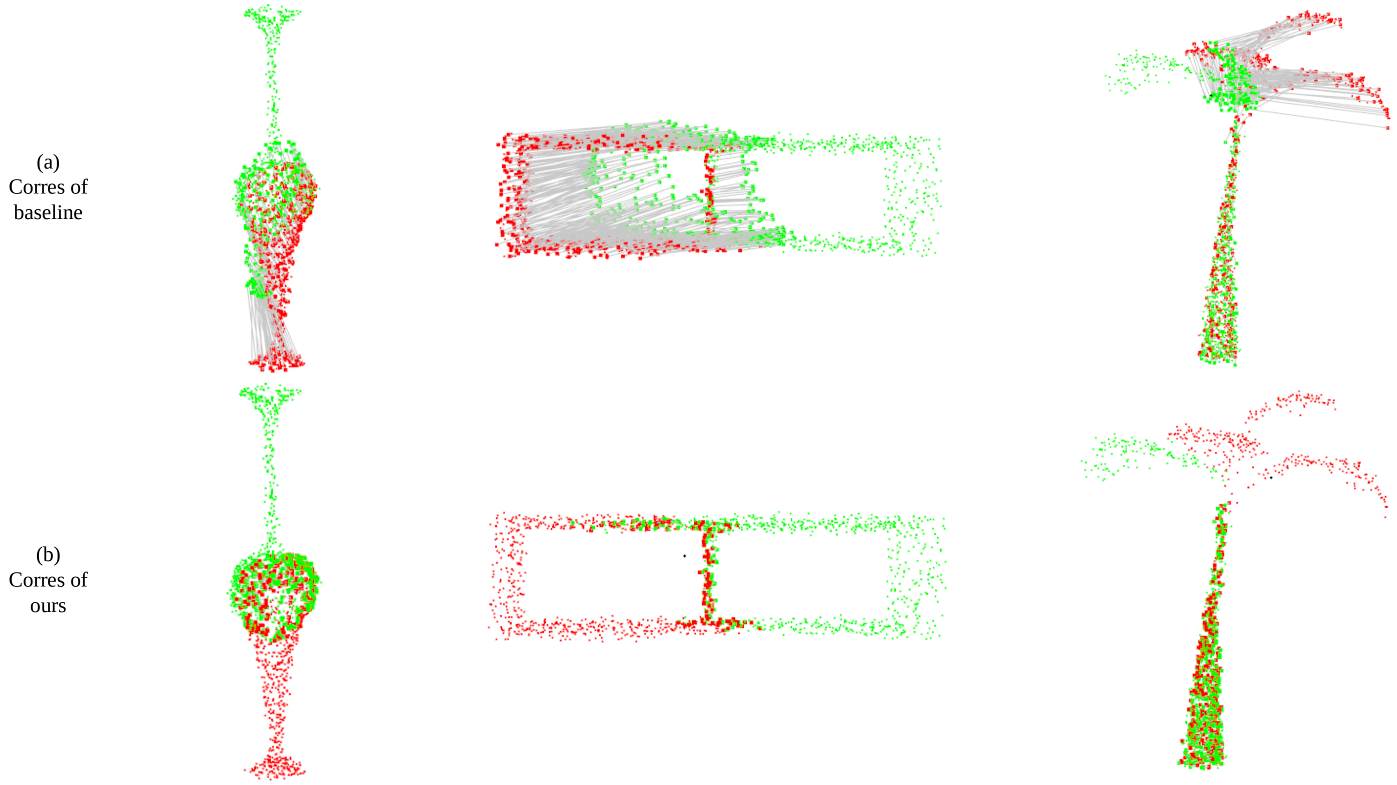


Figure 12. Predicted ModelLoNet overlap area, where points in red represent point cloud $\mathbf{P}_0$, points in green represent point cloud $\mathbf{Q}_0$, and gray lines represent the relationship between corresponding points. The first row (**a**) shows the correspondence of sparse matching keypoints from the baseline, and the second row (**b**) displays the correspondence of dense points predicted by our model located in the overlapping area, with each row representing a pair of point clouds to be matched.

5. Conclusions

Our proposed method calculates the relative transformation of point clouds by regressing the corresponding point coordinates of dense point clouds using two cascaded Transformers. We divide the point cloud registration into two steps. Firstly, we divide the points of pairs of point clouds into overlapping and non-overlapping regions. The proposed RIG-Transformer is used to distinguish the best-matching sparse superpoints located in the overlapping region, which transforms the point-to-point matching into a binary classification, reducing the difficulty of classification. The proposed RIG-Transformer integrates point cloud geometric features and positional encoding, possessing rotational invariance. By extracting more complex geometric features, and improving the robustness of feature matching, RIG-Transformer can effectively filter out incorrect superpoint correspondences with high structural similarity outside the overlapping area. Subsequently, the dense point clouds are indexed through the spatial clustering relationship between point cloud superpoints and dense point clouds. The dense point clouds located in the overlapping region play a key role in point cloud registration and have high spatial coordinate accuracy. By using the Transformer Cross-Encoder, corresponding point coordinates can be regressed with higher precision, thereby enhancing the estimated transformation accuracy of point clouds. By combining RIG-Transformer with a Transformer Cross-Encoder, we directly regress the transformation between dense points within the overlapping region. Our approach leverages both the geometric properties of features and the precision of the point coordinates in dense point clouds. Importantly, our regression mechanism avoids the time overhead incurred by using RANSAC. However, due to constraints on computational resources, we did not conduct extensive testing on the interaction times of RIG-Transformer.

Author Contributions: Conceptualization, Y.Z.; methodology, Y.Z.; software, Y.Z.; validation, Y.Z.; formal analysis, Y.Z.; investigation, Y.Z.; resources, Y.Z.; data curation, Y.Z.; writing—original draft preparation, Y.Z.; writing—review and editing, Y.Z., L.C., Q.Z., J.Z., H.W. and M.R.; visualization, Y.Z.; supervision, Y.Z. and M.R.; project administration, Y.Z. and M.R.; funding acquisition, M.R. All authors have read and agreed to the published version of the manuscript.

Funding: This research was supported in part by the National Natural Science Foundation of China under Grant 61703209 and Grant 62272231, in part by the Natural Science Foundation of the Higher Education Institutions of Jiangsu Province (No. 19KJB520022), in part by the Qing Lan Project of Jiangsu Province (2021), and in part by the Cultivation Object of Major Scientific Research Project of CZIMT (No. 2019ZDXM06). The APC was funded by Nanjing University of Science and Technology.

Data Availability Statement: All datasets used in this study are publicly available. The 3DMatch dataset is available at https://share.phys.ethz.ch/~gsg/pairwise_reg/3dmatch.zip, accessed on 15 August 2023, and the ModelNet dataset is available at https://shapenet.cs.stanford.edu/media/modelnet40_ply_hdf5_2048.zip, accessed on 15 August 2023).

Conflicts of Interest: The authors declare no conflicts of interest.

Abbreviations

The following abbreviations are used in this manuscript:

RIG-Transformer	Rotation-Invariant Geometric Transformer Cross-Encoder
TTReg	Transformer-to-Transformer Regression
RANSAC	Random Sample Consensus
FPS	Farthest Point Sampling
RR	Registration Recall
RRE	Relative Rotation Error
RTE	Relative Translation Error
CD	Chamfer Distance

References

1. Chen, Y.; Mei, Y.; Yu, B.; Xu, W.; Wu, Y.; Zhang, D.; Yan, X. A robust multi-local to global with outlier filtering for point cloud registration. *Remote Sens.* **2023**, *15*, 5641. [CrossRef]
2. Sumetheeprasit, B.; Rosales Martinez, R.; Paul, H.; Shimonomura, K. Long-range 3D reconstruction based on flexible configuration stereo vision using multiple aerial robots. *Remote Sens.* **2024**, *16*, 234. [CrossRef]
3. Choy, C.; Park, J.; Koltun, V. Fully convolutional geometric features. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Seoul, Republic of Korea, 27 October–2 November 2019; pp. 8958–8966.
4. Han, T.; Zhang, R.; Kan, J.; Dong, R.; Zhao, X.; Yao, S. A point cloud registration framework with color information integration. *Remote Sens.* **2024**, *16*, 743. [CrossRef]
5. Mei, G.; Tang, H.; Huang, X.; Wang, W.; Liu, J.; Zhang, J.; Van Gool, L.; Wu, Q. Unsupervised deep probabilistic approach for partial point cloud registration. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Vancouver, BC, Canada, 17–24 June 2023; pp. 13611–13620.
6. Cuturi, M. Sinkhorn distances: Lightspeed computation of optimal transport. *Adv. Neural Inf. Process. Syst.* **2013**, *26*, 2292–2300.
7. Qin, Z.; Yu, H.; Wang, C.; Guo, Y.; Peng, Y.; Xu, K. Geometric transformer for fast and robust point cloud registration. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, New Orleans, LA, USA, 18–24 June 2022; pp. 11143–11152.
8. Fischler, M.A.; Bolles, R.C. Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Commun. ACM* **1981**, *24*, 381–395. [CrossRef]
9. Yu, H.; Qin, Z.; Hou, J.; Saleh, M.; Li, D.; Busam, B.; Ilic, S. Rotation-invariant transformer for point cloud matching. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Vancouver, BC, Canada, 17–24 June 2023; pp. 5384–5393.
10. Yew, Z.J.; Lee, G.H. Regtr: End-to-end point cloud correspondences with transformers. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, New Orleans, LA, USA, 18–24 June 2022; pp. 6677–6686.
11. Wu, Y.; Zhang, Y.; Ma, W.; Gong, M.; Fan, X.; Zhang, M.; Qin, A.; Miao, Q. Rornet: Partial-to-partial registration network with reliable overlapping representations. *IEEE Trans. Neural Netw. Learn. Syst.* **2023**. [CrossRef] [PubMed]
12. Zhao, Y.; Chen, L.; Hu, B.; Wang, H.; Ren, M. HR-Net: Point cloud registration with hierarchical coarse-to-fine regression network. *Comput. Electr. Eng.* **2024**, *113*, 109056. [CrossRef]
13. Wang, H.; Liu, Y.; Hu, Q.; Wang, B.; Chen, J.; Dong, Z.; Guo, Y.; Wang, W.; Yang, B. Roreg: Pairwise point cloud registration with oriented descriptors and local rotations. *IEEE Trans. Pattern Anal. Mach. Intell.* **2023**, *45*, 10376–10393. [CrossRef] [PubMed]
14. Thomas, H.; Qi, C.R.; Deschaud, J.E.; Marcotegui, B.; Goulette, F.; Guibas, L.J. Kpconv: Flexible and deformable convolution for point clouds. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Seoul, Republic of Korea, 27 October–2 November 2019; pp. 6411–6420.
15. Qi, C.R.; Su, H.; Mo, K.; Guibas, L.J. Pointnet: Deep learning on point sets for 3D classification and segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 652–660.
16. Arya, S.; Mount, D.M.; Netanyahu, N.S.; Silverman, R.; Wu, A.Y. ANN: A library for approximate nearest neighbor searching. *ACM Trans. Math. Softw. (TOMS)* **1999**, *26*, 469–483.
17. Huang, S.; Gojcic, Z.; Usvyatsov, M.; Wieser, A.; Schindler, K. Predator: Registration of 3d point clouds with low overlap. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Nashville, TN, USA, 19–25 June 2021; pp. 4267–4276.
18. Li, J.; Chen, B.M.; Lee, G.H. SO-Net: Self-organizing network for point cloud analysis. In Proceedings of the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 9397–9406.
19. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, Ł.; Polosukhin, I. Attention is all you need. *Adv. Neural Inf. Process. Syst.* **2017**, *30*.
20. Rocco, I.; Cimpoi, M.; Arandjelović, R.; Torii, A.; Pajdla, T.; Sivic, J. Neighbourhood consensus networks. *Adv. Neural Inf. Process. Syst.* **2018**, *31*, 1651–1662.
21. Sun, J.; Shen, Z.; Wang, Y.; Bao, H.; Zhou, X. LoFTR: Detector-free local feature matching with transformers. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Nashville, TN, USA, 19–25 June 2021; pp. 8922–8931.
22. Kabsch, W. A solution for the best rotation to relate two sets of vectors. *Acta Crystallogr. Sect. A Cryst. Phys. Diffr. Theor. Gen. Crystallogr.* **1976**, *32*, 922–923. [CrossRef]
23. Umeyama, S. Least-squares estimation of transformation parameters between two point patterns. *IEEE Trans. Pattern Anal. Mach. Intell.* **1991**, *13*, 376–380. [CrossRef]
24. 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 Proceedings of the IEEE/CVF International Conference on Computer Vision, Montreal, BC, Canada, 11–17 October 2021; pp. 16014–16023.
25. Zeng, A.; Song, S.; Nießner, M.; Fisher, M.; Xiao, J.; Funkhouser, T. 3dmatch: Learning local geometric descriptors from rgb-d reconstructions. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 1802–1811.

26. Wu, Z.; Song, S.; Khosla, A.; Yu, F.; Zhang, L.; Tang, X.; Xiao, J. 3D shapenets: A deep representation for volumetric shapes. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Boston, MA, USA, 7–12 June 2015; pp. 1912–1920.
27. Yew, Z.J.; Lee, G.H. Rpm-net: Robust point matching using learned features. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 13–19 June 2020; pp. 11824–11833.
28. Bai, X.; Luo, Z.; Zhou, L.; Fu, H.; Quan, L.; Tai, C.L. D3feat: Joint learning of dense detection and description of 3d local features. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 13–19 June 2020; pp. 6359–6367.
29. Gojcic, Z.; Zhou, C.; Wegner, J.D.; Wieser, A. The perfect match: 3d point cloud matching with smoothed densities. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019; pp. 5545–5554.
30. Xu, H.; Liu, S.; Wang, G.; Liu, G.; Zeng, B. Omnet: Learning overlapping mask for partial-to-partial point cloud registration. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Nashville, TN, USA, 19–25 June 2021; pp. 3132–3141.
31. Choy, C.; Dong, W.; Koltun, V. Deep global registration. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 13–19 June 2020; pp. 2514–2523.
32. Cao, A.Q.; Puy, G.; Boulch, A.; Marlet, R. PCAM: Product of cross-attention matrices for rigid registration of point clouds. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Nashville, TN, USA, 19–25 June 2021; pp. 13229–13238.
33. Aoki, Y.; Goforth, H.; Srivatsan, R.A.; Lucey, S. Pointnetlk: Robust & efficient point cloud registration using pointnet. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019; pp. 7163–7172.
34. Wang, Y.; Solomon, J.M. Deep closest point: Learning representations for point cloud registration. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Seoul, Republic of Korea, 27 October–2 November 2019; pp. 3523–3532.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.