



A dynamic learning framework integrating attention mechanism for point cloud registration

Cuixia Li¹ · Yuyin Guan¹ · Shanshan Yang¹ · Yinghao Li¹

Accepted: 19 September 2023 / Published online: 20 October 2023

© The Author(s), under exclusive licence to Springer-Verlag GmbH Germany, part of Springer Nature 2023

Abstract

To improve the low accuracy problem of existing point cloud registration algorithms attributed to deficient point cloud geometric features, we proposed a new point cloud registration network inspired by dynamic feature extraction and the graph attention mechanism. The model uses the dynamic graph edge convolution neural network to characterize the multi-level semantics of the point cloud at first, then uses a feature fusion module based on attention mechanism to fuse the representation information, and finally uses the singular value decomposition (SVD) method to generate the transformation matrix. The experimental verification was carried out on the ModelNet40, ShapeNet Part datasets, and the local industrial part dataset. Experiment results show that our model gets competitive registration performance compared with other advanced models on three datasets. When tested on the untrained data class and the noisy circumstances, our model gets lower average registration errors than compared models. It shows that our framework has not only the characteristics of high registration accuracy and generalization ability but also strong robustness.

Keywords Point cloud registration · Dynamic feature extraction · Edge convolution · Graph attention network · Offset-attention

1 Introduction

Due to the application and popularization of high-precision 3D sensors in recent years, the point cloud data have become indispensable and widely used in numerous engineering practices and research. Especially deep learning has shown amazing excellent capabilities in many fields including but not limited to computer vision [1, 2], medical image analysis [3], and unsupervised learning [4]. Therefore, researchers began to focus on using various deep learning methods to solve the problem of point cloud in 3D object detection [5–8], quality monitoring [9], path planning of robots [10], point cloud registration [11], and other aspects.

In the field of point cloud registration that we focus on, the impact of surroundings, equipment, or angle of sampling caused the generated point cloud data are various even from the same scene, so how to register two-point clouds quickly and accurately has become complex and important research. The point cloud registration (PCR) is to estimate the mapping

between the source point cloud X and the target point cloud Y , including translation, rotation, stretch, affine, transmission, polynomial, and other transformed forms. We studied here mainly for translation and rotation problems in rigid registration from the same-source point clouds. Currently, there are two implementations of point cloud registration [12]: optimization-based methods and depth-based methods.

The optimization-based method is a process of gradually optimizing registration accuracy by iterating correspondence searching and transformation estimation adopting mathematical theory. Correspondence searching is a process to find the corresponding relation, and transformation estimation uses the corresponding relation to calculate the transformation matrix. These methods are mainly represented by iterative closest point (ICP) [13], normal distributions transform (NDT) [14], and 4-points congruent sets (4PCS) [15]. In addition, there are some methods designed with artificial coding features to improve the correspondence searching, such as point feature histograms (PFH) [16], fast point feature histograms (FPFH) [17], SHOT [18], the spin image proposed by Johnson [19], the 3D and harmonic shape contexts proposed by Frome [20], etc. They all implemented shape descriptors in various ways and made better improvements

✉ Yinghao Li
yinghaoli@zzu.edu.cn

¹ School of Cyber Science and Engineering, Zhengzhou University, Zhengzhou 450002, China

in robustness. However, most of the optimization-based registration doesn't perform satisfactorily facing the effects of noise, outliers, and low overlap, which cannot be avoided when sampling a point cloud. Not only that, but sensitivity to the initial position of the point cloud is also one of their disadvantages [12].

In addition to the optimization-based method, another method widely studied by scholars is the depth-based registration method. Due to the excellent intelligence, wide coverage and data-driven advantages of deep neural networks, such methods provide better accuracy, robustness, and generalization for point cloud registration tasks. According to the different functions and outputs of the neural network, two registration methods in depth-based methods have been divided [12]: end-to-end learning-based and feature learning-based registration methods.

The end-to-end learning-based registration methods estimate the mapping directly through an end-to-end framework, whose input is two-point clouds and the output is a transformation matrix. That is, the transformation estimation is embedded into the neural network optimization. It integrates feature extraction, correspondence, and transformation. One idea of end-to-end learning-based methods is to treat registration as a regression problem and fit it into a regression model for the transformation matrix estimation [21–23]. Besides, some methods combine conventional registration-related optimization theories with deep neural networks, like the maximum likelihood estimate (MLE) and Gaussian mixture model utilized in DeepGMR [24], and the minkowski proposed in DGR [25]. In addition, focus on the feature extraction provides feasible schemes, like rotation-invariant (RI) in DWC [26], two shape tensors proposed in PR-Net (Wang L et al. [27]), and two-point clouds alignment algorithms in PCRNet [28]. Generally, the end-to-end learning-based registration methods could leverage both the merits of mathematical theories and deep neural networks, also its neural network could be designed and optimized for registration tasks specifically. But an end-to-end framework is a black box that includes correspondence searching and transformation estimation, causing the network model to be sensitive to different environmental data [12].

The feature learning-based registration methods use deep neural networks for feature extraction and estimate accurate correspondences before optimization, and then, using a one-step optimization, like Singular Value Decomposition (SVD) algorithm, to determine the final transformation matrix. It provides robust and accurate correspondence searching because of the deep learning-based point feature, and the one-step estimation leads to more accurate registration results through accurate correspondences. And recent research has made some improvements. To solve the low correspondence between point clouds caused by insufficient semantic information, which further affects the registration effect, the

existing methods hope to obtain as many feature maps as possible from multiple dimensions, to improve the ability of the network to complete the determination of correspondence. The deep closest point (DCP) proposed by Wang et al. [29] estimates correspondences with depth features based on dynamically updating the graph structure [30] between layers combined with an attention module while maintaining the permutation of points, and it uses SVD computational transformation finally for the registration, which performs robustness to noise. Wang and Sun et al. propose the PRNet [31] to improve the DCP model, which utilizes global pooling to aggregate point-by-point features to obtain global features, then predicts annealing parameters through a subnetwork to control the sharpening degree of matching. RPM-Net [32] used the differentiable Sinkhorn layer and annealing algorithm to obtain the matching relation between point pairs by learning and integrating spatial features and local geometric information. Furthermore, IDAM [33] proposed an iterative distance-aware similarity matrix convolution network, which can be easily integrated with traditional features (such as FPFH [17]) or learning-based features to achieve registration. When the high-dimensional semantic information of each point is of the same importance in the point cloud, the network cannot accurately distinguish whether there are irrelevant points such as noise, so the network cannot show better performance in complex point cloud registration tasks. Many existing approaches try to overcome this problem by using attention mechanisms, but we find that the performance of attention mechanisms is spotty when processing feature maps obtained in different ways and dimensions.

To address the problem of feature richness and change the influence factor of feature vectors, a dynamic learning framework integrating the attention mechanism is proposed in this paper inspired by the ideas of dynamic feature fusion and attention mechanism. In our framework, the features of a point cloud could be extracted and fused dynamically by the multi-layer of EdgeConv in series. This method can effectively extract features from multiple dimensions to enrich semantic information. And we import a variant attention module of self-attention to measure the importance of feature maps; it integrates contextual information and enhances the features' representation. Our framework has been tested on ShapeNet Part and ModelNet40 datasets extensively and compared with other advanced networks. The experimental results prove that our framework takes advantage of robustness and generalization, and it has higher registration accuracy.

This paper extends a preliminary version of this work presented at the 2022 2nd International Conference on Consumer Electronics and Computer Engineering (ICCECE). Compared with the conference version, this paper provides some new additions. First of all, more theoretical additions about the edge convolution and offset-attention module are

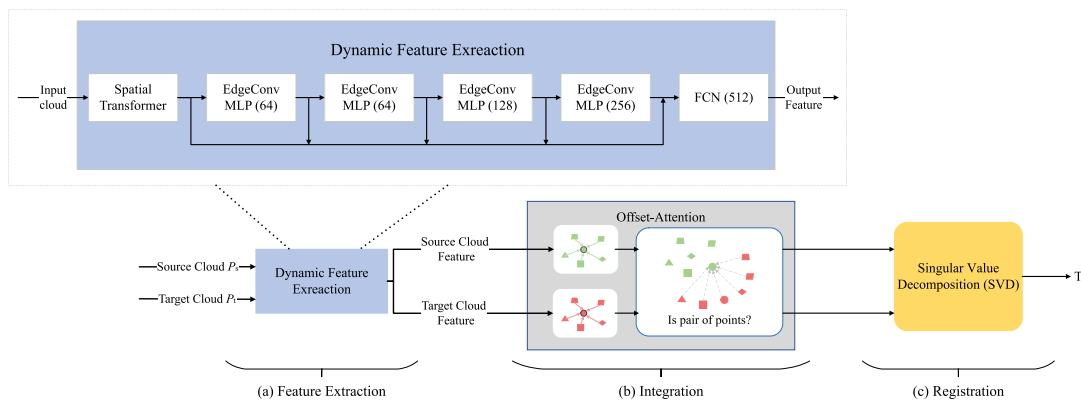


Fig. 1 The framework structure of this paper's network

provided. It is convenient for readers to understand and reproduce the ideas in this paper by means of mathematical modeling. Secondly, new contrast and ablation experiments are shown to fairly and objectively demonstrate the excellent results achieved by our framework. In addition, an analysis of the effect of our implementation on field regularity and runtime is added in this paper.

2 Method

The registration process of our framework is mainly divided into three parts: feature extraction, integration, and registration (as shown in Fig. 1). The source point cloud and the target point cloud are fed into the same feature extraction network, respectively, to embed them to high-dimensional, then rescreen the feature matrixes and obtain the correlation information between them using the attention module [34], and finally estimate transformation matrixes using SVD.

2.1 Feature extraction

When there is a large distribution difference between the unknown scene and the training data, the registration performance will drop sharply, which limits the generalization ability of the network [35]. We found that the accuracy of the corresponding relation and generalization can be effectively improved by fusing the global and local spatial features of point clouds. This cannot only be evidenced by the aforementioned DCP and RPM-Net, but also by Dynamic Graph CNN (DGCNN) [30]. DGCNN proposes to construct the K-nearest neighbor graph of points and uses the edge convolution (EdgeConv) module to capture the edges connecting pairs of points.

At the extraction stage, the point cloud data are embedded in a high-dimensional space and transforms local features and global features using multiple stacked EdgeConv modules [30]. The EdgeConv module performs convolution-like

operations on edges connecting neighboring pairs of points using local neighborhood graphs. We use the EdgeConv module in feature extraction so that the local feature information can be extracted without changing the number of features and preserving the feature information to the maximum extent. This cannot only improve the network's ability to extract local features and enrich feature information, but also take into account the relationships between pairs of points, which can more effectively characterize local features.

The directed graph is constructed dynamically in front of each EdgeConv module to enrich the expression of feature information. Our feature extraction exploits a Spatial Transformer and multi-layer edge convolutional network EdgeConv cascade (as shown in a dashed box in Fig. 1). The Spatial Transformer aligns all points to a unified point set space for learning rotation invariance. And the key lies in the multi-layer EdgeConv module which adopts dynamic feature extraction.

When calculating the features at each EdgeConv module, it is divided into three steps: constructing the k-nearest neighbor (k-NN) graph, calculating the edge feature, and the channel-wise symmetric aggregation operation (Fig. 2). The point cloud feature is described by a point set with n points, denoted as $P = \{p_1, p_2, \dots, p_n\} \in \mathbb{R}^F$, where F represents the feature dimension and $p_i = (x_{i1}, x_{i2}, \dots, x_{iF})$, $i = 1, 2, \dots, F$. Before the feature matrix is encoded into EdgeConv, the k-NN digraph $G = (V, E)$ should be constructed, where $V = \{1, 2, \dots, n\}$ and $E \subseteq V \times V$ represent the set of vertices and edges, respectively. Then, calculating edge feature by $e_{ij} = h_{\Theta}(p_i, p_j)$, where $h_{\Theta} : \mathbb{R}^F \times \mathbb{R}^F \rightarrow \mathbb{R}^{F'}$ is a nonlinear function containing a set of learning parameters Θ . The edge feature is a local feature expression about the edges of p_i connecting neighboring pairs of points. Finally, we use a channel-wise symmetric aggregation operation Ω (e.g., \sum or \max) to represent p_i with a new feature p'_i :

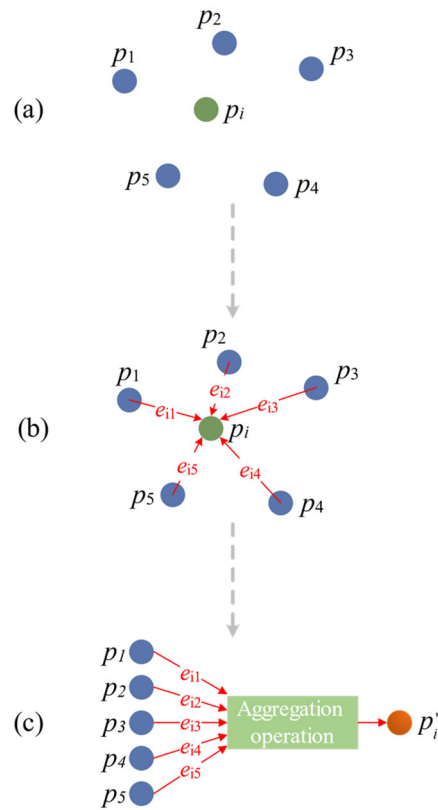


Fig. 2 **a** k-NN points of p_i ; **b** Edge feature; **c** Aggregation operation

$$p'_i = \Omega_{j:(i,j) \in E} \bar{h}_{\Theta}(p_j - p_i, p_i) \quad (1)$$

Here, each new feature p'_i corresponds to p_i to ensure that reduces the loss of feature information. $\bar{h}_{\Theta}(p_j - p_i, p_i)$ is an asymmetric function to select edge features; it is the way of fusion operation and was designed to have an impact on the performance of EdgeConv.

This equation \bar{h}_{Θ} takes into account both the local feature $p_j - p_i$ containing the nearest neighbor points and the global feature p_i . In M -layer filters, $\Theta_m = (\theta_1, \theta_2, \dots, \theta_M, \phi_1, \phi_2, \dots, \phi_M)$ is the encoding weight, and the nonlinear function $h_{\Theta} : \mathbb{R}^F \times \mathbb{R}^F \rightarrow \mathbb{R}^{F'}$ can be noted as follows:

$$e'_{ijm} = \text{ReLU}(\theta_m \cdot (p_j - p_i) + \phi_m \cdot p_i) \quad (2)$$

θ_m and ϕ_m have the same dimension as p , and \cdot is the Euclidean inner product. Eventually, the aggregation operation is implemented by the symmetric function \max as follows:

$$p'_{im} = \text{Max}_{j:(i,j) \in E} e'_{ijm} \quad (3)$$

The output will not be affected by different permutations because of the symmetry of the max function, which ensures the invariance of displacement.

2.2 Integration

To prevent the interference of too many features on the registration accuracy, the attention mechanism is added to screen the obtained point cloud features; it improves the generalization ability of the model on the premise of ensuring the registration accuracy. Attention mechanisms are introduced into point cloud tasks recently to extract feature information more effectively. Wang et al. [34]. proposed a Graph Attention Network (GAT), which achieved state-of-the-art tasks related to graph structure at that time. GAT can calculate the local attention coefficient or global attention coefficient for each point. The size of the coefficients is changed through training of the attention module so that more important features can be highlighted and the influence of irrelevant features can be reduced.

In the second step, integration, the goal is to rescreen the feature matrixes and obtain the correlation information between the source point cloud and the target point cloud. This paper introduced a variant of self-attention, offset-attention, which is used to process the graph structure data from the EdgeConv modules and focus the task on obtaining high-quality feature sets to realize the above work. Different features will get different attention scores according to their importance so that the feature information obtained in the extraction step can be rescreened and combined to obtain a similar relationship with a higher degree of fit for feature matching.

The offset-attention mechanism calculates the attention of a node in the graph relative to each adjacent node and connects the feature of the node itself and the attention feature as the node's final feature. The main purpose of the graph attention network is to learn a function $g : \mathbb{R}^F \rightarrow \mathbb{R}^K$, where F and K identify the feature dimension. This function maps the input feature H into a new set of vertex features $H' = \{h'_1, h'_2, \dots, h'_N\}$, $h'_i \in \mathbb{R}^K$. At the same time, the function can keep the relationship between these output features unchanged, that is, the category represented by the point cloud remains unchanged. Different from the relatively fixed neighborhood relations in 2D images, graph attention convolution can handle disordered and variable-sized neighborhoods according to the disordered structure of point clouds and assign weights reasonably. As shown in Fig. 3, it shows the effects of the point cloud convolution operation whether there is an attention mechanism. Unlike convolution without attention (as Fig. 3a), the attention mechanism distinguishes the degree of importance between point pairs (reflected in the thickness of edges in Fig. 3b). The importance degree between

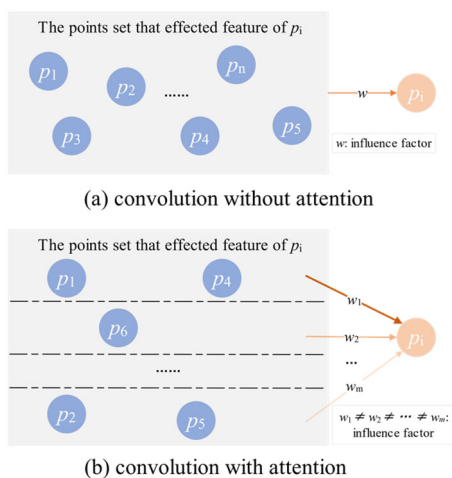


Fig. 3 The difference between the two types of convolution

the point pairs is the weight of the network, which is constantly updated by the network training after initialization, to obtain more excellent weights. The neighborhood point cloud with similar features to the point cloud will be assigned high weight and defined as associated point pair, while the neighborhood point cloud that is not similar to the point cloud will be assigned low weight and defined as non-associated point pair to distinguish matching point pairs, which is critical for efficient point cloud registration capabilities.

From a given point cloud set $P = (p_1, p_2, \dots, p_n) \in \mathbb{R}^3$ construct a point cloud graph structure $G = (V, E)$ according to its neighborhood information, where $V = \{1, \dots, N\}$ is the vertices of the graph, N represents the number of vertices, and E represents the edges between the points. Define $N(i) = \{j : (i, j) \in E\} \cup \{i\}$ to represent the neighborhood set of a point p_i in the point cloud. In our network, its input is a node feature vector matrix. In the point cloud registration task, the input is defined as $H = \{h_1, h_2 \dots h_N\}$, $h_i \in \mathbb{R}^F$, which represents the feature set of the input point cloud, and each feature $h_i \in \mathbb{R}^F$ corresponds to the graph vertex p_i , where F represents the dimension of the point cloud feature. The output is the new node feature matrix $H' = \{h'_1, h'_2 \dots h'_N\}$, $h'_i \in \mathbb{R}^F$.

The offset-attention, which is an improved version of [36] and more suitable for point cloud processing, is used to generate the output $P_S^{out}, P_T^{out} \in \mathbb{R}^{GAT}$, so that the internal point cloud and between the two-point clouds contextual information can be exchanged. The offset-attention used is a variant of multi-head self-attention. Its framework is shown in Fig. 4. The multi-head attention mechanism divides the feature space into N independent subspaces (here, $N = 4$) and calculates the attention scores in each subspace. Then, the scores of the subspaces are combined through the concatenation operator. This improves the parallel processing capacity of the self-attention mechanism. In addition, inspired by

the idea of using the Laplace matrix in graph neural networks, offset-attention calculates the difference between the self-attention (SA) features and the input features by element-wise subtraction [34].

The calculation process of offset-attention is shown in Fig. 5. It is divided into three stages: calculating the similarity score e_{ij} , the probability distribution of attention α_{ij} , and the final attention score. To obtain ampler feature expression capabilities, a learnable and shared linear transformation parameter matrix $W \in \mathbb{R}^{F \times F}$ is first applied to each node feature vector to obtain a new representation $z = Wh_i$. The similarity score e_{ij} between the i -th (h_i) and j -th (h_j) nodes is shown in Eq. (4), which is obtained from the features of itself and its neighbor points, and the convolution kernel can dynamically adapt to the structure of the object.

$$e_{ij} = \text{LeakReLU}(\alpha^T [z_i, z_j]) \tag{4}$$

The linear transformation z_i, z_j obtained by h_i and h_j is concatenated. Then do a dot product with a weight vector α , and obtain e_{ij} by using the LeakRelu activation function. The expression of LeakRelu is shown in Eq. (5).

$$\text{LeakReLU}(z) = \begin{cases} z & z > 0 \\ 0.1z & z < 0 \end{cases} \tag{5}$$

Next, in order to deal with neighbors that vary on different vertices and spatial scales, the attention weights are normalized in all neighbor points of vertex i as shown in Eq. (6):

$$\bar{\alpha}_{ij} = \text{SoftMax}(e_{ij}) = \frac{\exp(e_{ij})}{\sum_{k \in N(i)} \exp(e_{ik})}$$

$$\alpha_{ij} = \frac{\bar{\alpha}_{ij}}{\sum_k \bar{\alpha}_{ij}} \tag{6}$$

and the self-attention weight value α_{ij} is calculated by the SoftMax function. It should be pointed out that offset-attention uses a unique calculation method here, the experimental results show that it can reduce the interference of noise and is beneficial to downstream tasks.

Finally, a new node feature matrix is obtained using the offset-attention weight α_{ij} , and the symbolic representation is shown in Eq. (7), the symbol \cdot stands for convolution and b_i is bias.

$$\text{Attention}(h_i) = \sum_{j \in N(i)} \alpha_{ij} \cdot f(h_j) + b_i \tag{7}$$

The P_S and P_T encoded the contextual features of the source point cloud and the target point cloud, respectively. but the point clouds are not understood. To enhance the correlation between features, it is necessary to add some cross-talk

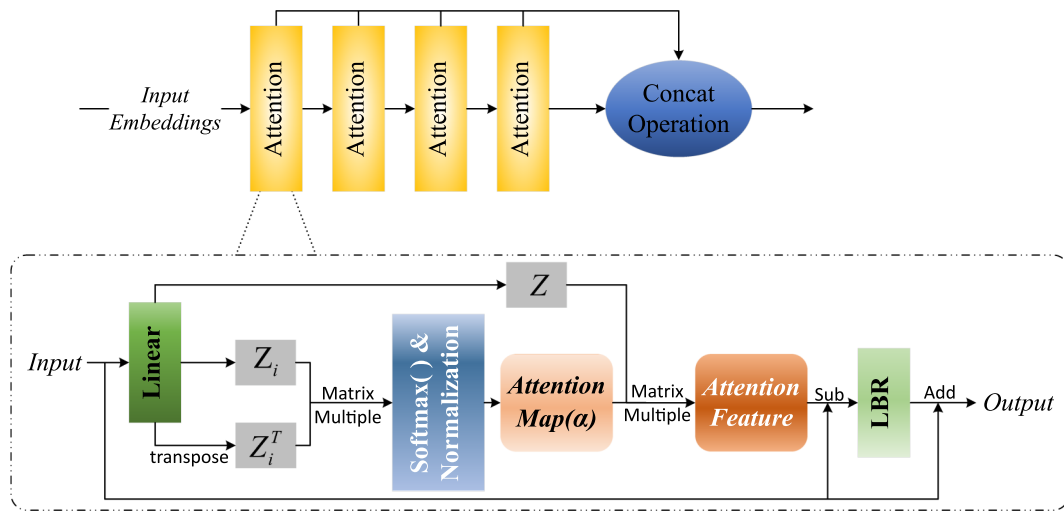


Fig. 4 The structure of offset-attention

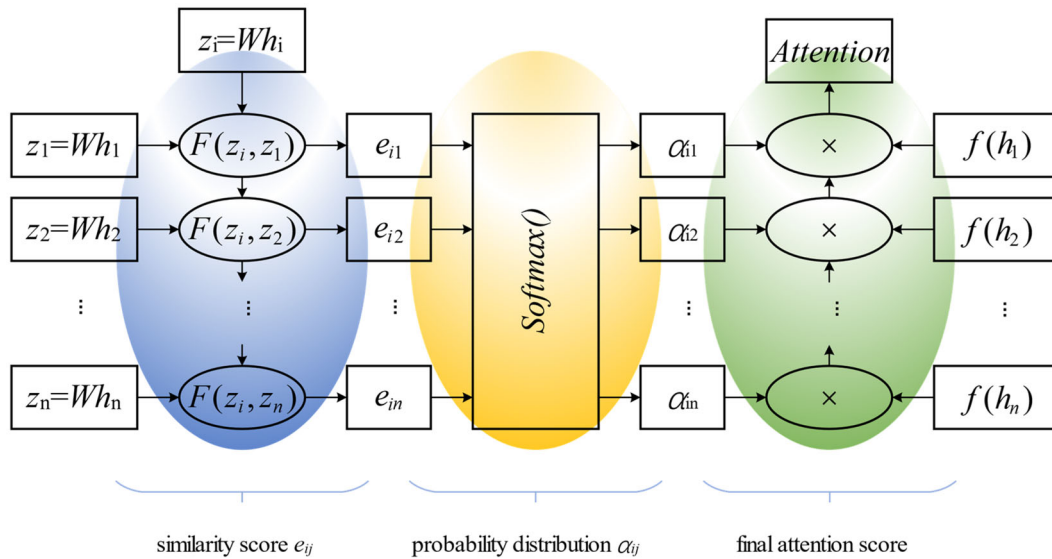


Fig. 5 The calculation process of the offset-attention

at the level of superpoints to the network and learn the importance weight of proximity [37]. Before concatenating the two feature codes, a graph neural network (GNN) is first used to further aggregate and strengthen their contextual relations. First, K-NN is used to connect the superpoints from $P_{feature}$ to the graph in Euclidean space. Let $x_i \in \mathbb{R}^{F'}$ denote the feature encoding of the superpoints $P_{feature}$, and $(i, j) \in E$ be the graph edge between x_i and x_j in $P_{feature}$. The encoded features are iteratively updated by the k-th EdgeConv block, and using Eq. (8).

$$^{(k+1)}x_i = \max_{(i, j) \in E} h_{\theta}(\text{cat}[^{(k)}x_i, ^{(k)}x_j - ^{(k)}x_i]) \tag{8}$$

The h_{θ} represents linear layer, LeakyReLU activation function, and instance normalization. The function max represents the maximum pooling layer. And the cat function represents concatenation.

The update is performed twice using the unshared parameter θ , and the final feature $x_i^{GNN} \in \mathbb{R}^G$ is shown in Eq. (9):

$$x_i^{GNN} = h_{\theta}(\text{cat}[^{(0)}x_i, ^{(1)}x_i, ^{(2)}x_i]) \tag{9}$$

Then, for obtaining the correlation information between the two point clouds, the two superpoints obtained by Eq. (12) are connected to form a bipartite graph. Inspired by the attention, based on the key value $k_j \in \mathbb{R}^{F'}$, the query vector $s_i \in \mathbb{R}^{F'}$ is used to retrieve other superpoints $v_j \in \mathbb{R}^{F'}$, the

symbols are expressed as Eq. (10) where W_k, W_v and W_s are learnable weight matrices:

$$\begin{aligned} k_j &= W_k \cdot x_j^{\text{GNN}} \\ v_j &= W_v \cdot x_j^{\text{GNN}} \\ s_i &= W_s \cdot x_i^{\text{GNN}} \end{aligned} \tag{10}$$

2.3 Registration based on SVD

After obtaining the encoded point cloud features, the final step is to estimate the transformation matrix. For the case where T is a rigid transformation, the point cloud registration problem can be described by Eq. (11). The purpose of this equation is to calculate the rotation matrix R and the translation matrix t that have a lower deviation between the transformed source point cloud $R \cdot P_S + t$ and target point cloud P_T .

$$(R, t) = \arg \min_{R, t} \sum_{i=1}^{|P_S|} w_i \|P_T^i - (R \cdot P_S^i + t)\|^2 \tag{11}$$

Let \hat{P}_S be the center of P_S and \hat{P}_T be the center of P_T . In theory, the center points of \hat{P}_S and \hat{P}_T should be the same after registration, denoted as $t = \hat{P}_T - R \cdot \hat{P}_S$. So Eq. (11) can be equivalent to Eq. (12).

$$R = \arg \min_{R, t} \sum_{i=1}^{|x_i|} w_i \|y_i - Rx_i\|^2 \tag{12}$$

where $y_i = P_T^i - \hat{P}_T$ and $x_i = P_S^i - \hat{P}_S$. Through this step, the rotation matrix R can be obtained first eliminating the influence of the translation matrix t . In the subsequent steps of solving the rotation matrix R separately, Eq. (12) is equivalent to solving a matrix R satisfying Eq. (13) using SVD. Let $S = XWY^T$, we have $tr(RXWY^T) = tr(RU\Sigma V^T) = tr(\Sigma V^T RU)$. After the rotation matrix R is calculated by SVD, the translation matrix t is obtained by $t = \hat{P}_T - R \cdot \hat{P}_S$.

$$\begin{aligned} R &= \arg \max_{R, t} \sum_{i=1}^{|x_i|} w_i y_i R x_i \\ &= \arg \max_{R, t} [tr(RXWY^T)] \end{aligned} \tag{13}$$

To avoid choosing a non-differentiable hard specification, a probabilistic approach from DCP [29] is used to generate a soft map from one point cloud to another. That is, each $x_i \in P_S$ is assigned a probability vector in P_T as shown in Eq. (14).

$$m(x_i, P_T) = \text{soft max}(y^{\text{GNN}}(x_i^{\text{GNN}})^T) \tag{14}$$

Among them, $y^{\text{GNN}} \in \mathbb{R}^{\text{GAT}}$ is the point cloud feature generated by offset-attention, x_i^{GNN} is the i -th row of the point cloud feature matrix, and $m(x_i, P_T)$ here can be regarded as the soft pointer that each x_i^{GNN} points to y^{GNN} . In this way, a matching average point in P_T^{out} can be generated for each point in P_S^{out} , as shown in Eq. (15).

$$\hat{y}_i = (P_T)^T m(x_i, P_T^{\text{out}}) \in \mathbb{R}^3 \tag{15}$$

2.4 Summary and analysis of our algorithm

The main idea of this model is to use EdgeConv to achieve the goal of enriching the feature map (Sect. 2.1). And using offset-attention to change the influence factor of the feature vectors (Sect. 2.2).

The point cloud feature extraction algorithm introduced in Sect. 2.1 is shown in Algorithm 1. It has a time complexity $O(n_1^2)$, where n_1 is influenced by the number of EdgeConv modules and points. Its space complexity is $S(n_2)$, where n_2 refers to the dimension of layers. And the attention mechanism algorithm introduced in Sect. 2.2 is shown in Algorithm 2. It has a time complexity of $O(n_3)$, where n_3 is influenced by the heads of offset-attention. Its space complexity is $S(1)$ due to the network and data dimensions have not changed.

Algorithm 1 The implement of feature extraction: EdgeConv module

Input: the data of source cloud Ps and target cloud Pt;

Output: the feature maps of point clouds Ps and Pt;

- 1: Initializes source and target point cloud data;
- 2: /*Using same Edge modules to compute the feature map of Ps and Pt, respectively.*/
- 3: **for** n=1 to the preset maximal number of EdgeConv module **do**
- 4: **for** i=1 to the number of a point cloud **do**
- 5: Find the k (k=20) nearest neighbor points of a point p_i ;
- 6: Compute each edge feature between a neighbor point and p_i according to equations (1) and (2);
- 7: A new eigenvector is calculated to represent p_i according to equation (3);
- 8: **end for**
- 9: Save the calculation result (local features) of the EdgeConv module;
- 10: **end for**
- 11: Compute the global feature H combined with the results of EdgeConv modules;

Algorithm 2 The implement integration: offset-attention module.

Input: the feature maps of point clouds Ps and Pt;

Output: establish a point pair mapping relationship between point clouds Ps and Pt;

- 1: Initializes the weight matrix;
- 2: **for** n=1 to the preset maximal number of attention heads **do**
- 3: Compute a new representation of feature maps;
- 4: Obtain the similarity score of each point by equation (4);
- 5: Implemente normalizeton by equations (5) and (6) to get probability distribution;
- 6: Compute and save final attention score of each point by equation (7);
- 7: **end for**
- 8: Concate operation to integrate the attention outputs;
- 9: Complete the establishment of point pairs mapping relationship between point clouds Ps and Pt by equations (8), (9) and (10)

3 Experiment

3.1 Setup of experiments

Compare the performance between our model and recent deep learning-based registration methods: RPMNet [38], DCP-V2 [29], PTRNet [39], GeoTransformer [40], and the classic optimization-based point cloud registration algorithm ICP, of which RPMNet, DCP-V2, and GeoTransformer are the more advanced feature-based deep point cloud registration method in recent years, and PTRNet is an advanced end-to-end registration method.

The experiment is performed utilizing PyTorch, and the hardware configuration involves an NVIDIA GTX 2080ti GPU. The initial learning rate is established at 0.001, the

number of training epochs is 250, and the batch size is configured as 8. Meanwhile, we measured the total parameter number as 4,514,176 using a third-party libraries thop from Python.

3.2 Datasets

When evaluating the effectiveness of our model, it is compared with other methods on two public datasets, ModelNet40 [41] and ShapeNet Part [42]. And we tested our model's universal and robust performance on the local industrial parts dataset; we will be covered this dataset in detail separately in Sect. 4.2.

The ModelNet40 dataset used in our experiment comes from the ModelNet dataset [41]. And this dataset contains

Table 1 ModelNet40: Contrast experimental results

Models	MSE(\mathbf{R})	RMSE(\mathbf{R})	MAE(\mathbf{R})	MSE(\mathbf{t})	RMSE(\mathbf{t})	MAE(\mathbf{t})
ICP	9.458	3.075	5.987	7.642	2.764	0.036
DCP-V2	7.825	2.717	1.967	0.000	0.008	0.006
RPMNet	3.704	1.925	1.538	0.034	0.200	0.016
PTRNet	6.447	2.545	1.762	0.000	0.007	0.023
GeoTransformer	2.084	1.443	1.395	0.000	0.005	0.002
Ours	1.363	1.168	1.311	0.000	0.002	0.001

Bold value represent the lowest error, or highest performance of the model in that specific experimental setting

Table 2 ShapeNet Part: Contrast experimental results

Models	MSE(\mathbf{R})	RMSE(\mathbf{R})	MAE(\mathbf{R})	MSE(\mathbf{t})	RMSE(\mathbf{t})	MAE(\mathbf{t})
ICP	3.326	1.824	4.613	1.481	1.217	0.021
DCP-V2	1.661	1.289	0.877	0.000	0.002	0.001
RPMNet	1.328	1.167	0.797	0.001	0.044	0.009
PTRNet	1.649	1.284	0.762	0.000	0.005	0.004
GeoTransformer	1.197	1.094	0.594	0.000	0.012	0.003
Ours	1.213	1.101	0.630	0.000	0.000	0.001

Bold value represent the lowest error, or highest performance of the model in that specific experimental setting

12,311 mesh CAD models from 40 categories. It is a large dataset released in recent years and is widely used in point cloud processing tasks. In this paper, the experimental configuration on the ModelNet40 dataset is consistent with that of DCP [29], and the training and testing are performed on the ModelNet40 complete dataset, using 9843 models for training and 2468 models for testing.

ShapeNet Part is a subset of ShapeNet Core, containing 16,881 models in 16 categories. To make training and evaluation of learning-based methods possible, ShapeNet Part is split into three parts: 12,136 training models, 1870 evaluation models, and 2874 test models. Also sample 1024 points uniformly from the outer surface of each model, as in previous work.

3.3 Evaluation metrics

The mean square error (MSE), root mean square error (RMSE), and mean absolute error (MAE) between the true value and the predicted value were measured experimentally, and MSE, RMSE, and MAE were used as the evaluation indicators in this paper. Ideally, if the source and template point clouds are perfectly registered, all of these error metrics should be zero. In our results, both the rotation matrix (\mathbf{R}) and the translation matrix (\mathbf{t}) are evaluated, which were denoted as MSE(\mathbf{R}), RMSE(\mathbf{R}), MAE(\mathbf{R}), and MSE(\mathbf{t}), RMSE(\mathbf{t}), MAE(\mathbf{t}) in table, respectively. And all angle measurements in the results are in degrees.

4 Results

4.1 Results on public datasets

(1) Contrast experiment

In the contrast experiment, all categories in ModelNet40 and ShapeNet Part datasets were divided into training and testing sets randomly without knowing their category labels. Tables 1 and 2 are the experiment results under ModelNet40 and ShapeNet Part datasets, respectively.

Comparing the results in Tables 1 and 2, it can be seen that the registration results based on the ShapeNet Part dataset are generally higher than those in the ModelNet40 dataset, which is related to the amount of data in each category in training. ShapeNet Part has an average of 1055 models for each type of object, while ModelNet40 is only equivalent to 30% of it, and the more data, the more fully train the model, so the registration effect of each model on the ShapeNet Part dataset is better than that on the ModelNet40 dataset.

Based on the ModelNet40 dataset, our model is slightly better than that of advanced networks proposed in recent years but underperforms the GeoTransformer model on the ShapeNet Part dataset. The rotation matrices MSE, RMSE, and MAE of GeoTransformer are 0.016, 0.013, and 0.036 lower than ours. It is worth noting that the registration effect of PTRNet is slightly worse than that of the advanced network, which is precisely because it is not sufficient in local feature extraction, and the transformation generation method has a strong dependence on the data, which is confirmed in Table 2.

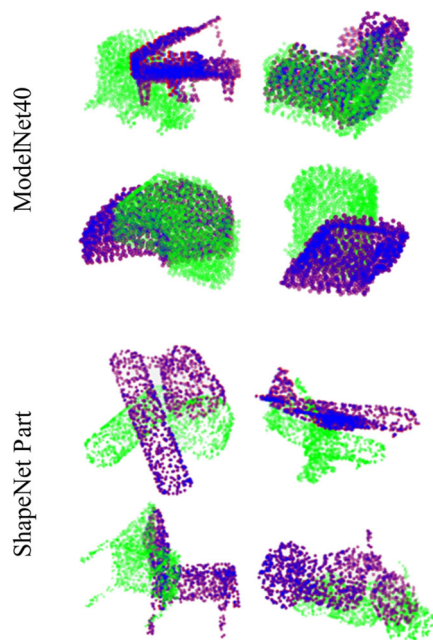


Fig. 6 The registration performance visualization of our framework

Figure 6 shows the registration results of some objects in our framework, the top is the registration results in the ModelNet40 dataset, and the bottom is the ShapeNet Part dataset. The green points and red points represent the source and target point cloud, respectively. The blue points represent the position of the source point cloud after transformation, and the higher the coincidence degree between it and the target point cloud, the darker the color.

(2) Generalizability experiment

For testing the generalization ability of networks, the ModelNet40 dataset is divided into two groups by category randomly: 30 categories are training sets and 10 categories are test sets. Uniformly, 10 categories from the ShapeNet Part dataset are used as the training sets, and 6 categories are used as the test sets.

The experimental results on different datasets are shown in Tables 3 and 4, respectively. Combined with the results of comparison and generalization experiments, the registration errors of all models are larger in the generalization experiment. On the ModelNet40 dataset, GeoTransformer gets a relatively low RMSE(t) value. Our framework reduces the MSE(\mathbf{R}), RMSE(\mathbf{R}), MAE(\mathbf{R}), and MAE(t) to 4.592, 2.143, 1.800, and 0.010, which are lower than all compared models. On the ShapeNet Part dataset, DCP-V2 performs better than others at MSE(t) and RMSE(t). Our framework gets the lower MSE(\mathbf{R}), RMSE(\mathbf{R}), MAE(\mathbf{R}), and MAE(t) values. This suggests that the framework that focuses more on point cloud geometry is more suitable for the generalization task of PCR.

(3) Robustness experiment

To test the robustness of our framework and other models, the noise was sampled independently from $N(0, 0.01)$; and the noise was clipped to $[-0.05, 0.05]$ adding to the data during testing. This experiment uses models trained on all ModelNet40 noise-free data, and the experimental results are shown in Tables 5 and 6.

Comparing Tables 1 and 5 based on the ModelNet40 dataset, and Tables 2 and 6 based on the ShapeNet Part dataset, it is easy to see that GeoTransformer does not perform better than ours, and our framework still comes out on top in the robustness experiments. And DCP-V2 has a large error fluctuation on the ShapeNet Part dataset, which is caused by poor performance on individual data. Although the MAE of RPMNet is not disturbed much, it can be seen from the MSE and RMSE indicators that its registration performance is not stable. In addition, the registration accuracy is not ideal because ICP is prone to fall into local optimal solutions. In summary, our model performs optimally in robustness experiments on the two datasets, indicating that our framework is relatively robust.

(4) Ablation experiment

We added ablation experiments to verify the performance improvement of each module in the proposed framework.

Only the EdgeConv module in the framework proposed in this paper was changed into ordinary convolution, and the dimensions of each layer of convolution were kept unchanged. The experimental results obtained were shown in Table 7. In theory, the EdgeConv module should have excellent performance because it can extract semantic information from multiple dimensions and takes into account both local and global features. This point is also verified by the experimental results in Table 7, which show that the error will be greatly increased when a common convolutional network is adopted.

Similarly, the offset-attention module is changed to self-attention, and the experimental results are shown in Table 8. The results show that offset-attention is more depressed than self-attention in the registration task.

4.2 Results on local industrial parts dataset

(1) Local industrial parts dataset

The local industrial parts dataset is composed of two-parts point cloud data, which are point cloud data collected from two actual parts, respectively, denoted as part a and part b . After denoising, the original data of part a are a point cloud with 344,962 (173×1994) points as shown in Fig. 7, and the original data of part b is a point cloud with 346,236 (172×2013) points as shown in Fig. 8. For using the method in this paper to register the local actual data set, 1024 points are uniformly sampled from each part point cloud when processing the data, and the same method as PointNet [29] is

Table 3 ModelNet40: Generalizability experimental results

Models	MSE(\mathbf{R})	RMSE(\mathbf{R})	MAE(\mathbf{R})	MSE(\mathbf{t})	RMSE(\mathbf{t})	MAE(\mathbf{t})
ICP	11.120	3.335	6.912	9.095	3.154	0.024
DCP-V2	12.604	3.550	2.629	0.003	0.057	0.013
RPMNet	11.529	3.395	2.338	0.001	0.042	0.017
PTRNet	8.168	2.858	2.664	0.003	0.056	0.017
GeoTransformer	6.428	2.535	2.278	0.001	0.037	0.015
Ours	4.592	2.143	1.800	0.003	0.055	0.010

Bold value represent the lowest error, or highest performance of the model in that specific experimental setting

Table 4 ShapeNet Part: Generalizability experimental results

Models	MSE(\mathbf{R})	RMSE(\mathbf{R})	MAE(\mathbf{R})	MSE(\mathbf{t})	RMSE(\mathbf{t})	MAE(\mathbf{t})
ICP	23.946	4.893	6.310	3.942	1.985	0.349
DCP-V2	8.188	2.862	1.975	0.000	0.005	0.004
RPMNet	5.573	2.361	1.361	0.001	0.044	0.006
PTRNet	2.849	1.688	1.235	0.003	0.054	0.011
GeoTransformer	4.485	2.118	1.528	0.004	0.060	0.008
Ours	2.058	1.434	0.961	0.001	0.035	0.003

Bold value represent the lowest error, or highest performance of the model in that specific experimental setting

Table 5 ModelNet40: Robustness experimental results

Models	MSE(\mathbf{R})	RMSE(\mathbf{R})	MAE(\mathbf{R})	MSE(\mathbf{t})	RMSE(\mathbf{t})	MAE(\mathbf{t})
ICP	29.343	5.417	9.831	4.023	2.006	0.201
DCP-V2	24.774	4.977	3.249	0.000	0.007	0.015
RPMNet	22.561	4.750	1.764	0.004	0.061	0.016
PTRNet	10.504	3.241	2.312	0.001	0.034	0.012
GeoTransformer	13.448	3.667	2.007	0.001	0.028	0.008
Ours	9.404	3.067	1.528	0.000	0.017	0.006

Bold value represent the lowest error, or highest performance of the model in that specific experimental setting

used to analyze the two part point clouds. And the data were normalized.

(2) Registration visualization

Combined with the actual industrial application scenario, we select GeoTransformer, PTRNet, and our framework to test the local industrial parts data, which have high registration accuracy, generalization, and noise resistance ability on

public datasets. It helps to save the public resources of the enterprise. These 3 models all run 50 tests on point clouds with different initial positions, and the mean square error (MSE) was used as the evaluation index. The registration effect is shown in Fig. 9.

It shows that in the test based on the point cloud of part α , the mean square errors of rotation matrix $MSE(\mathbf{R})$ and

Table 6 ShapeNet Part: Robustness experimental results

Models	MSE(\mathbf{R})	RMSE(\mathbf{R})	MAE(\mathbf{R})	MSE(\mathbf{t})	RMSE(\mathbf{t})	MAE(\mathbf{t})
ICP	21.203	4.605	8.231	2.734	1.653	0.120
DCP-V2	80.443	8.969	6.421	0.004	0.065	0.046
RPMNet	13.710	3.703	1.320	0.025	0.158	0.013
PTRNet	15.713	3.964	1.924	0.055	0.235	0.032
GeoTransformer	12.005	3.465	1.635	0.056	0.237	0.025
Ours	6.507	2.551	0.761	0.002	0.048	0.004

Bold value represent the lowest error, or highest performance of the model in that specific experimental setting

Table 7 Ablation experimental results of the EdgeConv module

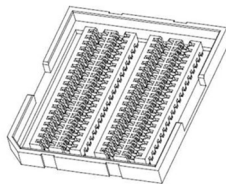
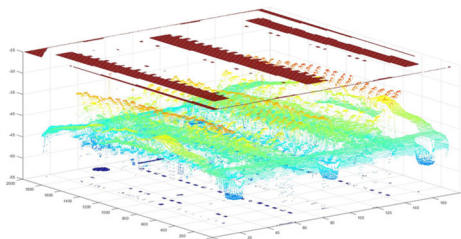
Matrix	Module	ModelNet40			ShapeNet part		
		MSE	RMSE	MAE	MSE	RMSE	MAE
Rotation (\mathbf{R})	EdgeConv	1.363	1.168	1.311	1.213	1.101	0.630
	Convolution	13.752	3.708	4.407	10.487	3.238	2.080
Translation (\mathbf{t})	EdgeConv	0.000	0.002	0.001	0.000	0.000	0.001
	Convolution	3.574	1.891	1.122	3.214	1.793	0.845

Bold value represent the lowest error, or highest performance of the model in that specific experimental setting

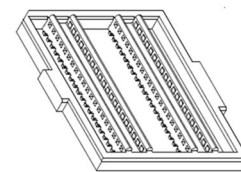
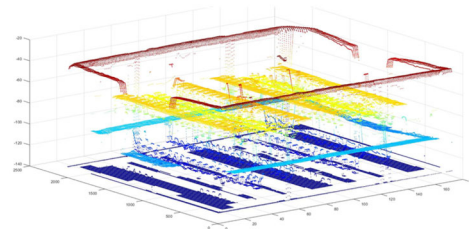
Table 8 Ablation experimental results of the offset-attention module

Matrix	Module	ModelNet40			ShapeNet part		
		MSE	RMSE	MAE	MSE	RMSE	MAE
Rotation (\mathbf{R})	Offset-attention	1.363	1.168	1.311	1.213	1.101	0.630
	Self-attention	2.213	1.488	1.437	1.434	1.198	0.700
Translation (\mathbf{t})	Offset-attention	0.000	0.002	0.001	0.000	0.000	0.001
	Self-attention	0.000	0.001	0.001	0.000	0.000	0.001

Bold value represent the lowest error, or highest performance of the model in that specific experimental setting

(a) The structure of part *a*(b) The point cloud of part *a***Fig. 7** Schematic diagram of part *a*

the mean square errors of translation matrix $MSE(\mathbf{t})$ of our framework are 0.056 and 0.000, respectively. The two errors of PTRNet are 0.472 and 0.005. And The two errors of Geo-Transformer are 1.376 and 0.070, respectively. And the test based on the point cloud of part *b*, the mean square errors of rotation matrix $MSE(\mathbf{R})$ and the mean square errors of translation matrix $MSE(\mathbf{t})$ of our framework are 0.121 and 0.001, respectively. The two errors of PTRNet are 0.179 and 0.004. And The two errors of GeoTransformer are 1.740 and 0.023, respectively.

(a) The structure of part *b*(b) The point cloud of part *b***Fig. 8** Schematic diagram of part *a*

According to the experimental results, our framework performs better in industrial data. The reason is our framework fully considers the geometric fusion relationship between the point cloud which leads to a good registration effect, and better generalization and robustness are also factors for success.

5 Conclusion

This paper proposes a point cloud registration framework based on feature fusion. By combining the graph attention

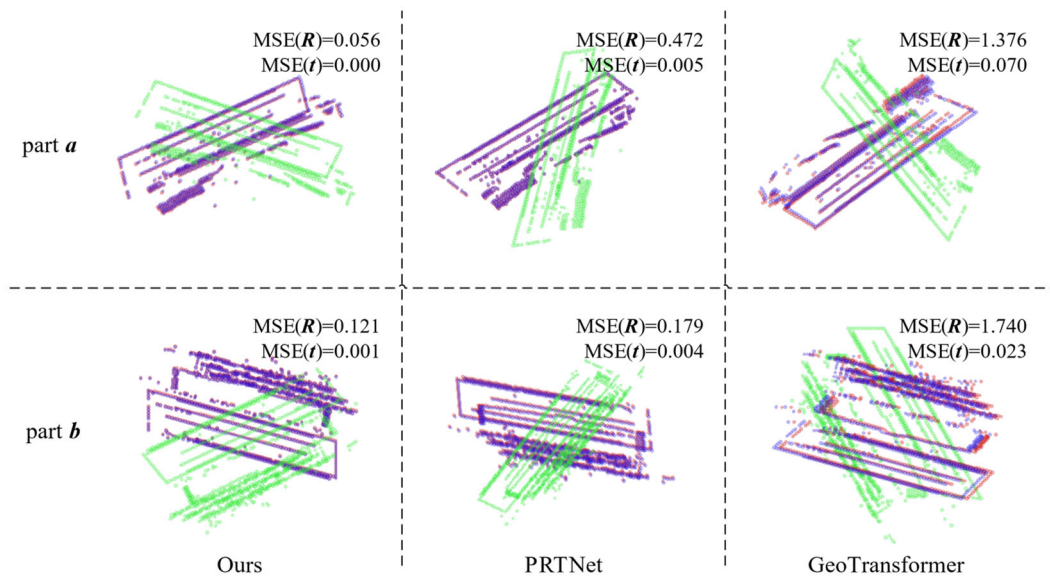


Fig. 9 Registration effect on local industrial parts dataset

network with offset-Attention, a new integration module that integrates point cloud context information and fuses point cloud information is proposed. The experiments show that our framework using this integration module can effectively improve the point cloud feature learning performance, which improves the accuracy of PCR with better generalizability and robustness. And it is highly integrated and can be embedded into other networks. Moreover, the experiments on the local parts dataset also show that our framework can be well applied to real data and is more universal.

We noticed that in actual production operations, the interference caused by complex environments to the point cloud data generated by 3D sensors should not be underestimated. Inspired by the operation of dehazing in 2D images [43, 44], we hope to solve the above problem of excessive point cloud noise effectively.

Acknowledgements This study was supported by the National Natural Science Foundation of China under Grant Nos. 62206252. And this work was supported in part by the National Key R&D Program (2020YFB1712401, 2018YFB1701400), Major Science and Technology Project in Henan Province (201300210500), Key scientific research projects of colleges and universities in Henan Province(23A520015).

Data availability The ModelNet40 and ShapeNet Part datasets that support reporting results can be found in the link: <https://3DShapeNets.cs.princeton.edu> and https://web.stanford.edu/~ericyi/project_page/part_annotation/index.html, respectively. In addition, the local industrial parts dataset can be provided on reasonable request.

Declarations

Conflict of interest The authors state that no conflict of interest exists in this study. And they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

- Tang, Y., et al.: Novel visual crack width measurement based on backbone double-scale features for improved detection automation. *Eng. Struct.* **274**, 115158 (2023). <https://doi.org/10.1016/j.engstruct.2022.115158>
- Que, Y., et al.: Automatic classification of asphalt pavement cracks using a novel integrated generative adversarial networks and improved VGG model. *Eng. Struct.* **277**, 115406 (2023). <https://doi.org/10.1016/j.engstruct.2022.115406>
- Tang, W., He, F., Liu, Y., Duan, Y.: MATR: multimodal medical image fusion via multiscale adaptive transformer. *IEEE Trans. Image Process.* **31**, 5134–5149 (2022). <https://doi.org/10.1109/TIP.2022.3193288>
- Si, T., He, F., Zhang, Z., Duan, Y.: Hybrid contrastive learning for unsupervised person re-identification. *IEEE Trans. Multimed.* (2022). <https://doi.org/10.1109/TMM.2022.3174414>
- Yin, J., Shen, J., Gao, X., Crandall, D.J., Yang, R.: Graph neural network and spatiotemporal transformer attention for 3D video object detection from point clouds. *IEEE Trans. Pattern Anal. Mach. Intell.* **45**(8), 9822–9835 (2023). <https://doi.org/10.1109/TPAMI.2021.3125981>
- Meng, Q., Wang, W., Zhou, T., Shen, J., Jia, Y., Gool, L.V.: Towards a weakly supervised framework for 3D point cloud object detection and annotation. *IEEE Trans. Pattern Anal. Mach. Intell.* **44**(8), 4454–4468 (2022). <https://doi.org/10.1109/TPAMI.2021.3063611>
- Yin, J. et al.: ProposalContrast: Unsupervised Pre-training for LiDAR-Based 3D Object Detection. In: *Computer Vision—ECCV 2022*, pp. 17–33. Springer Nature Switzerland, Cham (2022)
- Yin, J. et al.: Semi-supervised 3D Object Detection with Proficient Teachers. In: *Computer Vision—ECCV 2022*, pp. 727–743. Springer Nature Switzerland, Cham (2022)
- Chen, M., Tang, Y., Zou, X., Huang, K., Li, L., He, Y.: High-accuracy multi-camera reconstruction enhanced by adaptive point cloud correction algorithm. *Opt. Lasers Eng.* **122**, 170–183 (2019). <https://doi.org/10.1016/j.optlaseng.2019.06.011>
- Lin, G., Tang, Y., Zou, X., Wang, C.: Three-dimensional reconstruction of guava fruits and branches using instance segmentation

- and geometry analysis. *Comput. Electron. Agric.* **184**, 106107 (2021). <https://doi.org/10.1016/j.compag.2021.106107>
11. Tao, W., Hua, X., He, X., Liu, J., Xu, D.: Automatic multi-view registration of point clouds via a high-quality descriptor and a novel 3D transformation estimation technique. *Vis. Comput.* (2023). <https://doi.org/10.1007/s00371-023-02942-7>
 12. Huang, X., Mei, G., Zhang, J., Abbas, R.: A comprehensive survey on point cloud registration. *ArXiv*, vol. abs/2103.02690 (2021). <https://doi.org/10.48550/arXiv.2103.02690>
 13. Besl, P.J., McKay, N.D.: A method for registration of 3-D shapes. *Proc. SPIE Int. Soc. Opt. Eng.* **14**(3), 239–256 (1992). <https://doi.org/10.1109/34.121791>
 14. Biber, P.: The normal distributions transform: a new approach to laser scan matching. In: *Proceedings 2003 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2003)* (2003). <https://doi.org/10.1109/IROS.2003.1249285>
 15. Aiger, D., Mitra, N.J., Cohen-Or, D.: 4-Points congruent sets for robust pairwise surface registration. *ACM Trans. Graph.* (2008). <https://doi.org/10.1145/13606121360684>
 16. Rusu, R.B., Blodow, N., Marton, Z.C., Beetz, M.: Aligning point cloud views using persistent feature histograms. In: *2008 IEEE/RSJ International Conference on Intelligent Robots and Systems*, September 22–26, 2008, Acropolis Convention Center, Nice, France (2008). <https://doi.org/10.1109/IROS.2008.4650967>
 17. Rusu, R.B., Blodow, N., Beetz, M.: Fast point feature histograms (FPFH) for 3D registration. In: *IEEE International Conference on Robotics & Automation* (2009). <https://doi.org/10.1109/ROBOT.2009.5152473>
 18. Salti, S., Tombari, F., Stefano, L.D.: SHOT: unique signatures of histograms for surface and texture description. *Comput. Vis. Image Underst.* **125**(AUG), 251–264 (2014). <https://doi.org/10.1016/j.cviu.2014.04.011>
 19. Johnson, A.E., Hebert, M.: Using spin images for efficient object recognition in cluttered 3D scenes. *IEEE Trans. Pattern Anal. Mach. Intell.* **21**(5), 433–449 (2002). <https://doi.org/10.1109/34.765655>
 20. Frome, A., Huber, D., Kolluri, R., Bülow, T., Malik, J.: Recognizing Objects in Range Data Using Regional Point Descriptors. In: *Computer Vision—ECCV 2004*, pp. 224–237. Springer Berlin Heidelberg, Berlin, Heidelberg (2004). https://doi.org/10.1007/978-3-540-24672-5_18
 21. Pais, G.D., Ramalingam, S., Govindu, V.M., Nascimento, J.C., Miraldo, P.: 3DRegNet: a deep neural network for 3D point registration. In: *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (2020). <https://doi.org/10.1109/CVPR42600.2020.00722>
 22. Lu, W., Wan, G., Zhou, Y., Fu, X., Song, S.: DeepVCP: an end-to-end deep neural network for point cloud registration. In: *2019 IEEE/CVF International Conference on Computer Vision (ICCV)* (2019). <https://doi.org/10.1109/ICCV.2019.00010>
 23. Deng, H., Birdal, T., Ilic, S.: 3D local features for direct pairwise registration. In: *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 15–20 (2019). <https://doi.org/10.1109/CVPR.2019.00336>
 24. Yuan, W., Eckart, B., Kim, K., Jampani, V., Fox, D., Kautz, J.: DeepGMR: learning latent gaussian mixture models for registration. In: *Computer Vision—ECCV 2020*, pp. 733–750. Springer International Publishing, Cham (2020). https://doi.org/10.1007/978-3-030-58558-7_43
 25. Choy, C., Dong, W., Koltun, V.: Deep global registration. In: *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (2020). <https://doi.org/10.1109/CVPR42600.2020.00259>
 26. Ginzburg, D., Raviv, D.: Deep weighted consensus dense correspondence confidence maps for 3d shape registration. In: *2022 IEEE International Conference on Image Processing (ICIP)*, pp. 71–75 (2022). <https://doi.org/10.1109/ICIP46576.2022.9897800>
 27. Wang, L., Chen, J., Li, X., Fang, Y.: Non-Rigid Point Set Registration Networks. *ArXiv*, vol. abs/1904.01428 (2019). <https://doi.org/10.48550/arXiv.1904.01428>
 28. Sarode, V. et al.: PCRNet: Point Cloud Registration Network using PointNet Encoding. *ArXiv*, vol. abs/1908.07906 (2019). <https://doi.org/10.48550/arXiv.1908.07906>
 29. Wang, Y., Solomon, J.: Deep closest point: learning representations for point cloud registration. In: *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 3522–3531 (2019). <https://doi.org/10.1109/ICCV.2019.00362>
 30. Wang, Y., Sun, Y., Liu, Z., Sarma, S.E., Bronstein, M.M., Solomon, J.M.: Dynamic graph CNN for learning on point clouds. *ACM Trans. Graph.* (2018). <https://doi.org/10.1145/3326362>
 31. Wang, Y., Solomon, J.M.: PRNet: Self-Supervised Learning for Partial-to-Partial Registration. *ArXiv*, vol. abs/1910.12240 (2019). <https://doi.org/10.48550/arXiv.1910.12240>
 32. Yan, Z., Hu, R., Yan, X., Chen, L., Huang, H.: RPM-Net: recurrent prediction of motion and parts from point cloud. *ACM Trans. Graph.* **38**(6), 1–15 (2019). <https://doi.org/10.1145/3355089.3356573>
 33. Li, J., Zhang, C., Xu, Z., Zhou, H., Zhang, C.: Iterative distance-aware similarity matrix convolution with mutual-supervised point elimination for efficient point cloud registration. In: *Computer Vision—ECCV 2020*, pp. 378–394. Springer International Publishing, Cham (2020). https://doi.org/10.1007/978-3-030-58586-0_23
 34. Guo, M.H., Cai, J.X., Liu, Z.N., Mu, T.J., Martin, R.R., Hu, S.M.: PCT: point cloud transformer. *Comput. Vis. Media* **7**(2), 13 (2021). <https://doi.org/10.1007/s41095-021-0229-5>
 35. Ao, S., Hu, Q., Yang, B., Markham, A., Guo, Y.: SpinNet: learning a general surface descriptor for 3D point cloud registration. In: *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 11748–11757 (2021). <https://doi.org/10.1109/CVPR46437.2021.01158>
 36. Vaswani, A. et al.: Attention is all you need. Presented at the *Advances in Neural Information Processing Systems*, 2017 (2017)
 37. Huang, S., Gojcic, Z., Usvyatsov, M., Wieser, A., Schindler, K.: PREDATOR: registration of 3D point clouds with low overlap. In: *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4265–4274 (2021). <https://doi.org/10.1109/CVPR46437.2021.00425>
 38. Yew, Z.J., Lee, G.H.: RPM-Net: robust point matching using learned features. In: *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (2020). <https://doi.org/10.1109/CVPR42600.2020.01184>
 39. Li, C., Yang, S., Shi, L., Liu, Y., Li, Y.: PTRNet: global feature and local feature encoding for point cloud registration. *Appl. Sci.* **12**(3), 1741 (2022). <https://doi.org/10.3390/app12031741>
 40. Qin, Z., Yu, H., Wang, C., Guo, Y., Peng, Y., Xu, K.: Geometric transformer for fast and robust point cloud registration. In: *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 11133–11142 (2022). <https://doi.org/10.1109/CVPR52688.2022.01086>
 41. Zhirong, W. et al.: 3D ShapeNets: A deep representation for volumetric shapes. In: *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1912–1920 (2015). <https://doi.org/10.1109/CVPR.2015.7298801>
 42. Yi, L., et al.: A scalable active framework for region annotation in 3D shape collections. *ACM Trans. Graph.* **35**(6), 210 (2016). <https://doi.org/10.1145/2980179.2980238>
 43. Zhang, J., He, F., Duan, Y., Yang, S.: AIDEDNet: anti-interference and detail enhancement dehazing network for real-world scenes. *Front. Comput. Sci.* **17**(2), 172703 (2022). <https://doi.org/10.1007/s11704-022-1523-9>

44. Zhang, S., He, F.: DRCDN: learning deep residual convolutional dehazing networks. *Vis. Comput.* **36**(9), 1797–1808 (2020). <https://doi.org/10.1007/s00371-019-01774-8>

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Springer Nature or its licensor (e.g. a society or other partner) holds exclusive rights to this article under a publishing agreement with the author(s) or other rightsholder(s); author self-archiving of the accepted manuscript version of this article is solely governed by the terms of such publishing agreement and applicable law.



Cuixia Li received the B.S. degree in computer science and technology from Henan University, China, and received the Master's degree from Beijing Jiaotong University, China, in 2006. She is currently a professor and working toward the Ph. D. degree in Control Science and Engineering at Zhengzhou University, China. Her research interests include speech recognition, image registration, image recognition, and machine learning.



Yuyin Guan received the B.S. degree in computer science at Zhengzhou University, China, in 2020. He is currently a master's candidate in the School of Cyber Science and Engineering, Zhengzhou University, Henan, China. His research interests include point cloud registration, computer vision, and deep learning.



Shanshan Yang received the B.S. degree in computer science and technology from Henan University, China, in 2019, and received the Master's degree in computer science at Zhengzhou University, China, in 2022. Her research interests include point cloud registration, image recognition, and deep learning.



Yinghao Li received the B.S. degree in computer science and technology from the University of Zhengzhou University, China, and received the Ph. D. degree from Chongqing University, China, in 2016. He is currently an associate professor in the School of Software at Zhengzhou University. His research interests include image registration, point cloud registration, image segmentation, and machine learning.