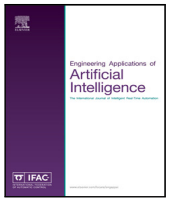
Engineering Applications of Artificial Intelligence 120 (2023) 105817

Contents lists available at ScienceDirect

Engineering Applications of Artificial Intelligence

journal homepage: www.elsevier.com/locate/engappai



A dynamic graph aggregation framework for 3D point cloud registration

Feilong Cao [a,b,*], Jiatong Shi [b], Chenglin Wen [a]

[a] Guangdong Provincial Key Laboratory of Petrochemical Equipment Fault Diagnosis, Guangdong University of Petrochemical Technology, Maoming 525000, Guangdong Province, China

[b] Department of Applied Mathematics, College of Sciences, China Jiliang University, Hangzhou 310018, Zhejiang, China

ARTICLE INFO

Keywords:
Deep learning
3D point cloud
Registration
Dynamic graph

ABSTRACT

Currently, most of the existing point cloud registration methods use stacked edge convolution as feature extractor, ignoring the importance of deep semantic information, meanwhile, the use of attention mechanism tends to increase the computational cost of the model and limits the matching effect. This paper proposes a dynamic graph aggregation framework for point cloud registration by building a dynamic deep network, which can capture richer semantic information and shape properties of point clouds. First, a hybrid feature extractor, which fully fuses local graph information and global graph information, is designed to obtain more discriminative feature descriptors. Second, a local graph neighborhood scoring module to update local graph features and achieve feature enhancement is built by learning the difference of similar point neighborhood features. Finally, a dual-constrained matching module is designed, which is used to measure the similarity of point pairs from the two aspects of Euclidean distance and affinity between features. The proposed framework achieves excellent registration performance, as demonstrated by experimental results on challenging 3D point cloud benchmarks. Especially in partial point cloud registration, MAE($\boldsymbol{R}$) achieved excellent results of 0.2614 and 0.2619 under unseen shapes and unseen categories.

1. Introduction

Background. Finding point-to-point correspondence between two rigid 3D shapes is one of the crucial problems in computer vision (Yang et al., 2019) and remote sensing (Hong et al., 2020). With the increasing availability of 3D data, point cloud registration (Tam et al., 2012) has become a significant part of several promising applications, such as robotics (Pomerleau et al., 2015; Károly et al., 2020), autonomous driving (Feng et al., 2020; Li et al., 2020a), virtual reality, scene reconstruction, multi-view stereo, motion tracking, and so on. However, there are still various influencing factors such as incomplete data, noise, object occlusion, and camera viewpoint changes. Estimating rigid transformations from point clouds with unknown correspondence remains challenging.

Literature review. Most classical methods, including iterative closest point (ICP) (Besl and Mckay, 1992) and its derivatives (Segal et al., 2009; Yang et al., 2013) obtained point correspondences by establishing geometric relations between point pairs. Furthermore, some global alignment methods operate on candidate point pairs, such as using pose clustering or random sample consensus (Fischler and Bolles, 1981) as an iterative fitting model. Each iteration samples a set of candidate correspondences from which alignment results are generated and evaluated. If it is not ideal, iteratively perform local refinement for better alignment. However, most of these methods are time-consuming and difficult to popularize in practical applications.

Inspired by the successful application of deep learning methods (Guo et al., 2020) in the field of point cloud processing, many learning-based registrations methods have been proposed. Generally, features are extracted by neural networks and combined with matching methods to obtain a correspondence matrix. Then, the singular value decomposition (SVD) (Akritas and Malaschonok, 2004) is performed on the matching matrix to obtain the transformation matrix with acceptable speed. Starting from deep closet point (DCP) (Wang and Solomon, 2019a), Wang et al. (2019) verified that edge convolution proposed in dynamic graph CNN (DGCNN) has excellent performance on the point cloud registration task. Since then, several researchers, e.g., Wang and Solomon (2019b), Fu et al. (2021), and Hezroni et al. (2021), have chosen the stacked edge convolution as the feature extraction module as well.

In addition, in Li et al. (2020b), Li et al. proposed a two-stage point elimination method. Features are extracted only through basic manual feature descriptor (Rusu et al., 2009) or graph convolution (Liu et al., 2019). In Wang et al. (2022), the matching strategy was improved to deal with spatial coordinate features and high-dimensional point cloud features separately. Unfortunately, it also failed to pay attention to the weakness in the feature extraction stage. To solve

* Corresponding author at: Department of Applied Mathematics, College of Sciences, China Jiliang University, Hangzhou 310018, Zhejiang, China.
E-mail addresses: flcao@cjlu.edu.cn (F. Cao), s20080701008@cjlu.edu.cn (J. Shi), wencl@gdupt.edu.cn (C. Wen).

https://doi.org/10.1016/j.engappai.2023.105817
Received 2 November 2022; Received in revised form 21 December 2022; Accepted 1 January 2023
Available online 11 January 2023

the partial point cloud matching problem better, Yew and Lee (2020) combined the robust point matching algorithm with the neural network, extracted features by aggregating neighborhood information, and used the multi-layer perceptron to learn parameters, which achieved excellent performance.

Moreover, many researchers take the form of feature interaction to achieve feature enhancement. At present, most of the existing works, for example, Qin et al. (2022) and Yew and Lee (2022), realize feature interaction through cross-attention between point cloud pairs. Fu et al. (2021) combined the idea of graph matching into point cloud registration, and used inter-graph convolution for feature interaction. There are also some methods to achieve feature fusion by learning through multi-layer perceptron after feature splicing. Recently, Xu et al. (2021, 2022) interacted point cloud features in the form of feature cross-concatenating and also obtained good achievement.

Motivation. First, different from the original DGCNN, the above methods only construct a local graph at the beginning. They do not take the form of dynamic updating in the whole process. The crucial importance of dynamically constructing local graph is ignored. It can constantly change the receptive field, thereby capturing deeper semantic information. Therefore, these methods are not superior in accuracy.

Second, many existing methods extract single features, most of which are local features. The combination of global feature information and local feature information is lacking. However, in practical registration tasks, a single local feature or a global feature often creates ambiguity. Points with similar local descriptors are considered globally to potentially belong to different parts of the point cloud. This can lead to incorrect correspondence and affect the matching results.

Besides, for the current feature interaction methods, the use of attention mechanism tends to increase the computational cost of the model. And the method of feature splicing learning is slightly simpler and lacks interpretability. More novel approaches for feature interaction improvements need to be considered.

This paper proposes a novel dynamic graph aggregation framework (DGAF) to further enhance the ability of feature learning from the above aspects. Specifically, we aim to apply the form of dynamically constructing local graphs between layers to the point registration task to learn feature descriptors. In this way, richer semantic information can be captured in deeper feature space. The transmission of point cloud information realizes long-distance dependence. Based on this general process, the next step is to consider the specific components of each graph aggregation module. In the graph aggregation module, a hybrid feature extractor is designed to obtain feature descriptors that fully fuse local and global information. We utilize central node features combined with neighbor features for learning and aggregation to obtain local graph descriptors. Next, the Euclidean distance of the spatial coordinates is used as the global graph edge to generate the global shape skeleton. Combined with the local features of each node, a global graph descriptor is learned through a graph neural network. An effective hybrid feature descriptor is obtained by concatenating local graph features and global graph features. Feature descriptors cover the information of local neighborhoods and are constrained globally, and the discrimination is significantly enhanced. After obtaining sufficient feature information, feature interaction between point cloud pairs is also necessary. Considering that the neighborhoods of corresponding points should also be similar, a local graph neighborhood scoring module is proposed. We cleverly get the index of the closest point in another point cloud to find its corresponding neighborhood feature. The weights are learned by neighborhood differences, updated and weighted to aggregate new local feature descriptors. So far, the entire graph aggregation module has been constructed. After that, the final similarity matrix is obtained through the double constraint matching module. The matching matrix is constrained from two aspects, Euclidean distance, and feature affinity, respectively. By measuring the differences and similarities between features, the interference of outliers can be effectively reduced.

Contributions. The main contributions of this paper are summarized as follows:

(1) A dynamic graph aggregation framework for point cloud registration is proposed by building dynamic deep network, which can capture richer semantic information and shape properties of point clouds.
(2) We specially design a hybrid feature extractor that fully fuses local graph information and global graph information to obtain more discriminative feature descriptors.
(3) We build a local graph neighborhood scoring module to update local graph features by learning the difference of similar point neighborhood features to achieve feature enhancement.
(4) A dual-constrained matching module is designed to measure the similarity of point pairs from the two aspects of Euclidean distance and affinity between features.

The structure of the rest of this paper is as follows. Section 2 is a brief review of some related works. In Section 3, the proposed framework is described in detail. Section 4 exhibits the experimental setup, results of point cloud registration, and discussions. Section 5 draws conclusions and presents potential outlooks for future work.

2. Related works

In this section, traditional model-free methods and learning-based methods are reviewed in detail.

2.1. Traditional model-free methods

Currently, the most classic point cloud registration method is iterative closest point (ICP) (Besl and Mckay, 1992), which minimizes the error between corresponding points by iteratively modifying the transformation. Unfortunately, ICP needs a good initialization to register effectively, otherwise, it will get stuck in local minima. Some ICP-based variants, such as Generalized-ICP (Segal et al., 2009) and point-to-plane ICP (Low, 2004), suffer from the same problem. Go-ICP proposed in Yang et al. (2013) then uses the branch-n-bound module to search for the global optimal solution, but it is sensitive to outliers due to the least-squares objective. Fast global registration (FGR) methods (Zhou et al., 2016) utilizes manually extracted features to globally register partially overlapping surfaces without initial alignment. In addition, Yang et al. (2020) and Yang and Carlone (2019) proposed feature extraction methods based on human descriptions and used truncated least squares to deal with rigid transformations of point clouds.

2.2. Learning-based methods

Recently, with the introduction of deep learning methods in the field of point clouds, several learning-based registration methods with well-known networks such as PointNet Qi et al. (2017) and DGCNN (Wang et al., 2019) as the backbone have emerged. Aoki et al. (2019) proposed a registration method called PointNetLK by integrating a modified Lucas–Kanade optical flow algorithm (Baker and Matthews, 2004) into PointNet. Wang and Solomon (2019a) proposed DCP, which utilizes DGCNN to extract point cloud features, generated implicit correspondences through transformer (Vaswani et al., 2017), and designed a differentiable SVD module for end-to-end network training. After that, Wang and Solomon (2019b) further designed an actor–critic closest point module that can better predict rigid transformations in order to solve the problem of partial point cloud registration, but there is an occasional non-convergence problem during training. Hezroni et al. (2021) also utilized edge convolution to extract point cloud features, and combined the best buddy criterion into the network to calculate the correct correspondence, which effectively improved the matching accuracy on the basis of DCP (Wang and Solomon, 2019a).

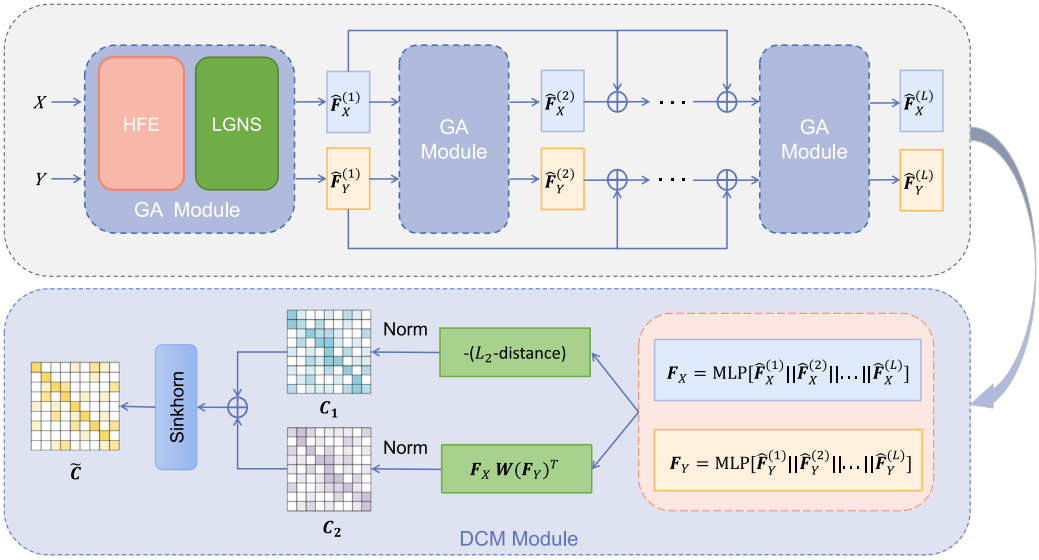


Fig. 1. The architecture of the proposed DGAF. Each GA module is linked by skip connections.

To address the problem of partial-to-partial point cloud registration, Li et al. (2020b) proposed an iterative distance-aware similarity matrix convolution (IDAM) for key point matching through a two-stage point elimination technique. The similarity matrix is learned by the feature and coordinate information of the selected key points. Based on the framework of IDAM, Wang et al. (2022) proposed a novel correspondence search module for correspondence assignment, which utilized shape features and spatial coordinates to independently calculate the matching matrix. Robust point matching network (RPM-Net) proposed by Yew and Lee (2020) combines the traditional algorithm (Gold et al., 1998) with neural networks and adapts a Sinkhorn layer (Cuturi, 2013) to enforce double random constraints on matching maps for reliable correspondence. Zhu et al. (2021) adopted point-by-point feature learning and achieved good performance in partial point cloud matching through a coarse-to-fine registration method, but it was slightly lacking in full point cloud matching. Zhang et al. (2021) designed a local-spatially aware encoder that can efficiently aggregate local and spatial features, focusing on local feature extraction but ignoring the importance of global information. In addition, the robust network based on graph matching (RGM) proposed by Fu et al. (2021) applied the idea of graph matching to the point cloud registration task, interacting features by means of intra-graph convolution and inter-graph convolution, and adopted Sinkhorn to solve the matching matrix, which achieved impressive performance. Xu et al. (2021) proposed an iterative network based on global features, interacting in a feature splicing manner, and learning masks to find overlapping regions. After that, Xu et al. (2022) further proposed a two-branch feature interaction-based structure to strengthen the information association between multiple inputs.

3. Methodology

In this section, the main components of the methodology are mainly introduced in detail, including: architecture, hybrid feature extractor, local graph neighborhood scoring module, and dual-constrained matching module.

3.1. Architecture

The overall framework of the proposed dynamic graph aggregation framework (DGAF) is illustrated in Fig. 1. The input source point cloud $X = \{v_{x_i}\}_{i=1}^{N_1}$ and target point cloud $Y = \{v_{y_j}\}_{j=1}^{N_2}$ are fed into the hierarchical stacked graph aggregation (GA) module and update node features in the form of dynamic graphs, where N_1 and N_2 denote the number of points in X and Y, and v_{x_i} and v_{y_j} represent the points contained in X and Y, respectively. The form of our dynamic graph update is inspired by DGCNN (Wang et al., 2019). The GA module shown in Fig. 2 mainly includes a hybrid feature extractor (HFE) and a local graph neighborhood scoring module (LGNS). The form of skip connection is adopted so that the underlying features are fully transmitted. In the matching process, the dual-constrained matching (DCM) module adopts the similarity matrix obtained from the L_2-distance and the affinity matrix among the features as constraints, and then the Sinkhorn algorithm is used to normalize the rows and columns to obtain the point correspondence matrix. The specific composition of the hybrid feature extractor, local graph neighborhood scoring module, and the dual-constrained matching module will be introduced in the following.

3.2. Hybrid feature extractor

We fuse the local graph structural features and the global graph structural features to form the hybrid feature extractor. The specific process is described in detail.

Let $X = \{x_i\}_{i=1}^{N_1}$ be the source point cloud set. For each $x_i \in X$ $(i = 1, 2, \ldots, N_1)$, we construct N_1 local graphs centered on x_i as well as its K neighbor points $x_{i,k}(k = 1, 2, \ldots, K)$: $G_{x_i}^{\mathrm{local}} = (V_{x_i}, E_{x_i})(i = 1, 2, \ldots, N_1)$, where $V_{x_i} = \{x_i, x_{i,1}, x_{i,2}, \ldots, x_{i,K}\}$ and E_{x_i} are all vertexes and all edges of the local graphs $G_{x_i}^{\mathrm{local}}$, respectively. Let $F_{x_i}^{\mathrm{local}} = \{\boldsymbol{f}_{x_i}, \boldsymbol{f}_{x_{i,1}}, \boldsymbol{f}_{x_{i,2}}, \ldots, \boldsymbol{f}_{x_{i,K}}\} = \{\boldsymbol{f}_{x_{i,k}}\}_{k=1}^{K}$ be a set of features corresponding to the set $V_{x_i} = \{x_i, x_{i,1}, x_{i,2}, \ldots, x_{i,K}\}$, where $\boldsymbol{f}_{x_i}, \boldsymbol{f}_{x_{i,k}} \in \mathbb{R}^C$, and C represents the feature dimension of each node. For all neighbor nodes $\{x_{i,k}\}_{k=1}^{K}$ and their corresponding features $\{\boldsymbol{f}_{x_{i,k}}\}_{k=1}^{K}$, the feature difference can be written as $\Delta\boldsymbol{f}_{x_{i,k}} = \boldsymbol{f}_{x_{i,k}} - \boldsymbol{f}_{x_i}$.

After the MLP operation η and the symmetric aggregation operation $\max$ pooling$(\cdot)$, the final local graph descriptor $\boldsymbol{f}_{x_i}^*$ is achieved:

$$\begin{aligned} \boldsymbol{f}_{x_{i,k}}^* &= \eta([\Delta\boldsymbol{f}_{x_i,k} \parallel \boldsymbol{f}_{x_i}]), \\ \boldsymbol{f}_{x_i}^* &= \max \text{ pooling}(\boldsymbol{f}_{x_{i,k}}^*). \end{aligned} \tag{1}$$

Here "$\parallel$" denotes the concatenation operation. By fusing the feature difference of local nodes, the obtained local features fully contain structural information, which is crucial for the registration of partial point clouds. Similarly, for target point cloud the local graph feature descriptor of the target point cloud $Y = \{y_i\}_{i=1}^{N_2}$, we can construct local graphs $G_{y_j}^{\mathrm{local}} = (V_{y_j}, E_{y_j})(j = 1, 2, \ldots, N_2)$, and the local neighbor features $\boldsymbol{f}_{y_{j,k}}^*$ and the final local graph descriptor $\boldsymbol{f}_{y_j}^*$ can be obtained.

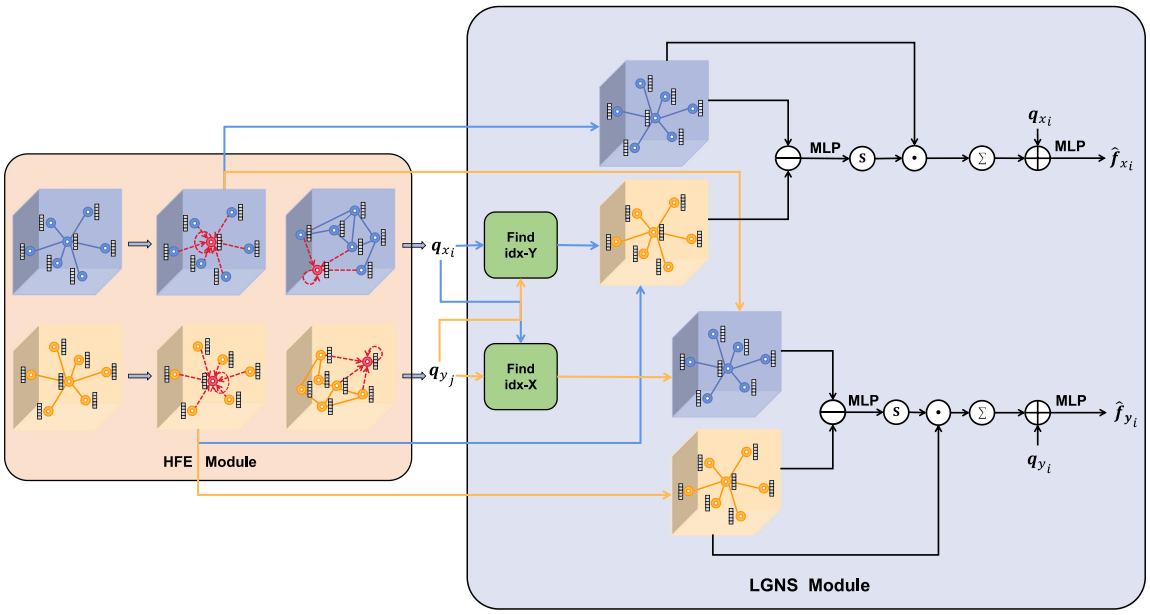


Fig. 2. The GA module mainly consists of a hybrid feature extractor (HFE) and a local graph neighborhood scoring module (LGNS).

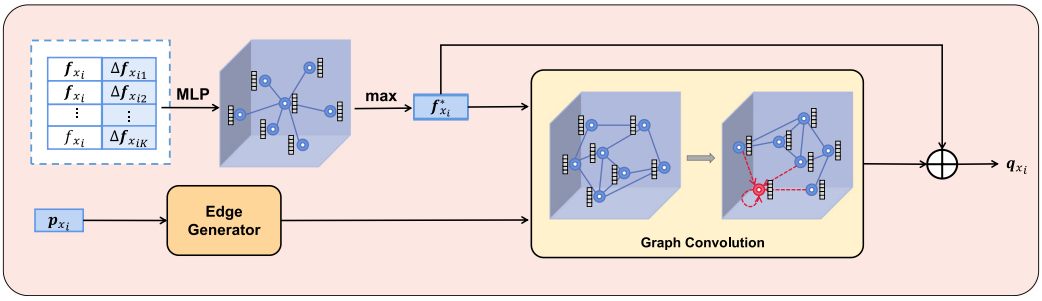


Fig. 3. Structure of the proposed hybrid feature extractor.

Let N_1 and N_2 denote the number of points in the point cloud X and Y, respectively. To obtain the global features of the point cloud, we define a graph $G_X = (V_X, E_X)$ by means of the set of nodes $V_X = \{v_{x_1}, v_{x_2}, \ldots, v_{x_{N_1}}\}$ and edges $E_X \subseteq V_X \times V_X$. Similarly, the graph $G_Y = (V_Y, E_Y)$ can be defined by the set of nodes $V_Y = \{v_{y_1}, v_{y_2}, \ldots, v_{y_{N_2}}\}$ and edges $E_Y \subseteq V_Y \times V_Y$. For the point cloud X, we take the extracted local descriptor $\{\boldsymbol{f}^*_{x_i}\}_{i=1}^{N_1}$ as node features of graph G_X. Then, soft edges are generated for the graph based on the pointwise distances of the original source point cloud coordinates (Luo et al., 2022):

$$E_{x_{i,s}} = \mathrm{e}^{-D_{x_{i,s}}/\sigma^2}, \tag{2}$$

where σ is a hyperparameter in Gaussian kernel, and $D_{x_{i,s}}$ represents the square of Euclidean distance between the coordinates of points v_{x_i} and v_{x_s}. Here, the graph edges generated by the Euclidean distance of the point cloud coordinates can be regarded as the structural skeleton of the point cloud. It can fully describe the overall shape characteristics.

Given the generated graph, the graph neural network is adopted to embed the topological structure information of the graph into their nodes (Wu et al., 2020):

$$\begin{aligned} \boldsymbol{m}_{x_i} &= \frac{\sum_{1\le s\le N_1, s\neq i} E_{x_{i,s}} g_{\text{edge}}(\boldsymbol{f}^*_{x_s})}{\sum_{1\le s\le N_1, s\neq i} E_{x_{i,s}}} \\ \boldsymbol{n}_{x_i} &= g_{\text{node}}(\boldsymbol{f}^*_{x_i}) \\ \boldsymbol{h}_{x_i} &= \boldsymbol{m}_{x_i} + \boldsymbol{n}_{x_i}\ , \end{aligned} \tag{3}$$

where g_{edge} and g_{node} represent two independent nonlinear transformations, which use ReLU (Zou et al., 2020) as activation function to transfer edge information and node information, respectively. Aggregated features from neighbor nodes are normalized by soft edges. Finally, we obtain the pointwise features of the graph aggregation layer through the concatenation operation and the operation ϕ, which can be written as $\boldsymbol{q}_{x_i} = \phi([\boldsymbol{f}^*_{x_i} \parallel \boldsymbol{h}_{x_i}]), \boldsymbol{q}_{x_i} \in \mathbb{R}^D$, here, ϕ is an MLP with the nonlinear activation function, and D denotes the feature dimension. $\boldsymbol{q}_{x_i}$ contains both the local graph structure information of each point and the global graph structure information of the point cloud. The entire process is demonstrated in Fig. 3. Similarly, the features of node v_{y_j} can be expressed as $\boldsymbol{q}_{y_j}$.

By fusing the structural skeleton with node features in the form of graph convolutions, long-range dependencies of point clouds can be achieved. In each dynamic update, the node features change, but the structural skeleton remains the same, which can effectively prevent the feature from deviating. The feature descriptor of each node contains global information, effectively identifying interior points during the point cloud matching process.

3.3. Local graph neighborhood scoring module

According to the graph features of the point cloud, given a pair of point clouds X and Y, for the point v_{x_i} and the point v_{y_j} with one-to-one correspondence, their respective neighborhood features should also be similar. Therefore, we weight the local features of the point

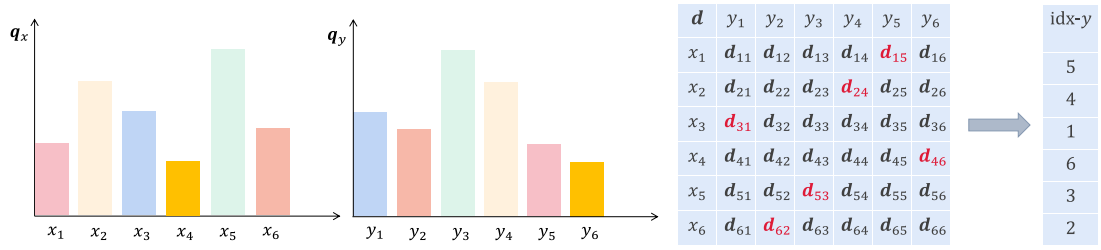


Fig. 4. The method to obtain the index of the point in Y that has the closest feature to X.

cloud by learning the difference of local structure. If the corresponding neighborhoods are similar, it indicates that the point pair is a corresponding relationship, judged as an interior point, and the feature score is high. On the contrary, it is determined as an outlier, and the proportion of features decreases. To this end, the local graph neighborhood scoring module is proposed to update node features by learning the neighborhood features of similar point pairs. As shown in Fig. 2, in the two figures on the left, the abscissa represents the point, and the ordinate represents the features of the point. The third one shows the distance between features, where red represents the minimum distance. The index of the point cloud Y corresponding to the minimum distance is the purpose. Specifically, in the source point cloud branch, according to the feature descriptors of point clouds X and Y, we find that the point in Y is most similar to the features of X, and capture the index of that point. This process is shown in Fig. 4. The point v_{x_i} traverses all points in Y to find the position index of the feature $\boldsymbol{q}_{y_j}$ closest to $\boldsymbol{q}_{x_i}$, where the distance between point v_{x_i} and point v_{y_j} can be calculated by

$$d(v_{x_i}, v_{y_j}) = \sqrt{\sum_{t=1}^{D}(q_{x_{i,t}} - q_{y_{j,t}})^2}\ , \tag{4}$$

and $t \in \{1, 2, \ldots, D\}$ represents the feature dimension. Then, the index of the point in Y that has the closest feature to v_{x_i} can be expressed as:

$$\hat{j} = \arg\min_j \{d(v_{x_i}, v_{y_j})\}_{j=1}^{N_2}. \tag{5}$$

Based on this, we capture the local graph neighborhood features $\boldsymbol{f}^*_{y_{j,k}}$ learned in the hybrid feature extraction stage according to the index $\hat{j}$, and characterize the difference between neighborhoods by fused edge feature subtraction:

$$\Delta \boldsymbol{f}_{ij,k} = \boldsymbol{f}^*_{x_{i,k}} - \boldsymbol{f}^*_{y_{j,k}}. \tag{6}$$

Next, the high-dimensional features are obtained by further learning the edge differences between the two neighborhoods through the operation φ, and the softmax function is applied to achieve the attention coefficients, where φ denotes an MLP with the LeakyReLU activation. Finally, the weighted summation of the local neighbor features is performed to update the node features:

$$\tilde{\boldsymbol{f}}_{x_i} = \sum_{k=1}^{K}(\mathrm{softmax}(\varphi(\Delta \boldsymbol{f}_{ij,k})) \odot \boldsymbol{f}^*_{x_{i,k}}). \tag{7}$$

Similarly, we use the information of the neighborhood of point v_{y_j} to update the neighborhood features of point v_{x_i}, and make its features more similar to $\boldsymbol{f}^*_{y_{j,k}}$ through weighted correction. Then, we concatenate the updated node features $\tilde{\boldsymbol{f}}_{x_i}$ with the features $\boldsymbol{q}_{x_i}$ obtained by the feature extraction module and perform an MLP operation β to obtain the feature descriptor $\hat{\boldsymbol{f}}_{x_i}$ with high discrimination:

$$\hat{\boldsymbol{f}}_{x_i} = \beta([\boldsymbol{q}_{x_i} \parallel \tilde{\boldsymbol{f}}_{x_i}])\ , \tag{8}$$

and $\hat{\boldsymbol{F}}_{x_i} = [\hat{\boldsymbol{f}}^{(1)}_{x_i} \parallel \hat{\boldsymbol{f}}^{(2)}_{x_i} \parallel \cdots \parallel \hat{\boldsymbol{f}}^{(L)}_{x_i}]$. The feature descriptor $\hat{\boldsymbol{f}}_{y_j}$ of the point cloud Y is obtained in the same way, where $\hat{\boldsymbol{F}}_{y_i} = [\hat{\boldsymbol{f}}^{(1)}_{y_i} \parallel \hat{\boldsymbol{f}}^{(2)}_{y_i} \parallel \cdots \parallel \hat{\boldsymbol{f}}^{(L)}_{y_i}]$, and $l \in \{1, 2, \ldots, L\}$ denotes the number of the GA module.

Different from other current feature interaction methods, e.g., feature stitching, cross-attention, and so on, the design of this module aims to achieve the purpose of feature enhancement by scoring the local neighborhood features between point pairs so that each node features are more discriminative.

3.4. Dual-constrained matching module

As shown in Fig. 1, after obtaining the features $\boldsymbol{F}_X = \mathrm{MLP}([\hat{\boldsymbol{F}}^{(1)}_X; \hat{\boldsymbol{F}}^{(2)}_X; \ldots; \hat{\boldsymbol{F}}^{(L)}_X])$ and $\boldsymbol{F}_Y = \mathrm{MLP}([\hat{\boldsymbol{F}}^{(1)}_Y; \hat{\boldsymbol{F}}^{(2)}_Y; \ldots; \hat{\boldsymbol{F}}^{(L)}_Y])$ through the multi-GA module, we apply dual constraints to calculate the similarity matrix. Here, $\hat{\boldsymbol{F}}^{(l)}_X$ and $\hat{\boldsymbol{F}}^{(l)}_F$ represent the overall features of the source point cloud and the target point cloud, respectively, after passing through each layer of GA module. Firstly, since the feature descriptors between corresponding points should be similar, the Euclidean distance between their features should be extremely small. Thus, the similarity matrix $\boldsymbol{C}_1$ is obtained by calculating the Euclidean distance of the features point by point and normalized. The Euclidean distance between a pair of points v_{x_i} and v_{y_j} is calculated as

$$C_{1_{i,j}} = \| \hat{\boldsymbol{F}}_{x_i} - \hat{\boldsymbol{F}}_{y_j} \|_2\ . \tag{9}$$

Since features $\boldsymbol{F}_X$ and $\boldsymbol{F}_Y$ fully contain the graph structure topology information of point clouds X and Y, we take the graph embedding layer-based approach to estimate correspondences and compute affinity matrices (Fu et al., 2021):

$$\boldsymbol{C}_2 = \boldsymbol{F}_X \boldsymbol{W} (\boldsymbol{F}_Y)^{\top}\ , \tag{10}$$

where $\boldsymbol{W}$ represents learnable weights in the affinity layer. By merging two similarity matrices, we obtain the final similarity matrix:

$$\boldsymbol{C} = \frac{\mathrm{IN}(-\boldsymbol{C}_1) + \mathrm{IN}(\boldsymbol{C}_2)}{2}, \tag{11}$$

where IN denotes instance normalization and the Sinkhorn algorithm is then adopted to estimate a soft similarity matrix $\tilde{\boldsymbol{C}} = \mathrm{Sinkhorn}(\boldsymbol{C})$ which is close to binary. When fusing the two matching matrices, we use the method of summing and averaging.

In this way, when the results of the two calculation methods are quite different, a more moderate score will be obtained. Measuring the correspondences from the two directions of the affinity and the difference between the features of the point cloud pair can reduce the misjudgment of outliers and effectively improve the matching rate.

4. Experiments

In this section, the relevant content of the experiment is introduced in detail, mainly including the introduction of the dataset; experimental setting; comparative methods and evaluation indicators; comparative experiments; generalization experiments; and ablation experiments.

4.1. Dataset

(1) *ModelNet40*. The ModelNet40 dataset (Wu et al., 2015) is a dataset that is widely used in 3D point cloud tasks, especially in

point cloud registration, which includes 12311 CAD models belonging to 40 object categories. Randomly sample 1024 points for each point cloud and shuffle the order as the source point cloud. After that, three Euler angle rotations in the range $[0, 45°]$ and translations in the range $[-0.5, 0.5]$ are randomly sampled on each axis to transform the source point cloud into the target point cloud.

Four types of experiments for different data processing methods on ModelNet40 have been completed. Under the registration task of the noisy full point cloud and the registration task of the noisy partial point cloud, experiments are carried out for two different cases of unseen objects and unseen categories, respectively. The specific methods of different data processing are introduced in the following.

Noisy data: Since the influence of noise is widespread, in these experiments, we add Gaussian noise which is sampled from $N(0, 0.01)$ and clipped to $[-0.05, 0.05]$ for each point cloud to evaluate the robustness of the model to noise.

Partial-to-partial point cloud: We adopt the processing of partial point clouds in RPM-Net (Yew and Lee, 2020) to make it closer to the distribution of point cloud data in the real world. All points are projected in a random direction and only 70% points are kept. Since the overlap rate of each set of point cloud pairs cannot be determined, the difficulty of registration is greatly increased.

Unseen objects: We use all samples from these 40 categories for training and testing, of which there are 9843 training samples and the remaining 2468 as test samples.

Unseen categories: In order to verify the generalization ability of the model, we only select samples in the first 20 categories during training and use the samples in the last 20 categories as the test set.

(2) *ShapeNet Part*. The ShapeNet dataset (Yi et al., 2016) is widely used for point cloud tasks and contains 16 880 3-D models in 16 object categories (14 006 for training and 2874 for testing). In this work, all 16 categories are used for testing.

(3) *Stanford 3D*. The Stanford 3D scanning repository[1] is a real-world dataset that has a collection of scan data on several categories of objects including armadillo, bunny, buddha, and dragon. Typically, each scan has more than 100,000 points of data. In this work, we randomly select 2048 points so that they can roughly describe the object and are uniformly distributed.

4.2. Experimental setting

During training, the network is trained by adopting the Adam optimizer and the cross-entropy loss with an initial learning rate of 1×10^{-3} and adaptively adjusting it until 6×10^{-8}. In the full point cloud registration task, the training epochs are set to 50, and the batch size is set to 4. In the partial point cloud registration task, the training epochs are set to 100.

During the experiment, we choose to build the DGAF by 4 GA modules. In the full point cloud registration task, the number of the K nearest neighbor search on the graph is set to $K = 24$. In the partial point cloud registration task, the number of the K nearest neighbor search on the graph is set to $K = 32$. The hyperparameter in Gaussian kernel is set to $\sigma^2 = 0.8$.

We implement our model using PyTorch on NVIDIA RTX 2080Ti. More detailed settings are shown in Table 1.

4.3. Compared methods and evaluation metrics

The proposed DGAF is evaluated on three challenging datasets, namely ModelNet40 dataset, ShapeNet Part dataset, and Stanford 3D dataset, and compared with existing methods. For example, for four different data processing methods on the ModelNet40 dataset, the quantitative results of the proposed network are compared with other methods, including traditional methods and learning-based methods under the same training and testing strategies. The traditional methods ICP (Besl and Mckay, 1992) and FGR (Zhou et al., 2016) are implemented by the Open3D library. Learning-based methods include: DCP-v2 (Wang and Solomon, 2019a), IDAM-GNN (Li et al., 2020b), RPM-Net (Yew and Lee, 2020), RGM (Fu et al., 2021), ROPNet (Zhu et al., 2021), registration based on deep best buddies (DeepBBS) (Hezroni et al., 2021), and MFGNet (Wang et al., 2022), these are based on the model provided by the author to train the network.

We adopt the mean absolute error (MAE) of $\boldsymbol{R}$ and $\boldsymbol{t}$ used in Wang and Solomon (2019a) and the mean isotropic error (Error) of $\boldsymbol{R}$ and $\boldsymbol{t}$ used in Yew and Lee (2020) as evaluation metrics. The mean absolute errors over Euler angles and translation vectors are denote as

$$\mathrm{MAE}(\boldsymbol{R}) = \frac{1}{J}\sum_{j=1}^{J} \| \boldsymbol{R}_{gt_j} - \hat{\boldsymbol{R}}_j \|_1$$

and

$$\mathrm{MAE}(\boldsymbol{t}) = \frac{1}{J}\sum_{j=1}^{J} \|\boldsymbol{t}_{gt_j} - \hat{\boldsymbol{t}}_j\|_1.$$

The mean isotropic rotation error is denoted as

$$\mathrm{Error}(\boldsymbol{R}) = \frac{1}{J}\sum_{j=1}^{J} \angle(\boldsymbol{R}_{gt_j}^{-1}\hat{\boldsymbol{R}}_j)$$

and the translation error is denoted as (Fu et al., 2021)

$$\mathrm{Error}(\boldsymbol{t}) = \frac{1}{J}\sum_{j=1}^{J} \| \boldsymbol{R}_{gt_j}^{-1}\hat{\boldsymbol{t}}_j - \boldsymbol{R}_{gt_j}^{-1}\boldsymbol{t}_{gt_j} \|_2,$$

where $\angle(\boldsymbol{P}) = \arccos(\frac{\mathrm{tr}(\boldsymbol{P})-1}{2})$ returns the rotation angle of matrix $\boldsymbol{P}$ in degrees. Here, J denotes the number of point clouds, and $\{\boldsymbol{R}_{gt}, \boldsymbol{t}_{gt}\}$

Table 1
The detail settings of DGAF (GA = 4) on ModelNet40 dataset.

Partial noisy point cloud (70%) registration on ModelNet40			
Layer name	Number of k-NN search	Input size	Output size
Input layer	–	X: 717 × 3 Y: 717 × 3	–
GA-1	32	X: 717 × 3 Y: 717 × 3	$\hat{F}_{X_1}$: 717 × 128 $\hat{F}_{Y_1}$: 717 × 128
GA-2	32	$\hat{F}_{X_1}$: 717 × 128 $\hat{F}_{Y_1}$: 717 × 128	$\hat{F}_{X_2}$: 717 × 128 $\hat{F}_{Y_2}$: 717 × 128
GA-3	32	$[\hat{F}_{X_1} \parallel \hat{F}_{X_2}]$: 717 × 256 $[\hat{F}_{Y_1} \parallel \hat{F}_{Y_2}]$: 717 × 256	$\hat{F}_{X_3}$: 717 × 128 $\hat{F}_{Y_3}$: 717 × 128
GA-4	32	$[\hat{F}_{X_1} \parallel \hat{F}_{X_3}]$: 717 × 256 $[\hat{F}_{Y_1} \parallel \hat{F}_{Y_3}]$: 717 × 256	$\hat{F}_{X_4}$: 717 × 128 $\hat{F}_{Y_4}$: 717 × 128
L_2-distance	–	$\hat{F}_X$: 717 × 256 $\hat{F}_Y$: 717 × 256	$\boldsymbol{C}_1$: 717 × 717
Affinity layer	–	$\hat{F}_X$: 717 × 256 $\hat{F}_Y$: 717 × 256	$\boldsymbol{C}_2$: 717 × 717
Noisy point cloud (100%) registration on ModelNet40			
Input layer	–	X: 1024 × 3 Y: 1024 × 3	–
GA-1	24	X: 1024 × 3 Y: 1024 × 3	$\hat{F}_{X_1}$: 1024 × 128 $\hat{F}_{Y_1}$: 1024 × 128
GA-2	24	$\hat{F}_{X_1}$: 1024 × 128 $\hat{F}_{Y_1}$: 1024 × 128	$\hat{F}_{X_2}$: 1024 × 128 $\hat{F}_{Y_2}$: 1024 × 128
GA-3	24	$[\hat{F}_{X_1} \parallel \hat{F}_{X_2}]$: 1024 × 256 $[\hat{F}_{Y_1} \parallel \hat{F}_{Y_2}]$: 1024 × 256	$\hat{F}_{X_3}$: 1024 × 128 $\hat{F}_{Y_3}$: 1024 × 128
GA-4	24	$[\hat{F}_{X_1} \parallel \hat{F}_{X_3}]$: 1024 × 256 $[\hat{F}_{Y_1} \parallel \hat{F}_{Y_3}]$: 1024 × 256	$\hat{F}_{X_4}$: 1024 × 128 $\hat{F}_{Y_4}$: 1024 × 128
L_2-distance	–	$\hat{F}_X$: 1024 × 256 $\hat{F}_Y$: 1024 × 256	$\boldsymbol{C}_1$: 1024 × 1024
Affinity layer	–	$\hat{F}_X$: 1024 × 256 $\hat{F}_Y$: 1024 × 256	$\boldsymbol{C}_2$: 1024 × 1024

[1] URL http://www-graphics. stanford. edu/data/3dscanrep.

Table 2
Point clouds with Gaussian noise on unseen objects.

Method	MAE($\boldsymbol{R}$)	MAE(t)	Error($\boldsymbol{R}$)	Error(t)
ICP(1992) (Besl and Mckay, 1992)	5.1373	0.06941	10.5031	0.15031
FGR(2016) (Zhou et al., 2016)	2.5739	0.02709	4.4025	0.05738
DCP-v2(2019) (Wang and Solomon, 2019a)	2.3044	0.00586	4.4185	0.01165
IDAM-GNN(2020) (Li et al., 2020b)	0.8203	0.00421	1.6398	0.00838
RPM-Net(2020) (Yew and Lee, 2020)	0.3051	0.00279	0.5839	0.00580
RGM(2021) (Fu et al., 2021)	0.0798	0.00069	0.1498	0.00141
ROPNet(2021) (Zhu et al., 2021)	0.5720	0.00438	1.1401	0.00928
DeepBBS(2021) (Hezroni et al., 2021)	0.3175	0.00056	0.6226	0.00111
MFGNet(2022) (Wang et al., 2022)	0.7646	0.003340	1.5528	0.00674
DGAF	**0.0596**	**0.00055**	**0.1102**	**0.00111**

Table 3
Point clouds with Gaussian noise on unseen categories.

Method	MAE($\boldsymbol{R}$)	MAE(t)	Error($\boldsymbol{R}$)	Error(t)
ICP(1992) (Besl and Mckay, 1992)	5.7086	0.07591	17.1391	0.23551
FGR(2016) (Zhou et al., 2016)	2.0352	0.02239	3.5493	0.04739
DCP-v2(2019) (Wang and Solomon, 2019a)	2.6288	0.00948	5.0642	0.01878
IDAM-GNN(2020) (Li et al., 2020b)	0.7717	0.00410	1.5267	0.00815
RPM-Net(2020) (Yew and Lee, 2020)	0.3063	0.00269	0.5837	0.00548
RGM(2021) (Fu et al., 2021)	0.0796	0.00069	0.1464	0.00143
ROPNet(2021) (Zhu et al., 2021)	0.4933	0.00391	0.9570	0.00820
DeepBBS(2021) (Hezroni et al., 2021)	0.5861	0.00078	1.1932	0.00159
MFGNet(2022) (Wang et al., 2022)	0.7887	0.00347	1.5672	0.00685
DGAF	**0.0573**	**0.00053**	**0.1046**	**0.00106**

and $\{\hat{\boldsymbol{R}}, \hat{\boldsymbol{t}}\}$ denote the ground truth and estimated transformation, respectively.

4.4. Registration on noisy data

In this experiment, we are interested in aligning the noisy point cloud pairs under unseen objects and unseen categories, respectively. The detail results are presented in Tables 2 and 3. It can be seen that our method has quite significant advantages in the registration of full point clouds. Since there are no outliers in the full point cloud registration task, the focus is to find the correct point correspondence, and the task is less difficult. Even so, after adding noise, only RGM and our method achieve relatively accurate registration. Although DeepBBS has achieved excellent performance in translation indicators, the analysis of experimental results shows that translation indicators are generally better for all these methods. When the translation index reaches an extremely small range, the difference in actual registration is negligible. In contrast, the advantage of our method on the rotation indicator is more obvious. Therefore, the proposed method can effectively learn highly discriminative features, accurately find the correspondence between point pairs, and achieve high-precision registration. Fig. 5(a) and (b) show the visualization comparisons with different methods for unseen objects and unseen categories in noisy full point cloud registration. It can be clearly seen that the other methods are able to achieve correct matching, except ICP. It can be clearly seen that the other methods are able to achieve the correct matching, except ICP. But most methods are flawed in detail. Apparently, in the DCP, the tail of the aircraft is not aligned and the tabletop of the IDAM is separated. And compared to RGM and ROPNet, our method more closely approximates ground truth in subtleties.

4.5. Registration on partial noisy data

In this experiment, partial-to-partial noisy point cloud pairs are aligned under unseen objects and unseen categories, respectively. Since the presence of outliers in the partial-to-partial point cloud registration task, the accuracy is greatly reduced compared to full point cloud registration. Among them, RGM and ROPNet can still show a good registration effect, DeepBBS is slightly worse than them but far higher than other methods. The results in Tables 4 and 5 demonstrate that DGANet still shows superior performance compared to other methods, which greatly improves the matching accuracy. A qualitative improvement has been achieved in both the rotation indicator and the translation indicator. This demonstrates that our DGAF has the ability to discriminate between inliers and outliers while learning highly discriminative features, and can effectively solve the locally overlapping point cloud registration problem. Fig. 5(c) and (d) show the visualizations in detail. It can be clearly seen that in local point cloud registration, most methods cannot get the transformation matrix very accurately. ICP and DCP cannot handle this kind of task. IDAM can only make rough matches. ROPNet and RGM have relatively good results, but some data cannot be effectively registered. Compared with other methods, our method has significant advantages, and the results are closer to the ground truth.

We now use a statistical analysis method (Demšar, 2006; Wang et al., 2013) to show the progressiveness and effectiveness of the proposed method. Following Demšar (2006) and Wang et al. (2013), for a given learning method, the number of wins in m kinds of data obeys the normal distribution $\mathcal{N}(\frac{m}{2}, \frac{\sqrt{m}}{2})$ under the null-hypothesis in the sign test. We assert that the given learning algorithm is significantly better than another one under the significance level a, when the number of wins is at least $\frac{m}{2}+\boldsymbol{Z}_{a/2}\times\frac{\sqrt{m}}{2}$. In our experiments, let $m = 4$ and $a = 0.05$, then $3 < \frac{m}{2} + 1.96 \times \frac{\sqrt{m}}{2} < 4$. Namely, DGAF will achieve considerably better robust performance if the number of wins on 4 kinds of data reaches 4. This can be verified from Table 6.

4.6. Test on the ShapeNet dataset

We test DGAF on the ShapeNet dataset. We test DGAF on the ShapeNet dataset. On the ShapeNet dataset, we use the same partial point cloud data processing method as ModelNet40, and compare the proposed method with other methods. The experimental results are shown in Fig. 6. The left coordinate represents the rotation angle, and the right coordinate represents the translation distance. The histogram shows the rotation indicator under different methods. The line chart shows the translation indicator under different comparison methods. The smaller the number of indicators, the smaller the error and the better the matching effect. It can be seen that the results obtained by ICP, FGR, DCP, and IDAM are not ideal, and it is difficult to

Table 4
Partial-to-partial noisy point clouds on unseen objects.

Method	MAE($\boldsymbol{R}$)	MAE(t)	Error($\boldsymbol{R}$)	Error(t)
ICP(1992) (Besl and Mckay, 1992)	15.9501	0.22913	31.2166	0.47994
FGR(2016) (Zhou et al., 2016)	19.9276	0.17325	33.1749	0.36753
DCP-v2(2019) (Wang and Solomon, 2019a)	6.8872	0.11474	13.0922	0.22893
IDAM-GNN(2020) (Li et al., 2020b)	7.1661	0.07173	13.8356	0.14199
RPM-Net(2020) (Yew and Lee, 2020)	0.7693	0.00702	1.5087	0.01485
RGM(2021) (Fu et al., 2021)	0.5292	0.00425	1.0560	0.00902
ROPNet(2021) (Zhu et al., 2021)	0.8001	0.00679	1.5638	0.01424
DeepBBS(2021) (Hezroni et al., 2021)	1.6016	0.01665	3.2170	0.03323
MFGNet(2022) (Wang et al., 2022)	5.5847	0.05439	10.9073	0.10807
DGAF	**0.2614**	**0.00228**	**0.4980**	**0.00470**

Table 5
Partial-to-partial noisy point clouds on unseen categories.

Method	MAE($\boldsymbol{R}$)	MAE(t)	Error($\boldsymbol{R}$)	Error(t)
ICP(1992) (Besl and Mckay, 1992)	15.6809	0.21883	30.6052	0.45839
FGR(2016) (Zhou et al., 2016)	19.8830	0.16923	32.6171	0.36048
DCP-v2(2019) (Wang and Solomon, 2019a)	7.5655	0.11604	14.4314	0.23055
IDAM-GNN(2020) (Li et al., 2020b)	9.2225	0.07257	17.7428	0.14425
RPM-Net(2020) (Yew and Lee, 2020)	0.8926	0.00867	1.7120	0.01825
RGM(2021) (Fu et al., 2021)	0.9375	0.00764	1.7372	0.01593
ROPNet(2021) (Zhu et al., 2021)	0.7896	0.00712	1.4875	0.01497
DeepBBS(2021) (Hezroni et al., 2021)	2.2020	0.02687	4.4376	0.05338
MFGNet(2022) (Wang et al., 2022)	5.9391	0.05619	11.5261	0.11060
DGAF	**0.2619**	**0.00218**	**0.4871**	**0.00453**

Table 6
The counts of wins, ties, and losses regarding the comparison.

	DGAF versus Contrast methods (win/tie/loss)
Testing result	4/0/0

achieve accurate registration. In contrast, MFGNet is slightly improved on the basis of IDAM, but compared with RPM-Net, ROPNet, RGM and DeepBBS, it still needs to be optimized. It can be clearly seen that our method has obvious advantages, and can achieve more accurate registration both on the rotation indicator and the translation indicator. This shows that our designed GA module as well as the DCM module together achieve an effective matching. The proposed dynamic graph aggregation method can effectively extract the discriminant features of different point cloud datasets. From the representative registration results performed in Fig. 7, our results are quite similar to the ground truth. The results show that our model has strong generalization ability. For a variety of point clouds with large differences in shape, this method can learn effective features and achieve advantageous matching.

4.7. Evaluate on the Stanford 3D

Next, DGAF is tested on 3D point clouds from the Stanford 3D dataset. Fig. 8 shows that DGAF derived from ModelNet40 can be generalized to the Stanford 3D dataset well. Among them, green represents the source point cloud, blue represents the target point cloud, red represents the ground truth and yellow represents the transformed predicted point cloud of the source point cloud. For the input pairs of point clouds in different coordinate systems, our model enables them to achieve an accurate match. Due to device limitations, for such a real object dataset, we can only sample a small number of points from the original point cloud for registration. From the registration results, it can be seen that the proposed method has a significant effect. Our designed GA module is able to learn deep semantic information from limited points, resulting in discriminative features. And our DCM module can further find the correct point correspondence according to the extracted features and achieve effective registration.

Table 7
Ablation studies on ModelNet40 dataset.

Baseline	Ablation studies					MAE($\boldsymbol{R}$)	MAE(t)	Error($\boldsymbol{R}$)	Error(t)
	LGF	LGNS	GGF	DCM	DG				
1	√				√	0.7576	0.00576	1.4486	0.01193
2	√	√			√	0.5727	0.00444	1.1276	0.00888
3	√	√	√		√	0.3273	0.00266	0.6180	0.00549
4	√	√	√	√		3.3772	0.03520	6.424	0.07444
5	√	√	√	√	√	**0.2619**	**0.00218**	**0.4871**	**0.00453**

4.8. Ablation studies

In this subsection, several ablation studies were further conducted on partial noisy point clouds on unseen categories as examples to illustrate the efficiency of various modules. In ablation experiments, we split the HFE module for validation the importance of the combination of the local and global features. There are four baselines set: the first model only extracts local graph features (LGF); the second model adds the local graph neighborhood scoring module (LGNS); the third model merges the global graph features (GGF), the fourth model verifies the importance of building dynamic graphs (DG), and the last model includes the dual-constrained matching module (DCM). As shown in Table 7, each module contributes to the network. With respect to all modules, using the LGNS module improves registration accuracy by correcting local neighborhood features. In addition, the GGF compensates for the global features on the point cloud and improves the MAE($\boldsymbol{R}$) index by 0.2722, thus obtaining the strong discriminative contextual information of the point cloud. Finally, the DCM module reduces false point pair matching, further improving the performance of the network.

4.9. Discussions

(1) Impact of Hyperparameters: The hyperparameters of the proposed model mainly include the number of GA module stacks, the number K of neighbor nodes, and the Gaussian kernel σ used to generate graph edges. In this regard, we conduct experiments on noisy local point clouds on the ModelNet40 dataset. In choosing the number of stacks of modules, we conduct experimental comparisons on the

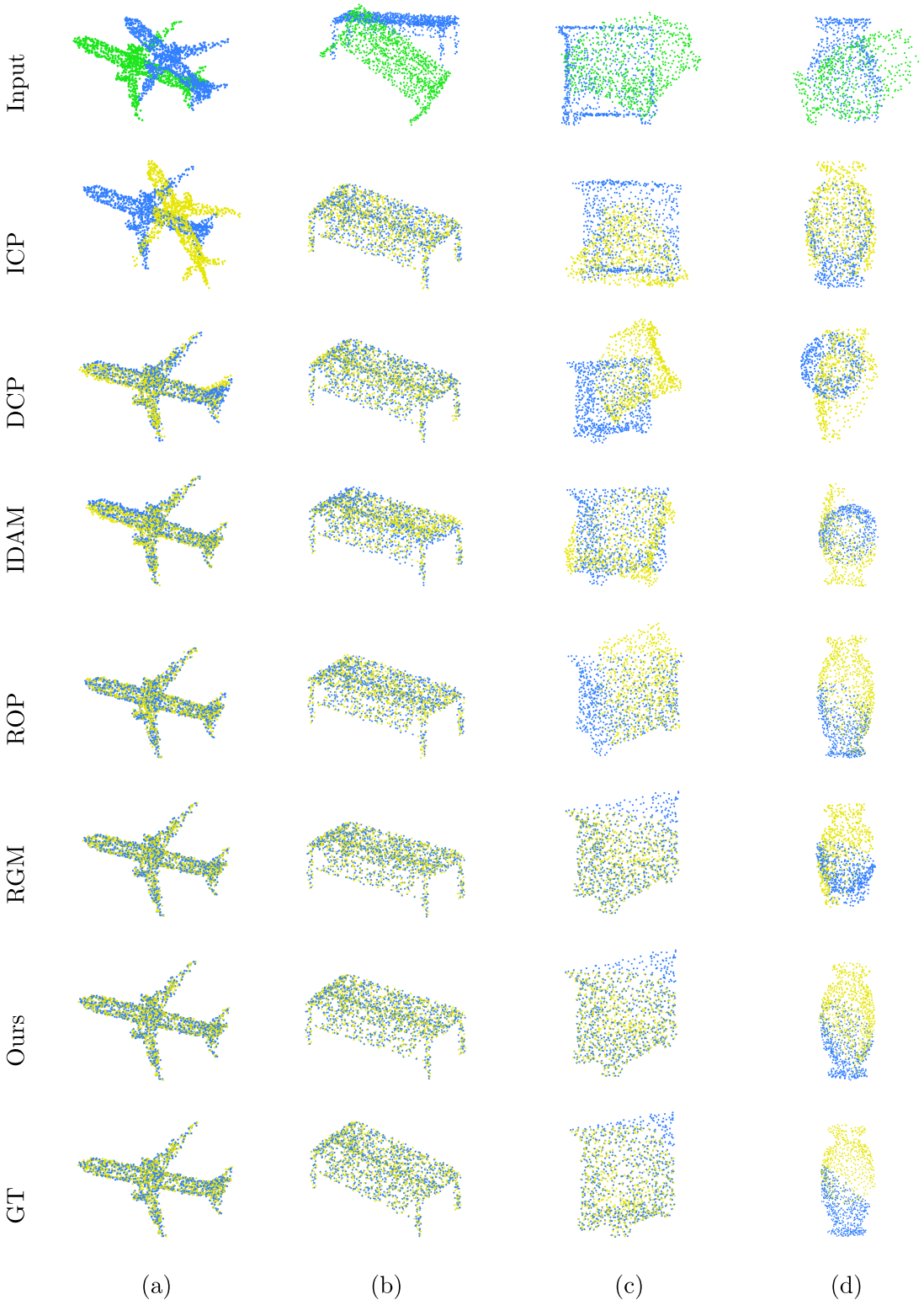


Fig. 5. Qualitative registration results on ModelNet40. The green is the source cloud, the blue is the target cloud, and the yellow is the predication point cloud. (a) Unseen objects with Gaussian noise; (b) Unseen categories with Gaussian noise; (c) Partial-to-partial point clouds on unseen objects; (d) Partial-to-partial point clouds on unseen categories.

models themselves. As a rule of thumb, stacks of modules are set to 3, 4, and 5. As shown in Table 8, when only 3 modules are stacked, a good registration effect can already be achieved, indicating that our module has excellent feature learning ability. Continue to increase the number of modules, it can be seen that when the number of stacks is 4, the accuracy is further improved. As the number of modules continues to increase, we find that although the accuracy has improved, the improvement is not obvious, and the training often increases significantly. Therefore, we finally choose to set the stacking number of modules to 4.

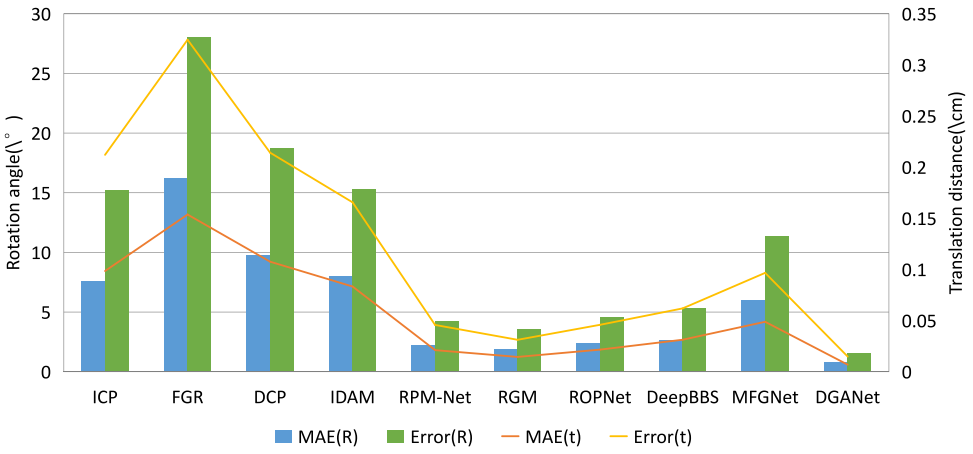


Fig. 6. Test on all categories of the ShapeNet dataset.

Table 8
Discussion of the number of GA layer. We take the experiment on the partial point cloud with Gaussian noise on unseen categories.

GA	MAE($\boldsymbol{R}$)	MAE(t)	Error($\boldsymbol{R}$)	Error(t)	Times (h)
GA = 3	0.2951	0.00231	0.5416	0.00474	**13.68**
GA = 4	0.2619	0.00218	0.4871	0.00453	16.01
GA = 5	**0.2550**	**0.00211**	**0.4808**	**0.00435**	18.53

Table 9
Discussion of the number of selected neighbor points.

K	MAE($\boldsymbol{R}$)	MAE(t)	Error($\boldsymbol{R}$)	Error(t)	Times (h)
$K = 24$	0.3306	0.00254	0.6122	0.00525	**14.22**
$K = 32$	**0.2619**	**0.00218**	**0.4871**	**0.00453**	16.01
$K = 40$	0.2690	0.00220	0.4972	0.456	17.50

Table 10
Discussion of the choice of σ^2 in operations that generate graph edges.

σ^2	MAE($\boldsymbol{R}$)	MAE(t)	Error($\boldsymbol{R}$)	Error(t)
$\sigma^2 = 0.5$	0.2686	0.00228	0.5107	0.00470
$\sigma^2 = 0.8$	**0.2619**	**0.00218**	**0.4871**	**0.00453**
$\sigma^2 = 1.0$	0.2866	0.00233	0.5420	0.00475

In order to choose the optimal number of neighbor nodes, a set of comparative experiments was conducted. Specifically, $K = 24, 32, 40$ were set empirically. The chosen size of the K value determines the size of the local graph. As shown in Table 9, when K is small, the constructed local graph is relatively simple, and the local graph features cannot be accurately captured, resulting in insufficient extraction and weak accuracy. On the other hand, when the value of K increases a lot, the constructed local graph is quite complicated, which will generate plenty of redundant node information. Obviously, the larger the value of K, the more complex the model calculation. And blindly increasing the value of K is not always effective for improving the performance of the model. Therefore, considering both the performance and complexity of the network, the number of neighbor points is empirically set to 32.

For the size of the Gaussian kernel, we empirically chose $\sigma^2 = 0.5, 0.8, 1.0$. From the results in Table 10, it can be seen that the transformation of the Gaussian kernel has a certain impact on the registration accuracy, but it can be kept at a good level. In order to achieve better results, we finally set $\sigma^2 = 0.8$.

Table 11
Number of parameters, FLOPs and the total training time of different methods.

Method	FLOPs (G)	Parameters (M)	Times (h)
DCP-v2 (Wang and Solomon, 2019a)	3.40	1.50	20.83
IDAM-GNN (Li et al., 2020b)	**0.97**	**0.09**	**2.95**
RPM-Net (Yew and Lee, 2020)	2.97	0.91	66.94
RGM (Fu et al., 2021)	36.29	25.00	35.95
ROPNet citezhu2021point	11.20	3.46	81.48
DeepBBS (Hezroni et al., 2021)	9.23	5.67	43.06
MFGNet (Wang et al., 2022)	2.06	**0.09**	5.97
Ours	9.12	2.15	16.01

(2) Convergence: Fig. 9 fully verifies the convergence of the model in the experiment of partial point clouds with unseen categories. The horizontal axis represents the number of training epochs. The coordinates on the left and right represent the rotation angle and translation distance under the MAE indicator, respectively.

(3) Analysis of Computational Cost: The proposed method is compared with some representative methods for some parameters and computational complexity. Taking the ModelNet40 dataset as an example, it is observed in Table 11 that the proposed method has more parameters compared to DCP, IDAM, RPM-Net, and MFGNet. Meanwhile, our framework has more FLOPs than other methods except for RGM, ROPNet and DeepBBS. We have an advantage over most methods in total training time except for IDAM and MFGNet. In addition, in terms of registration accuracy, our model significantly outperforms other methods within acceptable computational cost. Therefore, a better balance between accuracy and complexity is achieved in the proposed method.

5. Conclusion

In this paper, a novel dynamic graph aggregation framework (DGAF) for rigid point cloud registration was proposed, which can capture richer semantic information and global shape properties of point clouds in a dynamic graph update manner. As main contributions, a new hybrid feature extractor to fuses local graph information and global graph information, a novel local graph neighborhood scoring module to update local graph features, and a new dual-constrained matching module are designed, which can obtain more discriminative feature descriptors, achieve feature enhancement, and measure the similarity of point pairs, respectively. Each graph aggregation module consists of a hybrid feature extractor and a local graph neighborhood scoring module. The hybrid feature extractor first aggregates local graph descriptors by neighbor feature differences, and spatial

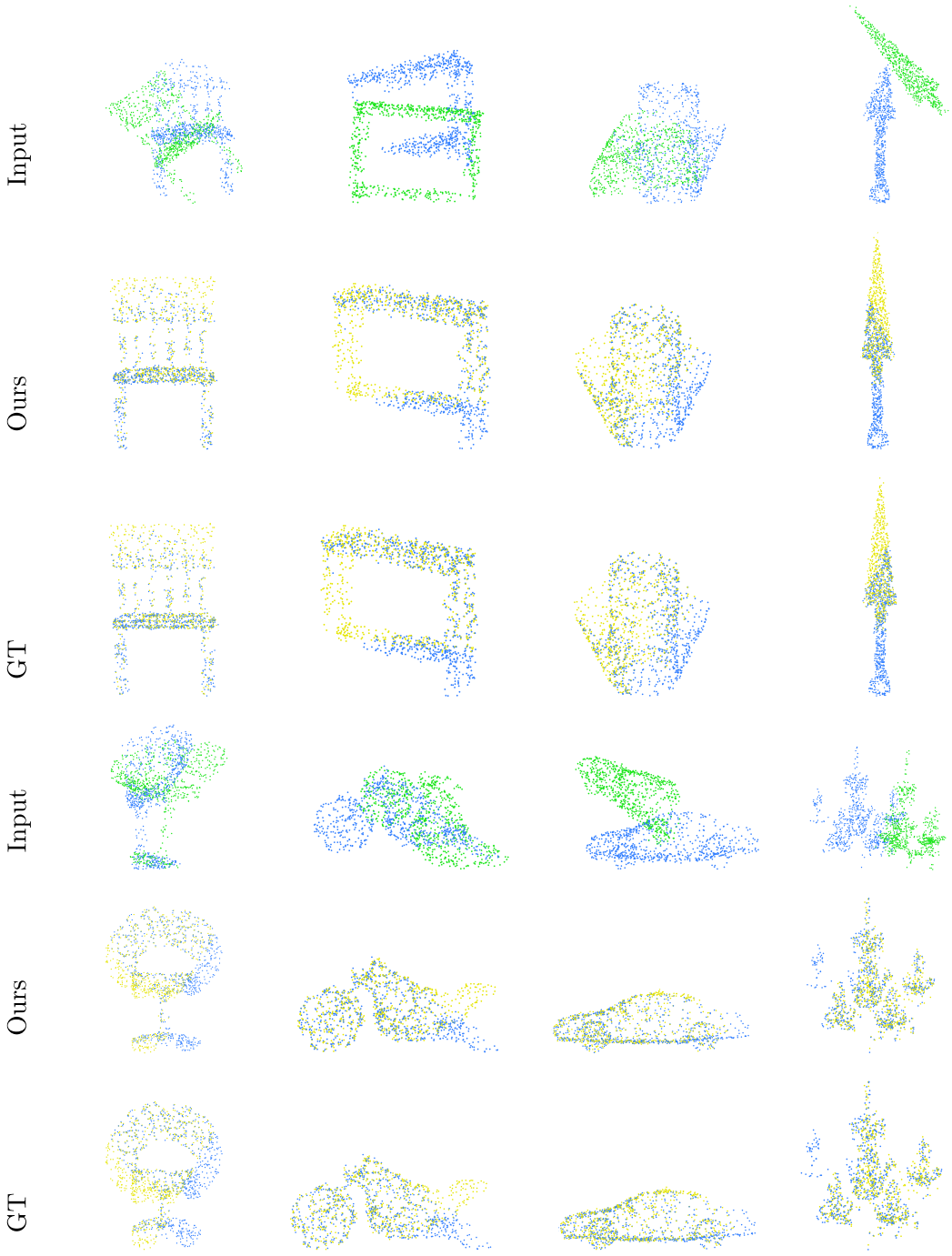


Fig. 7. Qualitative registration results on ShapeNet. The green is the source cloud, the blue is the target cloud, and the yellow is the transformed point cloud.

coordinates are used to generate global graph edges. After that, the local node features and edge features are aggregated by the graph neural network. The local graph neighborhood scoring module starts from the idea that the neighborhood features of the matching points should be similar. The node features are corrected by learning the neighborhood features of the corresponding point. The attention score is constructed by the difference between adjacent features of similar nodes between point cloud pairs, and the features are corrected and aggregated. Furthermore, the similarity matrix is constrained by two measures, Euclidean distance and feature affinity, to enhance the ability to distinguish outliers. Experiments demonstrate the effectiveness of the proposed DGAF for rigid registration of 3D point clouds. Especially for the partial point cloud of unseen categories, this method achieves an excellent effect of the average absolute error of the rotation index as low as 0.2619. It showed that the proposed method can achieve more accurate matching on the object dataset.

In the future, the real scene 3D point cloud registration still is great challenging. The reason is that the scanned real data often contains more types of complex noise, and the overlap rate of point pairs is even lower than 30%. Besides, the interference of numerous outliers results difficulty to discern inliers. Clearly, it is difficult to employ the existed point-by-point learning methods achieving effective registration. So, some effective feature extraction forms in voxels and some targeted matching strategies should be concerned.

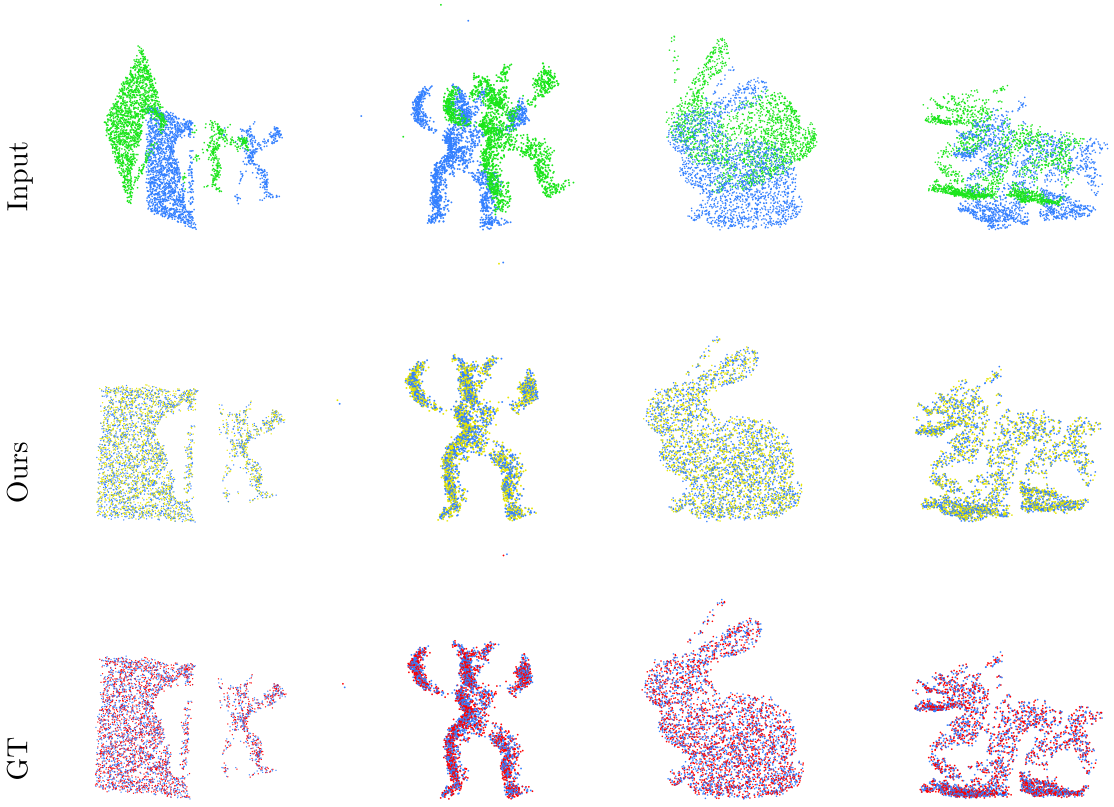


Fig. 8. Qualitative registration results on Stanford 3D.

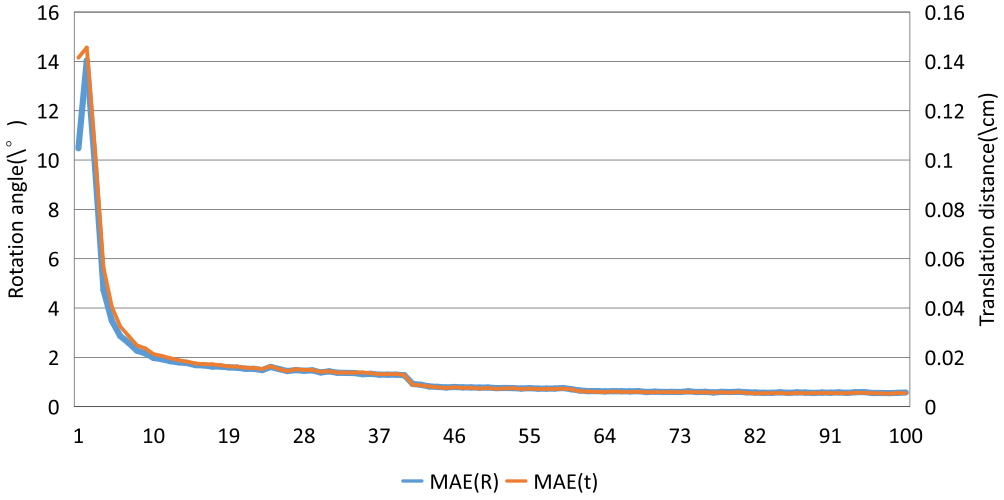


Fig. 9. Rotation mean absolute error and translation mean absolute error decrease during training.

CRediT authorship contribution statement

Feilong Cao: Supervision, Conceptualization, Methodology, Writing – review & editing, Funding acquisition. **Jiatong Shi:** Investigation, Writing – original draft, Software, Data curation. **Chenglin Wen:** Writing – review & editing, Funding acquisition.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

The data that has been used is confidential.

Acknowledgments

This work was supported by the National Natural Science Foundation of China under grants 62032022, 62176244, and 61933013, and the Zhejiang Provincial Natural Science Foundation of China under grant LZ20F030001.

References

Akritas, A.G., Malaschonok, G.I., 2004. Applications of singular-value decomposition (SVD). Math. Comput. Simulation 67 (1–2), 15–31.

Aoki, Y., Goforth, H., Srivatsan, R.A., Lucey, S., 2019. PointNetLK: Robust & efficient point cloud registration using PointNet. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition. Long Beach, USA, pp. 7156–7165.

Baker, S., Matthews, I., 2004. Lucas-Kanade 20 years on: A unifying framework. Int. J. Comput. Vis. 56 (3), 221–255.

Besl, P.J., Mckay, H.D., 1992. A method for registration of 3-D shapes. IEEE Trans. Pattern Anal. Mach. Intell. 14 (2), 239–256.

Cuturi, M., 2013. Sinkhorn distances: Lightspeed computation of optimal transport. In: Advances in Neural Information Processing Systems. Lake Tahoe, USA, pp. 2292–2300.

Demšar, J., 2006. Statistical comparisons of classifiers over multiple data sets. J. Mach. Learn. Res. 7, 1–30.

Feng, D., Haase-Schütz, C., Rosenbaum, L., Hertlein, H., Glaeser, C., Timm, F., Wiesbeck, W., Dietmayer, K., 2020. Deep multi-modal object detection and semantic segmentation for autonomous driving: Datasets, methods, and challenges. IEEE Trans. Intell. Transp. Syst. 22 (3), 1341–1360.

Fischler, M.A., Bolles, R.C., 1981. Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. Commun. ACM 24 (6), 381–395.

Fu, K., Liu, S., Luo, X., Wang, M., 2021. Robust point cloud registration framework based on deep graph matching. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition. Nashville, USA, pp. 8893–8902.

Gold, S., Rangarajan, A., Chien-Ping, L., Pappu, S., Mjolsness, E., 1998. New algorithms for 2D and 3D point matching: pose estimation and correspondence. Pattern Recognit. 31 (8), 1019–1031.

Guo, Y., Wang, H., Hu, Q., Liu, H., Liu, L., Bennamoun, M., 2020. Deep learning for 3D point clouds: A survey. IEEE Trans. Pattern Anal. Mach. Intell. 43 (12), 4338–4364.

Hezroni, I., Drory, A., Giryes, R., Avidan, S., 2021. DeepBBS: Deep best buddies for point cloud registration. In: Proceedings of International Conference on 3D Vision. IEEE, pp. 342–351.

Hong, D., Gao, L., Yokoya, N., Yao, J., Chanussot, J., Du, Q., Zhang, B., 2020. More diverse means better: Multimodal deep learning meets remote-sensing imagery classification. IEEE Trans. Geosci. Remote Sens. 59 (5), 4340–4354.

Károly, A.I., Galambos, P., Kuti, J., Rudas, I.J., 2020. Deep learning in robotics: Survey on model structures and training strategies. IEEE Trans. Syst., Man, Cybern.: Syst. 51 (1), 266–279.

Li, Y., Ma, L., Zhong, Z., Liu, F., Chapman, M.A., Cao, D., Li, J., 2020a. Deep learning for LIDAR point clouds in autonomous driving: A review. IEEE Trans. Neural Netw. Learn. Syst. 32 (8), 3412–3432.

Li, J., Zhang, C., Xu, Z., Zhou, H., Zhang, C., 2020b. Iterative distance-aware similarity matrix convolution with mutual-supervised point elimination for efficient point cloud registration. In: Proceedings of European Conference on Computer Vision. Glasgow, UK, pp. 378–394.

Liu, Y., Fan, B., Xiang, S., Pan, C., 2019. Relation-shape convolutional neural network for point cloud analysis. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 8895–8904.

Low, K.-L., 2004. Linear Least-Squares Optimization for Point-To-Plane ICP Surface Registration, Vol. 4. Chapel Hill, University of North Carolina, pp. 1–3, 10.

Luo, J., Yuan, M., Fu, K., Wang, M., Zhang, C., 2022. Deep graph matching based dense correspondence learning between non-rigid point clouds. IEEE Robot. Autom. Lett. 7 (3), 5842–5849.

Pomerleau, F., Colas, F., Siegwart, R., 2015. A review of point cloud registration algorithms for mobile robotics. Found. Trends Robot. 4 (1), 1–104.

Qi, C.R., Su, H., Mo, K., Guibas, L.J., 2017. PointNet: Deep learning on point sets for 3D classification and segmentation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. Honolulu, USA, pp. 77–85.

Qin, Z., Yu, H., Wang, C., Guo, Y., Peng, Y., Xu, K., 2022. Geometric transformer for fast and robust point cloud registration. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 11143–11152.

Rusu, R.B., Blodow, N., Beetz, M., 2009. Fast point feature histograms (FPFH) for 3D registration. In: Proceedings of IEEE International Conference on Robotics and Automation. IEEE, pp. 3212–3217.

Segal, A., Haehnel, D., Thrun, S., 2009. Generalized-ICP. In: Proceedings of Robotics: Science and Systems. Washington, Seattle, USA, p. 435.

Tam, G.K., Cheng, Z.-Q., Lai, Y.-K., Langbein, F.C., Liu, Y., Marshall, D., Martin, R.R., Sun, X.-F., Rosin, P.L., 2012. Registration of 3D point clouds and meshes: A survey from rigid to nonrigid. IEEE Trans. Vis. Comput. Graphics 19 (7), 1199–1217.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł., Polosukhin, I., 2017. Attention is all you need. In: Advances in Neural Information Processing Systems. Long Beach, USA, pp. 6000–6010.

Wang, X.-Z., He, Y.-L., Wang, D.D., 2013. Non-naive Bayesian classifiers for classification problems with continuous attributes. IEEE Trans. Cybern. 44 (1), 21–39.

Wang, H., Liu, X., Kang, W., Yan, Z., Wang, B., Ning, Q., 2022. Multi-features guidance network for partial-to-partial point cloud registration. Neural Comput. Appl. 34, 1623–1634.

Wang, Y., Solomon, J.M., 2019a. Deep closest point: Learning representations for point cloud registration. In: Proceedings of IEEE International Conference on Computer Vision. Seoul, Korea (South), pp. 3522–3531.

Wang, Y., Solomon, J., 2019b. PRNet: Self-supervised learning for partial-to-partial registration. In: Advances in Neural Information Processing Systems. Vancouver, Canada, pp. 8814–8826.

Wang, Y., Sun, Y., Liu, Z., Sarma, S.E., Bronstein, M.M., Solomon, J.M., 2019. Dynamic graph CNN for learning on point clouds. ACM Trans. Graph. 38 (5), 1–12.

Wu, Z., Pan, S., Chen, F., Long, G., Zhang, C., Philip, S.Y., 2020. A comprehensive survey on graph neural networks. IEEE Trans. Neural Netw. Learn. Syst. 32 (1), 4–24.

Wu, Z., Song, S., Khosla, A., Yu, F., Zhang, L., Tang, X., Xiao, J., 2015. 3D ShapeNets: A deep representation for volumetric shapes. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition. Boston, USA, pp. 1912–1920.

Xu, H., Liu, S., Wang, G., Liu, G., Zeng, B., 2021. Omnet: Learning overlapping mask for partial-to-partial point cloud registration. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 3132–3141.

Xu, H., Ye, N., Liu, G., Zeng, B., Liu, S., 2022. FINet: Dual branches feature interaction for partial-to-partial point cloud registration. In: Proceedings of the AAAI Conference on Artificial Intelligence. pp. 2848–2856.

Yang, H., Carlone, L., 2019. A polynomial-time solution for robust registration with extreme outlier rates. In: Proceedings of Robotics: Science and Systems. FreiburgimBreisgau, Germany, pp. 1–10.

Yang, J., Li, H., Jia, Y., 2013. Go-ICP: Solving 3D registration efficiently and globally optimally. In: Proceedings of IEEE International Conference on Computer Vision. Sydney, Australia, pp. 1457–1464.

Yang, H., Shi, J., Carlone, L., 2020. Teaser: Fast and certifiable point cloud registration. IEEE Trans. Robot. 37 (2), 314–333.

Yang, J., Xian, K., Wang, P., Zhang, Y., 2019. A performance evaluation of correspondence grouping methods for 3D rigid data matching. IEEE Trans. Pattern Anal. Mach. Intell. 43 (6), 1859–1874.

Yew, Z.J., Lee, G.H., 2020. RPM-Net: Robust point matching using learned features. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition. Seattle, USA, pp. 11821–11830.

Yew, Z.J., Lee, G.H., 2022. REGTR: End-to-end point cloud correspondences with transformers. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 6677–6686.

Yi, L., Kim, V.G., Ceylan, D., Shen, I.-C., Yan, M., Su, H., Lu, C., Huang, Q., Sheffer, A., Guibas, L., 2016. A scalable active framework for region annotation in 3D shape collections. ACM Trans. Graph. 35 (6), 1–12.

Zhang, Z., Chen, G., Wang, X., Shu, M., 2021. DDRNet: Fast point cloud registration network for large-scale scenes. ISPRS J. Photogramm. Remote Sens. 175, 184–198.

Zhou, Q.-Y., Park, J., Koltun, V., 2016. Fast global registration. In: Proceedings of European Conference on Computer Vision. Amsterdam, Netherlands, pp. 766–782.

Zhu, L., Liu, D., Lin, C., Yan, R., Gómez-Fernández, F., Yang, N., Feng, Z., 2021. Point cloud registration using representative overlapping points. arXiv preprint arXiv:2107.02583.

Zou, D., Cao, Y., Zhou, D., Gu, Q., 2020. Gradient descent optimizes over-parameterized deep ReLU networks. Mach. Learn. 109 (3), 467–492.