RegiFormer: Unsupervised Point Cloud Registration via Geometric Local-to-Global Transformer and Self-Augmentation

Chengyu Zheng, Mengjiao Ma, Zhilei Chen, Honghua Chen, Weiming Wang, and Mingqiang Wei, *Senior Member, IEEE*

***Abstract*— Representation learning for two partially overlapping point clouds remains an open challenge in unsupervised point cloud registration (U-PCR). In this article, we introduce RegiFormer, a geometric local-to-global transformer (GLGT)-based unsupervised framework equipped with a self-augmentation (SA) strategy, for point cloud registration. The GLGT not only aggregates features from local neighborhoods but also extracts global intrarelationships within the entire point cloud using a transformation-invariant geometry embedding. In addition, it enhances the interrelationships between paired point clouds. To overcome the limited ability of U-PCR methods to learn alignment knowledge, we design an SA strategy that can be flexibly integrated into advanced models, significantly boosting their registration performance. Extensive experiments, conducted on five popular synthetic and real-scanned benchmarks, demonstrate the superior performance of RegiFormer compared to state-of-the-art methods, both qualitatively and quantitatively.**

***Index Terms*— Geometric local-to-global transformer (GLGT), point cloud registration, RegiFormer, self-augmentation (SA).**

I. Introduction

POINT cloud registration serves as the cornerstone in various fields such as 3-D reconstruction [2], SLAM [3], and 3-D location [4], making it a classical topic that has recently experienced a surge in research interest due to the advent of deep learning. The primary objective of point cloud registration is to estimate a rigid transformation, aligning two partially overlapping point clouds.

Many promising solutions have been proposed [5], [6], [7], [8], [9], [10], [11] for point cloud registration. Among them, the traditional methods, e.g., ICP [5] and FPFH [12], involve complicated optimization procedures and many parameters to tweak, which heavily discount the efficiency and user experience; the deep learning-based methods [13], [14], benefiting from large training data, can automatically achieve the optimal registration performance in the run-time stage. The success of deep learning-based methods mainly imputes to the supervised learning paradigm that is fed with a huge amount of data with ground-truth transformation labels. However, in real-world scenarios, we often lack labeled aligned 3-D scans, and human annotations of them are quite labor-intensive and time-consuming due to their irregular structures. Although training on synthetic scans is promising to alleviate the shortage of labeled aligned real-world data, such trained models inevitably suffer from domain shifts. To solve this issue, some efforts [1], [15], [16], [17] are made to focus on unsupervised point cloud registration (U-PCR), achieving substantial progress.

Nevertheless, U-PCR remains an open problem for two reasons. First, current methods, e.g., [1], [16], are often confronted with the challenges related to insufficient and indistinctive feature representations, such as the loss of local cues, the absence of global features and cross-features, and numerous outlier matches. As illustrated in the purple oval regions in Fig. 1(b), depending only on local features makes some outliers salient, leading to confused mismatches. Second, the difficulty of U-PCR lies in the weak ability of networks to learn 3-D alignment knowledge without the supervision of ground-truth transformation labels.

To this end, we propose RegiFormer, a U-PCR method that utilizes a geometric local-to-global transformer (GLGT) for robust deep feature learning, along with a self-augmentation (SA) strategy to facilitate alignment knowledge acquisition, which can make unsupervised methods learn a better initialization of registration status.

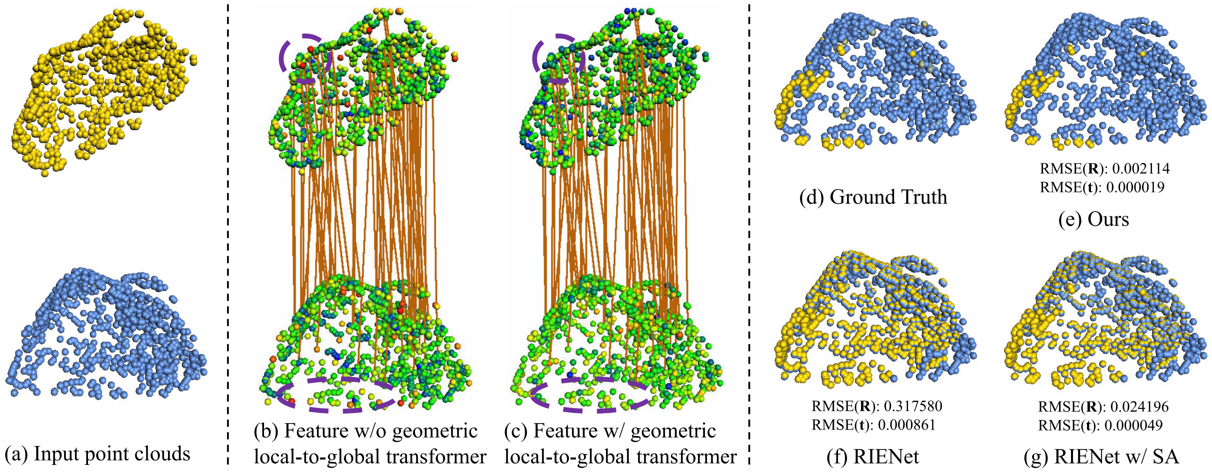
We first design a local-to-global transformer module with a transformation-invariant geometry embedding to enrich the feature representation. The local transformer module adaptively aggregates the local features from the neighborhood of each point, which serves as a remedy for the loss of local cues. In the global transformer module, we utilize the self-attention [18] to capture the long-range dependence within each point cloud and use cross-attention to facilitate the feature interaction between point cloud pairs. In order to make the features more distinctive, we design a transformation-invariant

Manuscript received 29 March 2024; revised 18 June 2024; accepted 11 July 2024. Date of publication 29 July 2024; date of current version 12 August 2024. This work was supported in part by the Research Grants Council of the Hong Kong Special Administrative Region, China, under Grant UGC/FDS16/E14/21; in part by the National Natural Science Foundation of China under Grant T2322012, Grant 62172218, and Grant 62032011; in part by Shenzhen Science and Technology Program under Grant JCYJ20220818103401003 and Grant JCYJ20220530172403007; and in part by Guangdong Basic and Applied Basic Research Foundation under Grant 2022A1515010170. *(Corresponding authors: Weiming Wang; Mingqiang Wei.)*

Chengyu Zheng, Mengjiao Ma, Zhilei Chen, Honghua Chen, and Mingqiang Wei are with the School of Computer Science and Technology, Nanjing University of Aeronautics and Astronautics, Nanjing 211106, China, and also with Shenzhen Research Institute, Nanjing University of Aeronautics and Astronautics, Shenzhen 518000, China (e-mail: zhengcyluda@gmail.com; mengjiao@nuaa.edu.cn; zhil_chen@outlook.com; chenhonghuacn@gmail.com; mingqiang.wei@gmail.com).

Weiming Wang is with the School of Science and Technology, Hong Kong Metropolitan University, Hong Kong SAR (e-mail: wmwang@hkmu.edu.hk).

Digital Object Identifier 10.1109/TGRS.2024.3434436



(a) Input point clouds (b) Feature w/o geometric local-to-global transformer (c) Feature w/ geometric local-to-global transformer (d) Ground Truth (e) Ours (f) RIENet (g) RIENet w/ SA

Fig. 1. We propose an effective unsupervised RegiFormer for point cloud registration. (a) Two point clouds for registration (yellow: source; blue: target). (b) and (c) Feature visualization of our RegiFormer without and with the GLGT. We colorize the points with the (blue, green, yellow, red) color scale according to feature values and draw the ground-truth correspondence in brown. Note that there are many outlier points with high feature values in the purple oval regions of (b), which misleads the correspondence search. By contrast, the GLGT makes the features in (c) more reliable to reduce the mismatches effectively. (d)–(g) Registration results of the ground truth, our RegiFormer, RIENet [1], and RIENet with the SA strategy, respectively.

geometry embedding and inject it into the local-to-global transformer. With the merits of affluent and reliable features [see Fig. 1(c)], it is easy to obtain pseudocorrespondences through feature similarity matching. However, point clouds tend to be partially overlapping in most cases, and there are definitely mismatches in the pseudocorrespondences. Therefore, we feed the coordinates of the pseudocorrespondences into the confidence prediction module to further compute confidence scores, which are subsequently used as weights for estimating the final transformation.

In addition, based on the observation that neural networks can align completely overlapping pairwise point clouds more easily than partially overlapping, we devise an SA strategy (without using any GT transformations) to enhance the learning alignment ability of networks when trained without ground-truth transformation labels. As shown in the right-hand side of Fig. 1, with the proposed SA strategy, the network can learn the 3-D alignment knowledge more easily and estimate the transformation more accurately.

Comprehensive experiments show that RegiFormer achieves state-of-the-art performance among unsupervised methods and even surpasses some supervised methods on both synthetic and real-scanned data. Our main contributions are threefold.

1) We propose a novel unsupervised method for point cloud registration, which can handle the partial overlap scenes with high registration accuracy and serve as a robust baseline, facilitating further advancements in the field.
2) We design a GLGT to obtain sufficient and distinctive feature representation, which not only aggregates the local features from the neighborhood but also excavates the global intrarelationship in each point cloud and enhances the cross-interrelationship between paired point clouds with geometry priors.
3) We devise a plug-and-play SA strategy to enable learning 3-D alignment knowledge, which improves the unsupervised registration performance.

This article is organized as follows. Section II provides a concise review of both traditional and learning-based registration methods. Section III presents a detailed description of the proposed method. Section IV presents the results and discussions. Section V discusses the limitations and failure cases of our method under specific conditions, followed by the conclusion in Section VI.

II. Related Work

A. Traditional Registration Methods

Iterative closet point (ICP) [5] is a classical point cloud registration method, which iteratively searches correspondences and estimates transformation to minimize the error between corresponding points. Several ICP variants modify the criterion of correspondence search from point-to-point to point-to-plane [19] and plane-to-plane [20], which improves the robustness of ICP. However, ICP and its variants heavily depend on a good initialization and easily fall into a local minimum. To solve this problem, Go-ICP [6] leverages a branch-and-bound scheme [21] to guarantee global optimality, but it is time-consuming to search the entire 3-D motion space. Besides, FPFH [12] proposes a fast point feature histogram descriptor and uses RANSAC [22] to estimate transformation robustly. Teaser [23] proposes a fast and certifiable algorithm for the registration of two point sets in the presence of large amounts of outlier correspondences. Chen et al. [24] design a novel plane-/line-based descriptor specifically for establishing structure-level correspondences between point clouds. Deng et al. [25] design a novel metric based on the intersection points between two shapes and a random straight line to conquer the instability of the closest-point criterion, which does not assume a specific correspondence. GFOICP [26] statistically selects registration points by the cross-entropy of geometric features of the points, then matches correspondences based on a variable distance threshold, and filters out correct correspondences using an iterative strict constraint on geometric feature similarity. Nevertheless, the traditional registration methods require complicated optimization strategies and may fail in complex scenes.

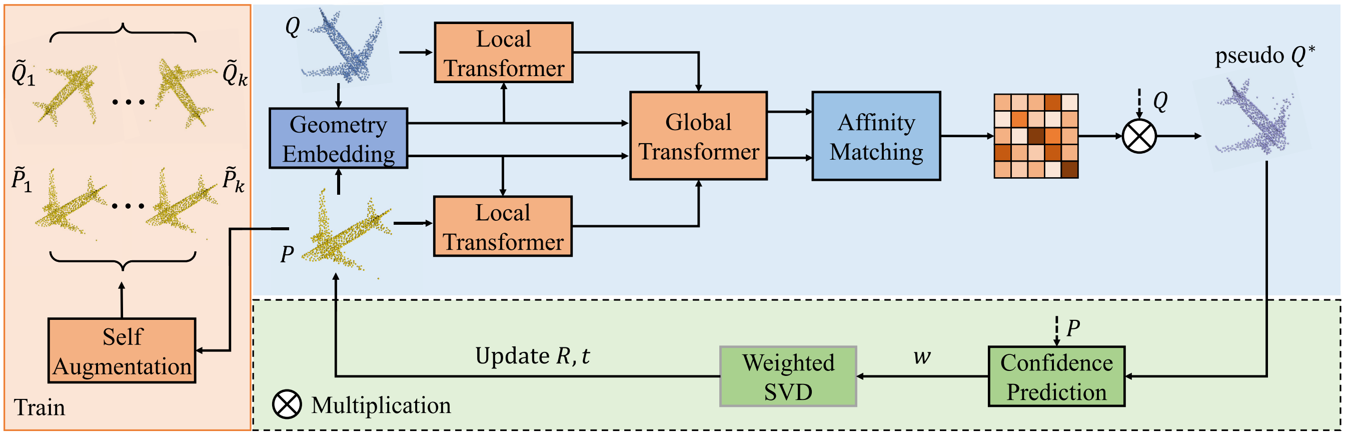


Fig. 2. Architecture of our unsupervised RegiFormer. RegiFormer mainly consists of four components: local-to-global transformer, affinity matching, confidence prediction, and SA. Note that the SA module is only utilized when training.

B. Supervised Registration Methods

Many supervised point cloud registration methods are proposed with the rapid development of deep learning techniques. PointNetLK [27] combines PointNet [28] and the Lucas & Kanade optimization algorithm to align point clouds. Following the ICP pipeline [5], DCP [7] utilizes an attention-based module to approximate matching and a differentiable singular value decomposition (SVD) layer to obtain the final rigid transformation. To tackle the partial registration problem, PRNet [29] introduces a keypoint detector to search keypoint-to-keypoint correspondences for alignment. IDAM [13] proposes an iterative distance-aware similarity matrix convolution module, which incorporates information from both the feature and Euclidean space into point matching process. RPM-Net [30] uses the differentiable Sinkhorn layer and annealing algorithm to get soft correspondences from hybrid features. RGM [31] transforms point clouds into graphs and designs a module based on deep graph matching to calculate a soft correspondence matrix. DeepGMR [32] represents the input point clouds using Gaussian mixture models and formulates registration as the minimization of KL-divergence between two probability distributions. Besides, several works [33], [34], [35], [36], [37], [38], [39] strive to remove outlier correspondences, which tackle the registration problem from another perspective. DGR [33] leverages a fully convolutional network for correspondence confidence prediction and employs a differentiable weighted Procrustes algorithm for closed-form pose estimation. PointDSC [35] incorporates spatial consistency into spectral matching for pruning outlier correspondences. In addition, Predator [14] designs an overlap attention module to align low-overlap point clouds. Chen et al. [40] solves the low-overlap registration problem by introducing a misaligned image between paired point clouds to obtain cross-modality features, which are used for two-stage overlap point classification. REGTR [8] utilizes attention mechanisms to replace explicit feature matching and directly predicts the final set of correspondences. GeoTransformer [41] encodes pairwise distances and tripletwise angles into the transformer, which is robust in low-overlap cases. GLORN [42] introduces a rotation-invariant full convolutional network searching for super points located in the overlapping region and generating feature descriptors at the super points simultaneously. Incorporating contour cues to enhance the point cloud registration task, Ma and Wei [43] propose a novel sketch-based framework for point cloud registration that incorporates contour cues to enhance the point cloud registration task. PointDifformer [44] utilizes graph neural partial differential equations (PDEs) and heat kernel signatures to extract high-dimensional features from point clouds by aggregating information from the 3-D neighborhood of points, thus enhancing the robustness of the feature representations. HECPG [45] leverages hyperbolic information to enhance feature representations and suppresses the effects of nonoverlapping regions through confidence guidance. Although supervised point cloud registration methods achieve profound progress, they have to consume huge data with ground-truth transformation labels for training, which greatly increases the training cost and hinders their applications in the real world.

C. Unsupervised Registration Methods

Since the cost of annotation for ground-truth transformation labels is expensive, U-PCR has attracted more research attention. FMR [15] proposes a feature-metric projection error to optimize registration, which is fast and does not search the correspondences. CEMNet [16] models the point cloud registration task as a Markov decision process and designs a sampling network module to generate a prior sampling distribution of transformation as a good initialization for the differentiable CEM module. Jiang et al. [46] leverage reinforcement learning to deal with registration by developing a latent dynamic model for point clouds. Li et al. [17] jointly handle shape completion and registration by a learnable latent code in an unsupervised manner. Besides, several methods [47], [48] focus on designing effective feature descriptors for registration by unsupervised learning. RIENet [1] proposes a reliable inlier evaluation method to improve the accuracy of alignment, while it only takes the local neighborhood information into consideration.

By contrast, our method devises a GLGT that considers the local, global, and cross-feature representation simultaneously with geometry priors. Besides, we rethink the learning alignment ability of U-PCR methods and design an SA strategy to promote registration performance.

III. Method

A. Problem Formulation and Overview

Point cloud registration aims at estimating a rigid transformation $\mathbf{T} = \{\mathbf{R}, \mathbf{t}\}$, where $\mathbf{R} \in \mathbf{SO}(3)$ and $\mathbf{t} \in \mathbb{R}^3$, to align a source point cloud $\mathbf{P} = \{\mathbf{p}_i \in \mathbb{R}^3 \mid i = 1, \ldots, N\}$ and a target point cloud $\mathbf{Q} = \{\mathbf{q}_j \in \mathbb{R}^3 \mid j = 1, \ldots, M\}$. Given the correspondences $\mathcal{C}$ between $\mathbf{P}$ and $\mathbf{Q}$, $\mathbf{T}$ can be solved by

$$\min_{\mathbf{R},\mathbf{t}} \sum_{(\mathbf{p}_i,\mathbf{q}_j)\in\mathcal{C}} \|\mathbf{R}\cdot\mathbf{p}_i + \mathbf{t} - \mathbf{q}_j\|_2^2. \tag{1}$$

Obtaining the ground-truth correspondences $\mathcal{C}$ is nontrivial in an unsupervised setting. Instead, we first generate the pseudocorrespondences $\mathcal{C}^*$ and then predict the confidence scores w as weights for the final transformation estimation. Therefore, we reformulate the registration problem as

$$\min_{\mathbf{R},\mathbf{t}} \sum_{(\mathbf{p}_i,\mathbf{q}_i^*)\in\mathcal{C}^*} w_i \|\mathbf{R}\cdot\mathbf{p}_i + \mathbf{t} - \mathbf{q}_i^*\|_2^2 \tag{2}$$

where w_i is the confidence score of the ith correspondence. The pseudotarget point cloud $\mathbf{Q}^* = \{\mathbf{q}_\mathbf{i}^* \in \mathbb{R}^3 \mid i = 1, \ldots, N\}$ builds the pseudocorrespondence set $\mathcal{C}^* = \{(\mathbf{p}_i, \mathbf{q}_i^*) \mid i = 1, \ldots, N\}$ with $\mathbf{P}$.

As illustrated in Fig. 2, our method mainly consists of four components: GLGT, affinity matching, confidence prediction, and SA. First, we design a local transformer module to extract distinctive features from the neighborhood adaptively. Then, a global transformer module is leveraged to capture the intrarelationship within each point cloud and enhance the interrelationship between paired point clouds. In addition, we devise a transformation-invariant geometry embedding, which is injected into the local-to-global transformer to make features more distinctive. With sufficient and reliable features, we utilize an affinity matching module to calculate the feature similarity matrix, which generates the pseudotarget point cloud and pseudocorrespondences. Considering that there may be some wrong pairs in pseudocorrespondences due to partial visibility, a confidence prediction module is employed to estimate confidence scores that serve as weights to obtain the final transformation. Besides, we propose a plug-and-play SA strategy to produce additional simple training samples when training, which improves the learning alignment ability of neural networks and increases registration accuracy.

B. Geometric Local-to-Global Transformer

1) Geometric Relation Embedding: To make features extracted by the transformer more distinctive, we introduce a novel geometric relation embedding to encode the transformation-invariant geometry structure of the points (see Fig. 3). The core idea of the geometric relation embedding is to leverage the pointwise distance, the point-to-plane angle, and the maximal side difference between two local triangles for relation measurement. In detail, given two points $\mathbf{p}_i$ and $\mathbf{p}_j$, their geometric relation is described as follows.

1) *Pointwise Distance:* Pointwise distance is the Euclidean distance $\rho_{i,j} = \|\mathbf{p}_i - \mathbf{p}_j\|_2$ between two points.

2) *Point-to-Plane Angle:* We select two nearest neighbors $\mathbf{p}_i^{k_1}$ and $\mathbf{p}_i^{k_2}$ of $\mathbf{p}_i$ to form a local plane. Then, we calculate the normal vector $\vec{\mathbf{n}}$ of the local plane by simple cross-product. The point-to-plane angle is computed as $\alpha_{i,j} = \angle(\vec{\mathbf{n}}, \mathbf{p}_j - \mathbf{p}_i)$.

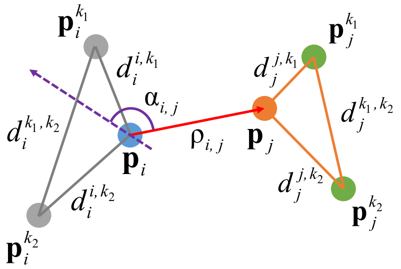


Fig. 3. Detail of the geometric relation embedding. The pointwise distance, the point-to-plane angle, and the maximal side difference between two local triangles are calculated to form the transformation-invariant geometry structure of a pair of points.

3) *Maximal Side Difference of Triangles:* Similar to $\mathbf{p}_i$, we denote the two nearest neighbors of $\mathbf{p}_j$ as $\mathbf{p}_j^{k_1}$ and $\mathbf{p}_j^{k_2}$. The maximal side difference between two local triangles is computed as $\eta_{i,j} = \max((|d_i^{i,k_1} - d_j^{j,k_1}|, |d_i^{i,k_2} - d_j^{j,k_2}|, |d_i^{k_1,k_2} - d_j^{k_1,k_2}|)$, where $d_i^{i,k_1} = \|\mathbf{p}_i - \mathbf{p}_i^{k_1}\|_2$.

Finally, the geometric relation embedding $\mathbf{g}_{i,j}$ is computed by aggregating the pointwise distance, the point-to-plane angle, and the maximal side difference between two local triangles as

$$\mathbf{g}_{i,j} = \left[\rho_{i,j}, \alpha_{i,j}, \eta_{i,j}\right] \tag{3}$$

where $[\cdot, \cdot]$ represents the concatenation operation.

2) Local Transformer Module: Most of the current unsupervised registration methods [1], [16] utilize edge convolution [49] to extract the local features from the neighborhood of each point. However, edge convolution only chooses the max value of the neighborhood as the center point's feature. This means that some equally important nonmaximum features will be discarded in each dimension. Thus, the local structure of a neighborhood is potentially not fully explored. Inspired by vector attention [50], we design a local transformer module to adaptively aggregate the local distinctive features from the neighborhood, which is shown in Fig. 4.

Given a point cloud, we first search the k nearest neighbors for each point and use the difference between the center point p_i and its neighbor p_j as initial edge features $e_{i,j}^0 = p_i - p_j$. Then, we leverage four hierarchical layers with local attention to extract deep features as

$$\mathbf{f}_i^l = \sum_{\mathbf{e}_{i,j}^l \in \mathcal{E}^l(i)} \phi\left(\beta\left(\rho\left(\mathbf{e}_{i,j}^l\right) + \eta\left(\mathbf{g}_{i,j}\right)\right)\right) \odot \alpha\left(\mathbf{e}_{i,j}^l\right) \tag{4}$$

where $\mathcal{E}^l(i), l = 1, 2, 3, 4$, denotes the set of edge features for the ith point in the lth layer and $\odot$ is the Hadamard product operation. $\rho(\cdot)$ and $\eta(\cdot)$ are the MLP layers to project edge features and geometric relation embedding into attention scores, respectively. $\beta(\cdot)$ is the MLP layer to fuse two kinds of attention scores, and $\alpha(\cdot)$ is another MLP layer that transfers edge features to node features. $\phi(\cdot)$ denotes the softmax function. Note that we update the lth edge feature as $\mathbf{e}_{i,j}^l = \text{MLP}(\mathbf{e}_{i,j}^{l-1})$. With the four hierarchical features, we can obtain the final local features by

$$\mathbf{f}_i = \gamma\left(\left[\mathbf{f}_i^1, \mathbf{f}_i^2, \mathbf{f}_i^3, \mathbf{f}_i^4\right]\right) \tag{5}$$

where $\gamma(\cdot)$ is an MLP layer and $[\cdot, \cdot]$ is the concatenation.

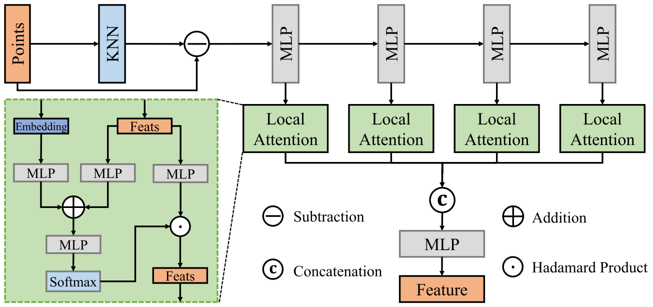


Fig. 4. Illustration of the local transformer module.

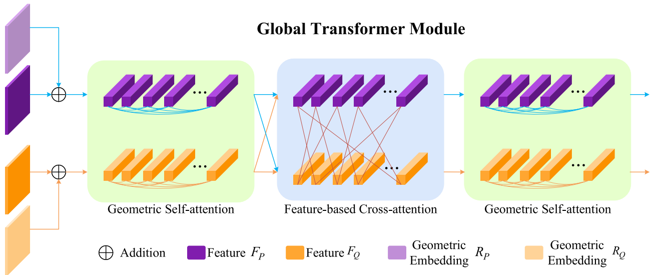

Global Transformer Module
Geometric Self-attention
Feature-based Cross-attention
Geometric Self-attention
Addition
Feature F_P
Feature F_Q
Geometric Embedding R_P
Geometric Embedding R_Q

Fig. 5. Illustration of the global transformer module.

3) Global Transformer Module: Although distinctive, the local features are still insufficient due to their limited receptive fields. We observe that the global contextual features and even the cross-hybrid features of two point clouds registered are often required for the task of point cloud registration. First, there e exist similar structures in a point cloud, which may help each other to learn a better feature. Second, feature interaction between the source and target point clouds plays a vital role in gaining knowledge about potential inlier correspondences. To this end, we introduce Transformer [18] to capture long-range dependencies within a single point cloud and enhance the cross-feature fusion between paired point clouds for unsupervised registration. The global transformer consists of two self-attention blocks and one cross-attention block between them (see Fig. 5). Unlike other transformer-based methods [8], [14], we inject the novel geometric relation embedding into self-attention blocks to reduce mismatches from feature matching.

Self-Attention: The self-attention block is utilized to explore the intrarelationship in both feature and geometry spaces for each point cloud. We take the computation for **P** as an example (the same for **Q**). Given the features $\mathbf{F} \in \mathbb{R}^{N\times d}$, where d is the dimension of features, the output features $\mathbf{Z} \in \mathbb{R}^{N\times d}$ are the weighted sum of all projected input features as

$$\mathbf{z}_i = \sum_{j=1}^{N} a_{i,j}\left(\mathbf{f}_j \mathbf{W}^V\right) \tag{6}$$

where $a_{i,j}$ is the weight and computed by a rowwise softmax on the attention scores $r_{i,j}$, which is computed as

$$e_{i,j} = \frac{\left(\mathbf{f}_i \mathbf{W}^Q\right)\left(\mathbf{f}_j \mathbf{W}^K\right)^{\mathrm{T}} + \mathbf{g}_{i,j}\mathbf{W}^G}{\sqrt{d_t}} \tag{7}$$

where $\mathbf{g}_{i,j}$ is the geometric relation embedding, and $\mathbf{W}^Q$, $\mathbf{W}^K$, and $\mathbf{W}^V \in \mathbb{R}^{d_t\times d_t}$ are the projection weights for queries, keys, and values. $\mathbf{W}^G \in \mathbb{R}^{3\times 1}$ is the projection weights for geometric relation embedding. Fig. 6 shows the computation of self-attention with geometric relation embedding.

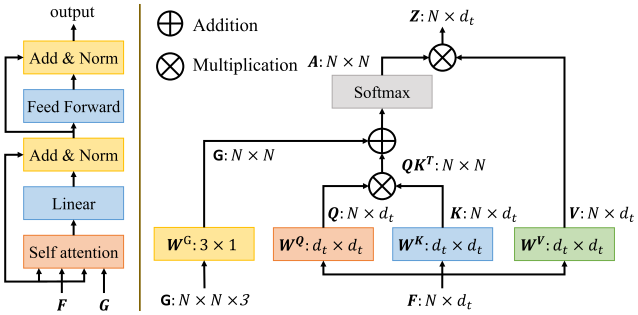


Fig. 6. Flow of self-attention with geometric relation embedding.

Cross-Attention: The cross-attention can promote the interrelationship between two point clouds. Given the features $\mathbf{F}^P$, $\mathbf{F}^Q$ of **P** and **Q**, the cross-attention features $\mathbf{Z}^P$ of **P** is computed as

$$\mathbf{z}_i^P = \sum_{j=1}^{M} a_{i,j}\left(\mathbf{f}_j^Q \mathbf{W}^V\right). \tag{8}$$

Similarly, $a_{i,j}$ is computed by a rowwise softmax on the cross-attention scores $r_{i,j}$ as

$$r_{i,j} = \frac{\left(\mathbf{f}_i^P \mathbf{W}^Q\right)\left(\mathbf{f}_j^Q \mathbf{W}^K\right)^{\mathrm{T}}}{\sqrt{d}}. \tag{9}$$

The cross-features $\mathbf{Z}^Q$ are obtained in the reverse direction. Thanks to the GLGT, we obtain not only the local distinctive features but the global contextual and cross-hybrid features $\hat{\mathbf{Z}}^P$ and $\hat{\mathbf{Z}}^Q$ with geometry priors, improving the accuracy of later feature similarity matching.

C. Affinity Matching

With the merits of reliable and abundant features, an affinity matching module [31] is leveraged to generate the feature similarity matrix. We first calculate the similarity matrix of features $\hat{\mathbf{Z}}^P$ and $\hat{\mathbf{Z}}^Q$ in a learnable way instead of simple dot-product of feature vectors as

$$\mathbf{A}_{i,j} = \left(\hat{\mathbf{z}}_i{}^P\right)\mathbf{W}\left(\hat{\mathbf{z}}_j{}^Q\right)^{\mathrm{T}} \tag{10}$$

where $\mathbf{W} \in \mathbb{R}^{d\times d}$ is a learnable parameter. To handle outliers, we utilize the Sinkhorn algorithm [51] to calculate the soft assignments $S \in \mathbb{R}^{N\times M}$, and then, the pseudotarget point cloud $\mathbf{Q}^* = \{\mathbf{q}_\mathbf{i}^* \in \mathbb{R}^3 \mid i = 1, \ldots, N\}$ is obtained by

$$q_i^* = \frac{1}{\sum_j^M \exp\left(s_{ij}\right)} \sum_j^M \exp\left(s_{ij}\right)\cdot \mathbf{q}_j \tag{11}$$

where $\exp(\cdot)$ is the exponential function. We generate the pseudocorrespondences $\mathcal{C}^* = \{(\mathbf{p}_i, \mathbf{q}_i^*) \mid i = 1, \ldots, N\}$ from the source point cloud **P** and the pseudotarget point cloud $\mathbf{Q}^*$ for alignment.

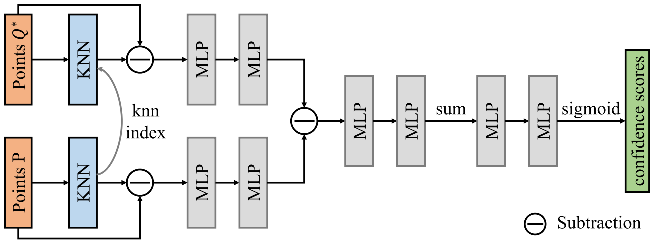


Fig. 7. Details of the confidence prediction module.

D. Confidence Prediction

In most cases, paired point clouds are often partially overlapped, i.e., there are no one-to-one correspondences. Thus, we need to distinguish the wrong pairs in the pseudocorrespondences for accurate registration. Like [1], we introduce a confidence prediction module to calculate confidence scores for correspondences. We aim to exploit the geometric difference between the source neighborhood and the corresponding pseudotarget neighborhood for predicting pointwise confidence based on the learned correspondences $\{(p_i, q_i^*)\}$.

We use the confidence prediction module to adaptively capture the structurewise difference between the neighborhoods $\mathcal{N}_{p_i}$ and $\mathcal{N}_{q_i^*}$ by constructing a learnable graph representation on the neighborhood. As illustrated in Fig. 7, this module takes a source point cloud $\mathbf{P}$ and a pseudotarget point cloud $\mathbf{Q}^*$ as inputs and outputs the confidence scores w. We first search the k neighbors for $\mathbf{P}$ to obtain edge features $e_{i,j}^{\mathbf{P}}$ like the local transformer module, and then, the KNN index of $\mathbf{P}$ is shared to gain $e_{i,j}^{\mathbf{Q}^*}$ for $\mathbf{Q}^*$, which guarantees that the neighborhood points of $\mathbf{P}$ and $\mathbf{Q}^*$ are consistent. We utilize two MLP layers to embed points into high-dimensional features and use a simple subtraction operation to highlight differences. With the fused features, another MLP layer and a summation operation are leveraged to obtain correspondence features, which are finally projected into the confidence scores. The whole module can be described as

$$w_i = \operatorname{sigmoid}\left(\eta\left(\sum_{j=1}^{k}\phi\left(\varphi\left(e_{i,j}^{\mathbf{P}}\right)-\varphi\left(e_{i,j}^{\mathbf{Q}^*}\right)\right)\right)\right) \tag{12}$$

where $\eta(\cdot)$, $\phi(\cdot)$, and $\varphi(\cdot)$ are two MLP layers with different parameters and sigmoid$(\cdot)$ is the sigmoid function.

With the pseudocorrespondences set $\mathcal{C}^* = \{(\mathbf{p}_i, \mathbf{q}_i^*) \mid i = 1, \ldots, N\}$ and the confidence scores w, we can estimate the final transformation by a weighted SVD layer according to (2).

E. Self-Augmentation

Considering the weak learning ability of unsupervised registration methods, we propose a plug-and-play SA strategy to assist the training process. Naturally, we can add more paired samples by applying random rotations and translations to one of the input point clouds when training, which is simple but effective for unsupervised registration. In particular, we copy the source point cloud $\mathbf{P}$ as $\tilde{\mathbf{P}}$ and randomly rotate and translate it to obtain $\tilde{\mathbf{Q}}$. We repeat this process I times to get self-augmented training samples. Note that the SA strategy is different from the data processing in [29] and [30], as they just simulate partially overlapped point pairs with GT transformations for training on synthetic data, while ours aims to learn from self-transformations to grasp 3-D alignment knowledge for unsupervised methods (also effective on real-scanned data). The SA strategy only leverages the basic rotation and translation operation, which does not carry out any partial cropping for two reasons: 1) we observe that registration methods align paired point clouds of complete overlap more easily than that of partial overlap and 2) completely overlapped point cloud pairs are more generic because different training data may have different patterns of partial overlap. In addition, several methods [52], [53] utilize data augmentation such as random transformation to generate positive samples for representation learning, while our SA strategy is designed to enhance the ability of unsupervised registration methods to learn alignment knowledge.

F. Loss Function

In an unsupervised manner, we iteratively train our approach with three loss terms like in [1] as

$$\mathcal{L} = \mathcal{L}_a + \mathcal{L}_k + \mathcal{L}_n. \tag{13}$$

1) Alignment Loss: Most unsupervised methods [15], [17] use the Chamfer distance loss [54] for optimization, while this loss is sensitive to outliers. To tackle this problem, we utilize the robust Huber function as

$$\ell_\beta(u) = \begin{cases} \dfrac{1}{2}u^2, & \text{if } |u| \leq \beta \\ \beta\left(|u| - \dfrac{1}{2}\beta\right), & \text{otherwise} \end{cases} \tag{14}$$

where β is the hyperparameter that controls the range of the inlier. With the transformed source point cloud $\overline{\mathbf{P}}$ by the estimated transformation, we describe the alignment loss (AL) as

$$\begin{aligned} \mathcal{L}_a\left(\overline{\mathbf{P}}, \mathbf{Q}\right) = &\sum_{\overline{\mathbf{p}}\in\overline{\mathbf{P}}} \ell_\beta\left(\min_{\mathbf{q}\in\mathbf{Q}} \|\overline{\mathbf{p}} - \mathbf{q}\|_2^2\right) \\ &+ \sum_{\mathbf{q}\in\mathbf{Q}} \ell_\beta\left(\min_{\overline{\mathbf{p}}\in\overline{\mathbf{P}}} \|\mathbf{q} - \overline{\mathbf{p}}\|_2^2\right). \end{aligned} \tag{15}$$

2) Keypoint Loss: To guide the learning of the confidence prediction module, we choose pseudocorrespondences with the g highest confidence scores w. We denote the chosen points of the source point cloud and the pseudotarget point cloud as $\mathbf{X}$ and $\mathbf{Y}$, respectively. $\overline{\mathbf{X}}$ represents the transformed $\mathbf{X}$ by the estimated transformation. Then, the keypoint loss (KL) is defined as

$$\mathcal{L}_k = \sum_{\overline{\mathbf{x}}_i \in \overline{\mathbf{X}}, \mathbf{y}_i \in \mathbf{Y}} \|\mathbf{R} \cdot \overline{\mathbf{x}}_i + \mathbf{t} - \mathbf{y}_i\|_2. \tag{16}$$

3) Neighborhood Loss: In the confidence prediction module, we utilize neighborhood information to improve the reliability of confidence scores. Since the KNN search of the source point cloud $\mathbf{P}$ and the pseudotarget point cloud $\mathbf{Q}^*$ share the same KNN index, neighbors of $\overline{\mathbf{X}}$ and $\mathbf{Y}$ are consistent and ordered. Thus, we calculate the relative coordinates of neighbors and formulate the neighborhood loss (NL) as

$$\mathcal{L}_n = \sum_{\substack{\bar{\mathbf{x}}_i \in \bar{\mathbf{X}},\\ \mathbf{y}_i \in \mathbf{Y}}} \sum_{\substack{\mathbf{p}_j \in \mathcal{N}(\bar{\mathbf{x}}_i),\\ \mathbf{q}_j \in \mathcal{N}(\mathbf{y}_i)}} \left\| \mathbf{R} \cdot \mathbf{p}_j + \mathbf{t} - \mathbf{q}_j \right\|_2. \tag{17}$$

Here, $\mathcal{N}(\cdot)$ denotes the nearest neighborhood search, and $\mathbf{p}_j$ and $\mathbf{q}_j$ are the relative coordinates of neighborhood points.

IV. Experiment

Datasets: We evaluate our approach on five datasets: ModelNet [55] (synthetic object data), ScanObjectNN [56] (real object data), ICL-NUIM [57] (synthetic scene data), 7Scenes [58] (real scene data), and KITTI odometry datasets [59].

1) ModelNet is generated from ModelNet40 [55], which contains point clouds sampled from 12 311 CAD models of 40 different categories. We sample 1024 points for each point cloud and transform it by a random rotation in the range of $[0, 45]^\circ$ and a random translation in the range of $[-0.5, 0.5]$ along each axis to obtain the paired point cloud. To simulate the partial-overlap condition, we adopt the same cropping mode as [29]. Finally, each point cloud is shuffled to reorder all points. We apply three different settings on ModelNet for comprehensive analysis.

a) *Seen setting:* We follow the official training split (9840 samples) and testing split (2468 samples), both containing all categories.

b) *Unseen setting:* To evaluate the generalization ability of our method to different shapes, we use the first 20 categories for training (5112 samples) and the last 20 categories for testing (1266 samples).

c) *Gaussian noise outliers' setting:* We modify the testing part of the seen setting by randomly copying 25% points of each point cloud and adding Gaussian noise to them, sampled from $\mathcal{N}(0, 0.5)$ and clipped to $[-1.0, 1.0]$ as [1]. We train all approaches under the seen setting and test them on the modified testing part to measure the robustness of outliers.

2) ScanObjectNN [56] is a real-world dataset based on scanned indoor object data, which contains 2309 training samples and 581 testing samples. We apply the same data preprocessing on ScanObjectNN as ModelNet to generate partial-overlap point cloud pairs for alignment.

3) ICL-NUIM [57] is a synthetic indoor scene dataset. We resample the source point clouds to 2048 points, operate the rigid transformation on them for the target point clouds, and then downsample the point clouds to 1536 points like in [1] to generate the partial data. The ICL-NUIM dataset is split into 1278 samples for training and 200 samples for testing.

4) 7Scenes [58] is a widely used benchmark registration dataset of seven indoor scenes, including Chess, Fires, Heads, Office, Pumpkin, RedKitchen, and Stairs, which is divided into 296 samples for training and 57 samples for testing. The data preprocessing is the same as the ICL-NUIM dataset.

5) The KITTI odometry dataset [59] consists of 11 sequences of outdoor driving scenarios scanned by LiDAR. We use Sequence 00-05 for training, 06-07 for validation, and 08-10 for testing. The GT poses are refined with ICP, and we only use point cloud pairs that are at least 10 m away for evaluation.

Evaluation Metrics: Following the prior work DCP [7], we use the anisotropic metrics of mean absolute error (MAE) over Euler angles and translations. In addition, we evaluate the mean isotropic error (MIE) for rotation and translation proposed in [30]. All angles are in degrees.

Comparison: We compare RegiFormer with seven registration methods, including traditional methods, ICP [5] and FPFH + RANSAC [12], earlier supervised methods, IDAM [13] and RPM-Net [30], and unsupervised methods, FMR [15], CEMNet [16], RIENet [1], and IFNet [60]. In addition, to evaluate the performance of the proposed method more comprehensively, we compared it with four additional supervised methods in the last part of the experiment, including Geotransformer [41] and BUFFER [61]. For all comparison methods, we use their released code and follow the same settings to retrain them.

Implementation Details: All experiments are conducted on a single Nvidia RTX 2080Ti. We train RegiFormer for 75 epochs with a batch size of 4. The Adam optimizer is used with an initial learning rate of 0.001. The number I of the SA strategy is set to 1, and the hyperparameter β for the Huber function is 0.01. The number k of KNN search and g of KL are set to 5 and 256, respectively. We update the estimated transformation iteratively three times during both training and testing.

A. Evaluation on Synthetic ModelNet

1) Seen Setting: We quantitatively evaluate the registration performance of several methods and our method in the seen setting, which is reported on the left-hand side of Table I. It is obvious that RegiFormer outperforms the traditional, state-of-the-art unsupervised, and even some early supervised methods on all four metrics. Compared with RIENet [1] on the MAE of rotation and translation, our method has a 66.7% and 57.9% improvement, respectively.

2) Unseen Setting: From the middle side of Table I, one can see that our method achieves the lowest error among all comparison methods. Compared with the seen setting, the performance of RegiFormer only decreases slightly, demonstrating that our approach is insensitive to different shapes. The visual comparison can be found in Fig. 8(b).

3) Gaussian Noise Outliers' Setting: To evaluate the robustness of outliers, we train all methods in the seen setting and test them under the Gaussian noise outliers' setting. In the right-hand side of Table I, although all methods have different degrees of degradation, RegiFormer obtains the best performance, benefiting from the reliable and sufficient feature representation produced by the GLGT module. Most methods fail to achieve satisfying registration results when there exist many disturbing outlier points, except our method [see Fig. 8(c)].

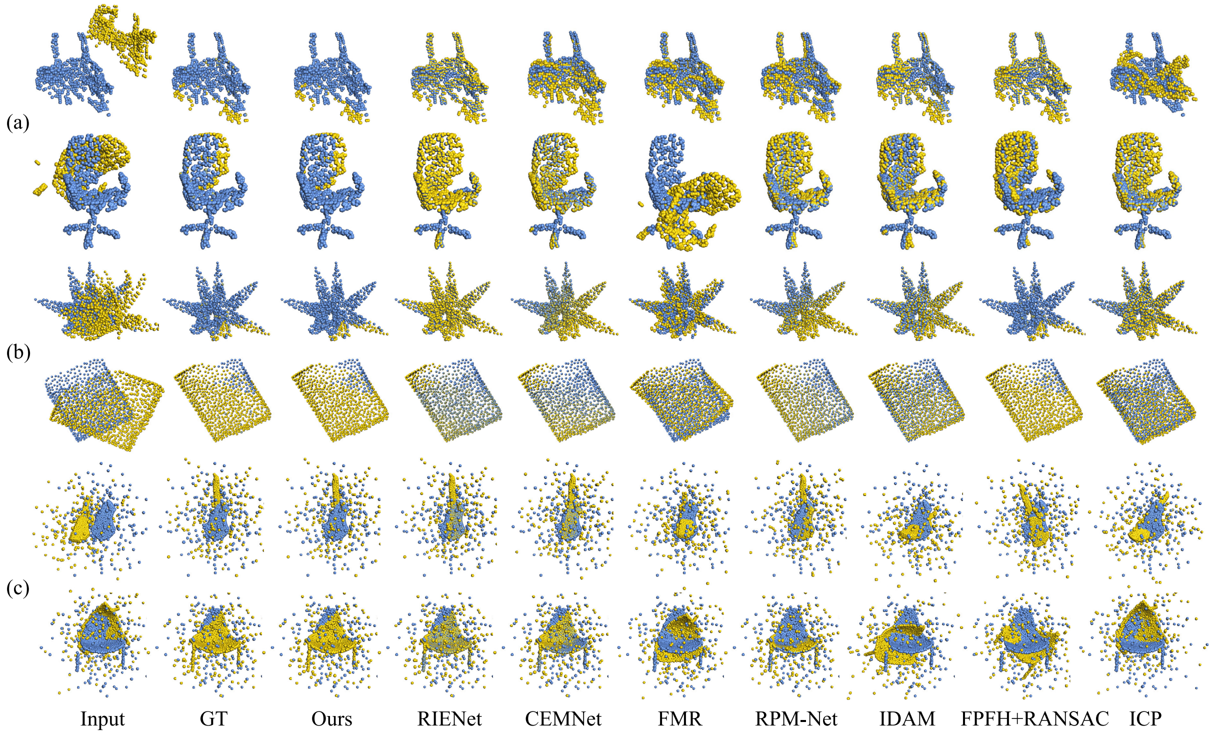


Fig. 8. Visual comparison of seven methods and our method on (a) ScanObjectNN, (b) ModelNet with an unseen setting, and (c) ModelNet with Gaussian noise outliers' setting. We colorize the source and target point clouds in yellow and blue, respectively.

TABLE I

QUANTITATIVE REGISTRATION RESULTS OF DIFFERENT METHODS ON MODELNET. (⋆), (○), AND (◇) REPRESENT THE TRADITIONAL, SUPERVISED, AND UNSUPERVISED METHODS, RESPECTIVELY. THE BEST PERFORMANCE IS HIGHLIGHTED IN **BOLD**

Model	ModelNet Seen				ModelNet Unseen				ModelNet Outlier			
	MAE(**R**)	MAR(**t**)	MIE(**R**)	MIE(**t**)	MAE(**R**)	MAR(**t**)	MIE(**R**)	MIE(**t**)	MAE(**R**)	MAR(**t**)	MIE(**R**)	MIE(**t**)
ICP (⋆)	3.301565	0.011453	6.788486	0.022761	3.439718	0.011325	6.851783	0.022392	14.53045	0.047355	27.84787	0.094257
FPFH (⋆)	0.822125	0.005310	1.548697	0.010440	0.941397	0.005212	1.947544	0.010312	14.63749	0.050137	26.95863	0.100049
IDAM (○)	0.532887	0.003063	1.058790	0.006085	0.649301	0.004138	1.265476	0.008097	22.82362	0.087075	41.40119	0.173985
RPM-Net (○)	0.015641	0.000089	0.042008	0.000173	0.028268	0.000255	0.057080	0.000505	1.151825	0.003431	2.257282	0.006792
FMR (◇)	4.200312	0.010185	8.470898	0.020054	4.267972	0.010745	8.575333	0.021401	22.59343	0.072660	45.02045	0.146379
CEMNet (◇)	0.188002	0.001850	0.348530	0.000378	0.084114	0.000118	0.155070	0.000228	0.417110	0.000479	0.642402	0.001011
RIENet (◇)	0.002286	0.000019	0.032273	0.000038	0.005858	0.000048	0.033038	0.000095	0.328612	0.000915	0.571871	0.001880
IFNet(◇)	0.002060	0.000016	0.031545	0.000032	0.004265	0.000036	0.032176	0.000072	0.239252	0.000689	0.539649	0.001421
Ours (◇)	**0.000761**	**0.000008**	**0.030507**	**0.000016**	**0.000814**	**0.000009**	**0.031823**	**0.000019**	**0.068217**	**0.000371**	**0.144074**	**0.000745**

TABLE II

QUANTITATIVE COMPARISON OF DIFFERENT METHODS ON SCANOBJECTNN. (⋆), (○), AND (◇) REPRESENT TRADITIONAL, SUPERVISED, AND UNSUPERVISED METHODS, RESPECTIVELY. THE BEST PERFORMANCE IS HIGHLIGHTED IN **BOLD**

Model	ScanObjectNN			
	MAE(**R**)	MAR(**t**)	MIE(**R**)	MIE(**t**)
ICP (⋆)	5.517674	0.011137	10.71583	0.022205
FPFH (⋆)	0.950798	0.005714	1.801037	0.011369
IDAM (○)	0.315976	0.002109	0.612576	0.004071
RPM-Net (○)	0.611338	0.002027	1.145267	0.004022
FMR (◇)	2.479400	0.010488	4.796422	0.020819
CEMNet (◇)	1.474654	0.002528	2.889640	0.004961
RIENet (◇)	0.013160	0.000081	0.042860	0.000163
IFNet(◇)	0.009581	0.000061	0.040445	0.000123
Ours (◇)	**0.001491**	**0.000021**	**0.028483**	**0.000041**

B. Evaluation on Real-Scanned ScanObjectNN

Aligning point cloud pairs from real-scanned devices is quite challenging because this kind of data often has irregular shapes and cluttered noise [see Fig. 8(a)]. As reported in the left-hand side of Table II, our method achieves better registration results than its competitors by a large margin. Qualitative comparisons are consistent with quantitative statistics, which demonstrates the great adaptability of RegiFormer to real-scanned data.

C. Evaluation on Indoor/Outdoor Sences

1) ICL-NUIM: From the middle side of Table III, we observe that our method obtains satisfactory performance on the synthetic scene data. The visual results are found in Fig. 9(a).

2) 7Scenes: Given the registration accuracy of all methods on the challenging real scene data [see the right-hand side of Table III and Fig. 9(b)], the superiority of RegiFormer is outstanding.

3) KITTI Odometry Dataset: From Table III, one can see that our model has also achieved satisfactory performance on outdoor scenes, the KITTI dataset. The translation errors of supervised RPMNet on KITTI are smaller than ours, but the runtime of our method is shorter than theirs. The visual results can be found in Fig. 9(c).

TABLE III

Quantitative Comparison of Different Methods on ICL-NUIM, 7Scenes, and KITTI Datasets. (⋆), (○), and (◇) Represent Traditional, Supervised, and Unsupervised Methods, Respectively. The Best Performance Is Highlighted in **Bold**

Model	ICL-NUIM				7Scenes				KITTI			
	MAE(**R**)	MAR(**t**)	MIE(**R**)	MIE(**t**)	MAE(**R**)	MAR(**t**)	MIE(**R**)	MIE(**t**)	MAE(**R**)	MAR(**t**)	MIE(**R**)	MIE(**t**)
ICP (⋆)	2.4022	0.0699	4.4832	0.1410	6.0091	0.0130	13.0484	0.0260	4.7433	0.9174	11.9982	2.5742
FPFH (⋆)	1.2349	0.0429	2.3167	0.0839	1.2325	0.0062	2.1875	0.0137	1.9353	0.0230	4.2766	0.0470
IDAM (○)	4.4153	0.1385	8.6178	0.2756	5.6727	0.0303	11.5949	0.0629	1.6348	0.0230	3.8151	0.0491
RPM-Net (○)	0.3267	0.0125	0.6277	0.0246	0.3885	0.0021	0.7649	0.0042	0.9164	**0.0146**	2.1291	**0.0303**
FMR (◇)	1.1085	0.0398	2.1323	0.0786	2.5438	0.0072	4.9089	0.0150	1.6786	0.0329	4.0571	0.0703
CEMNet (◇)	0.2374	0.0005	0.3987	0.0010	0.0559	0.0001	0.0772	0.0003	\	\	\	\
RIENet (◇)	0.0492	0.0023	0.0897	0.0049	0.0121	0.0001	0.0299	0.0001	0.8251	0.0183	1.8754	0.0414
IFNet(◇)	0.1563	0.0062	0.1041	0.0131	0.0089	0.0001	0.0301	0.0001	1.0182	0.0180	2.1144	0.0375
Ours (◇)	**0.0010**	**0.0001**	**0.0289**	**0.0001**	**0.0029**	**0.0001**	**0.0265**	**0.0001**	**0.7992**	0.0176	**1.8154**	0.0411

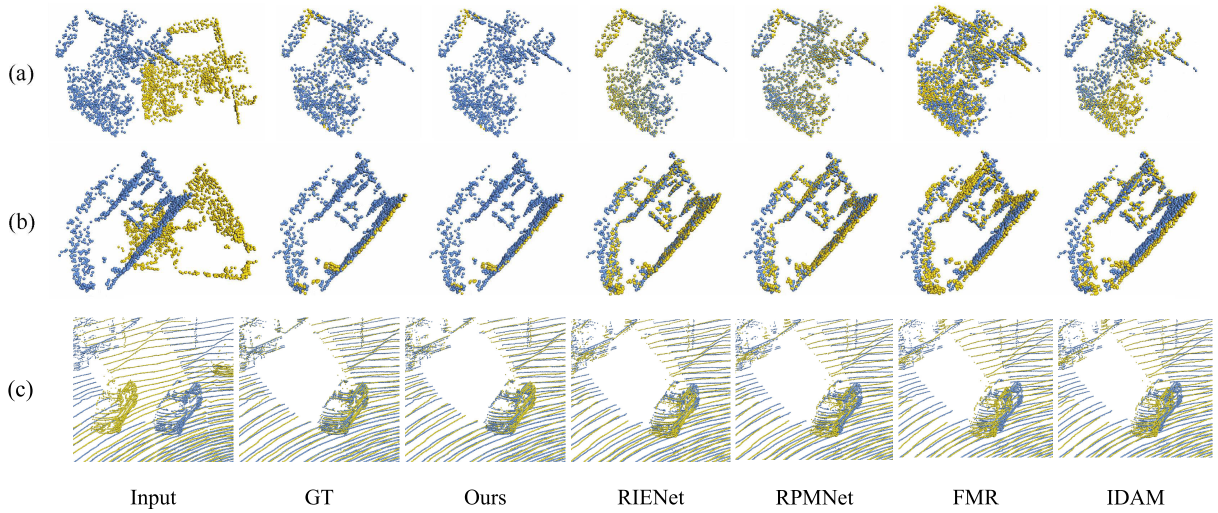


Fig. 9. Visual comparison of competitors and our approach on (a) ICL-NUIM, (b) 7Scenes, and (c) KITTI datasets. The registration results depicted in the figure only showcase a partial area within the entire scene of the KITTI dataset. We colorize the source and target point clouds in yellow and blue, respectively.

TABLE IV

Ablation Studies on ScanObjectNN. LT, GT, LGE, GGE, and SA Are Baselines With the Local Transformer Module, Global Transformer Module, Local Geometric Embedding, Global Geometric Embedding, and SA, Respectively

LT	GT	LGE	GGE	SA	ScanObjectNN			
					MAE(**R**)	MAE(**t**)	MIE(**R**)	MIE(**t**)
					0.120819	0.000618	0.286068	0.001215
√					0.004516	0.000043	0.032927	0.000084
√	√				0.002836	0.000029	0.034090	0.000057
√	√	√			0.002574	0.000023	0.032368	0.000046
√	√	√	√		0.002334	0.000021	0.031332	0.000042
√	√	√	√	√	0.001491	0.000021	0.028483	0.000041

TABLE V

Ablation Study of Loss Function on ScanObjectNN. AL, KL, and NL Are Alignment Loss, Global Keypoint Loss, and Neighborhood Loss, Respectively

AL	KL	NL	ScanObjectNN			
			MAE(**R**)	MAE(**t**)	MIE(**R**)	MIE(**t**)
√			0.063532	0.000956	0.132501	0.001925
√	√		0.012876	0.000214	0.038331	0.000325
√	√	√	0.001491	0.000021	0.028483	0.000041

TABLE VI

Error Statistics of Position Embeddings. NPE, APE, and GRE(UTOPIC) Denote Variants With No Position Embedding, Absolute Position Embedding, and Geometric Relation Embedding of UTOPIC, Respectively

Model	ScanObjectNN			
	MAE(**R**)	MAE(**t**)	MIE(**R**)	MIE(**t**)
NPE	0.002836	0.000029	0.034090	0.000057
APE	0.002928	0.000027	0.034236	0.000054
GRE(UTOPIC)	0.002804	0.000021	0.032986	0.000042
Ours	0.001491	0.000021	0.028483	0.000041

D. Analysis

To analyze the main ideas of RegiFormer, various specified experiments are carried out and reported in this section.

1) Ablation Study: The Effectiveness of Key Component: For a better understanding of our method, we conduct ablation studies on ScanObjectNN. We develop the baseline by removing the SA strategy and replacing the GLGT with DGCNN [49] in RegiFormer. Table IV reports the results of variants with: 1) the local transformer module (LT); 2) the global transformer module (GT); 3) the local geometric embedding (LGE); 4) the global geometric embedding (GGE); and 5) the SA strategy. It is clear that our full pipeline achieves the best performance on four metrics, and removing any component degrades the overall performance.

Loss Functions: We train our model with the combination of the AL, the KL, and the NL. In Table V, we train our model using the different loss functions and present the results on ScanObjectNN. One can see that with the keypoint loss and neighborhood loss, we can obtain high registration precision.

2) Different Position Embeddings: To testify to the effectiveness of the proposed geometric relation embedding, we compare four variants in Table VI: 1) no position embedding (NPE) [8]; 2) absolute position embedding (APE) [62]; 3) geometric relation embedding of UTOPIC [63]; and 4) our geometric relation embedding. We improve the geometric

TABLE VII

COMPARISON OF RPM-NET [30] AND RIENET [1] OVER THEIR VARIANTS WITH THE GLGT ON SCANOBJECTNN

Model	ScanObjectNN			
	MAE(**R**)	MAE(**t**)	MIE(**R**)	MIE(**t**)
RPM-Net	0.611338	0.002027	1.145267	0.004022
RPM-Net+GLGT	0.001049	0.000011	0.17311	0.000022
RIENet	0.013160	0.000081	0.042860	0.000163
RIENet+GLGT	0.005080	0.000035	0.030643	0.000071

TABLE VIII

COMPARISON OF FMR [15], RIENET [1], AND THEIR VARIANTS WITH THE SA STRATEGY ON SCANOBJECTNN

Model	ScanObjectNN			
	MAE(**R**)	MAE(**t**)	MIE(**R**)	MIE(**t**)
FMR	2.479400	0.010488	4.796422	0.020819
FMR+SA	2.371696	0.008081	4.723234	0.016085
RIENet	0.013160	0.000081	0.042860	0.000163
RIENet+SA	0.007052	0.000038	0.039670	0.000076

TABLE IX

REGISTRATION ERRORS OF EVERY ITERATION OF REGIFORMER VARIANTS WITH AND WITHOUT THE SA STRATEGY ON SCANOBJECTNN

Model	ScanObjectNN: RMSE(**R**)		
	iter 1	iter 2	iter 3
RegiFormer w/o SA	0.012669	0.006208	0.006183
RegiFormer	0.009399	0.003146	0.003140

embedding of UTOPIC by representing the neighborhood by the local normal vector, which is easily calculated by cross-product, and measuring the similarity between two points by the maximum difference of side between their local triangles. We observe that the absolute position embedding has no obvious advantages. By contrast, the variant equipped with our geometric relation embedding achieves the best performance.

3) Geometric Local-to-Global Transformer: To provide a better understanding of the critical role played by the proposed GLGT, we replace the original feature extraction modules of RPM-Net [30] and RIENet [1] with it. As reported in Table VII, it is obvious that GLGT improves the performance of both supervised and unsupervised registration methods by a large margin because it sufficiently extracts local distinctive, global contextual, and cross-interactive features with geometry priors. In particular, we observe that RPM-Net + GLGT has a tremendous improvement compared with its original version, indicating the potential of the proposed GLGT to promote the performance of current registration methods.

4) Effectiveness of SA Strategy: To evaluate the effectiveness of the SA strategy for unsupervised registration methods, we compare the performance of FMR [15], RIENet [1], and their variants with the proposed SA strategy on the ScanObjectNN dataset. From Table VIII, we observe that the proposed SA strategy brings a significant improvement to both two unsupervised registration methods, demonstrating its generality. In addition, we compare the registration of every iteration of RegiFormer variants with and without the SA strategy on the ScanObjectNN dataset. As reported in Table IX, it is worth noting that the SA strategy improves the performance of the initial iteration, which proves the advantage of facilitating alignment knowledge acquisition.

TABLE X

COMPARING RIENET [1] AND REGIFORMER UNDER DIFFERENT OVERLAP RATIOS

Model	Overlap	ScanObjectNN			
		MAE(**R**)	MAE(**t**)	MIE(**R**)	MIE(**t**)
RIENet	0.69	0.038879	0.000293	0.081137	0.000573
	0.58	0.127330	0.000715	0.240075	0.001414
	0.47	0.254867	0.001479	0.444799	0.002975
	0.40	0.361562	0.002448	0.664048	0.004878
	0.32	0.810180	0.005079	1.398852	0.010206
Ours	0.69	0.013009	0.000130	0.036514	0.000265
	0.58	0.034258	0.000313	0.063964	0.000643
	0.47	0.064939	0.000592	0.108498	0.001202
	0.40	0.122549	0.001213	0.221883	0.002424
	0.32	0.465570	0.003294	0.783340	0.006732

TABLE XI

QUANTITATIVE PERFORMANCE OF RPMNET [30], RIENET [1], AND OUR METHOD ON SCANOBJECTNN WITH LARGE ROTATION ANGLES IN THE RANGE OF [0, 180]°. THE BEST PERFORMANCE IS HIGHLIGHTED IN **BOLD**

Model	ScanObjectNN ([0, 180]°)			
	MAE(**R**)	MAR(**t**)	MIE(**R**)	MIE(**t**)
RPM-Net	48.80241	0.050401	75.63832	0.101040
RIENet	2.112916	0.003753	2.281461	0.007341
Ours	**0.314630**	**0.001109**	**0.578956**	**0.002072**

TABLE XII

TIMING STATISTICS (IN SECONDS) ON 7SCENES. N DENOTES THE NUMBER OF INPUT POINTS. WE REPORT THE TIME FOR ONE PAIR OF POINT CLOUDS

7Scenes (N: 1536)	ICP (⋆) 0.4582	FPFH (⋆) 0.5044	IDAM (∘) 0.3871	RPM-Net (∘) 0.8093
	FMR (⋄) 0.4286	RIENet (⋄) 0.4286	Ours (⋄) 0.4343	

5) Different Overlap Ratios: We analyze the performance of RIENet [1] and our method on the ScanObjectNN dataset when the overlap ratio decreases gradually. We use the same crop setting of PRNet [29]. The number of points is set to 768, 700, 640, 600, and 560 to generate point clouds with approximate overlap ratios of 0.69, 0.58, 0.47, 0.40, and 0.32, respectively. Table X shows the registration errors for different overlap ratios. As observed, our method is very stable until the overlap ratio decreases to 0.32, but still better than RIENet [1].

6) Performance With Large Rotation Angles: To evaluate the performance of methods at large rotation angles on the object registration data, we train RPMNet [30], RIENet [1], and our approach on the ScanObjectNN dataset with rotation angles in the range of [0, 45]° and test them at rotation angles in the range of [0, 180]°. From Table XI, the comparison methods fail to register point cloud pairs with large rotation angles. By contrast, our approach still performs well, benefiting from the transformation-invariant geometric relation embedding.

7) Running Time: Table XII reports the average running time (in seconds) of different comparison methods. The testing data are collected from the 7Scenes dataset. We conduct all experiments on a single Nvidia RTX 2080Ti with Intel Core i7-4790 @ 3.6 GHz. With the best registration accuracy, our method also runs at a comparable speed.

8) Unseen Real-Scanned Multiview Airplane Shapes With Complex and Tiny Geometry Details: We provide the visual registration result of our RegiFormer on real-scanned multiview airplane shapes with complex and tiny geometry details.

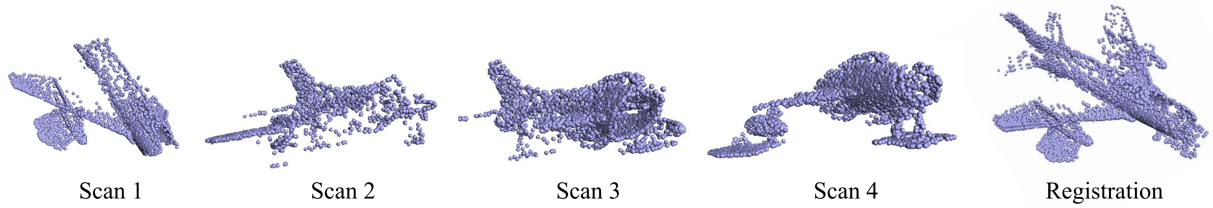


Fig. 10. Visual registration result of RegiFormer on real-scanned multiview airplane shapes with complex and tiny geometry details. Scan 1–Scan 4 denote four parts of the real complex airplane shape from different views. We register two adjacent point clouds in turn, and the final registration result can be found in the Registration part.

TABLE XIII

QUANTITATIVE RESULTS OF RPM-NET [30], GEOTRANSFORMER [41], AND OUR METHOD ON MODELNET. (○) AND (◇) REPRESENT THE SUPERVISED AND UNSUPERVISED METHODS, RESPECTIVELY. THE BEST PERFORMANCE IS HIGHLIGHTED IN **BOLD**

Model	ModelNet Seen				ModelNet Unseen				KITTI			
	MAE(**R**)	MAR(**t**)	MIE(**R**)	MIE(**t**)	MAE(**R**)	MAR(**t**)	MIE(**R**)	MIE(**t**)	MAE(**R**)	MAR(**t**)	MIE(**R**)	MIE(**t**)
RPM-Net (○)	0.015641	0.000089	0.042008	0.000173	0.028268	0.000255	0.057080	0.000505	0.9164	0.0146	2.1291	0.0303
GeoTransformer (○)	0.123463	0.000544	0.227637	0.001087	0.109644	0.000502	0.199214	0.001004	**0.1104**	**0.0123**	0.3685	0.0392
BUFFER (○)	0.045325	0.000129	0.097543	0.000237	0.147667	0.001149	0.02021	0.002298	0.1247	0.0185	**0.2674**	**0.0293**
Ours (◇)	**0.000761**	**0.000008**	**0.030507**	**0.000016**	**0.000814**	**0.000009**	**0.031823**	**0.000019**	0.7992	0.0176	1.8154	0.0411

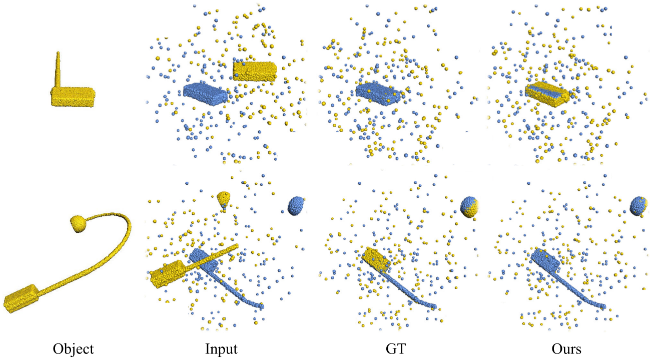


Fig. 11. Two failure examples. RegiFormer fails when: 1) the inputs lack geometrically indistinguishable features (Row 1) and 2) the point clouds have inconsistent separate parts (Row 2).

The inputs are sampled with 5000 points from the original scanned point clouds. Notably, our method is trained on the synthetic ModelNet dataset with an unseen setting. As illustrated in Fig. 10, RegiFormer registers two adjacent point clouds in turn and combines them into a complete airplane shape. The airplane shape is more geometrically complicated than the models in ModelNet40, but our method works well, demonstrating the potential of RegiFormer in the complex real-world registration application.

9) More Quantitative Results: To provide a complete comparison with existing supervised methods, we retrain GeoTransformer [41] and BUFFER [61] on ModelNet and KITTI using its released public code. Quantitative results are recorded in Table XIII. Surprisingly, we find that the performance of GeoTransformer and BUFFER is worse than that of RPM-Net [30] on the seen and unseen settings of the synthetic ModelNet dataset. This is because GeoTransformer is specifically designed for low-overlap, large-scale indoor, and outdoor scenes, and the downsampling operation is crucial for processing point clouds of large data volumes. However, downsampling may have certain side effects, such as the discarding of corresponding points, when the number of points is not very large. However, for the KITTI dataset, we only applied downsampling to our approach while using the original settings from the article for the GeoTransformer and BUFFER. The results in the table indicate that the performance of the supervised method significantly outperforms the unsupervised method.

V. FAILURE CASES AND LIMITATIONS

RegiFormer has some limitations. First, if the point cloud pairs have no distinctive geometry structures (see the first row of Fig. 11), it may fail to align them. This is because outlier points near the flat surface (without distinctive geometry structures) confuse the feature extraction, and indistinguishable features lead to mismatches. Second, it is hard to register point clouds with separate parts (see the second row of Fig. 11) because features of separate parts may be inconsistent and mislead the final registration. In the future, we intend to handle the mismatches from indistinctive and repeated structures in point clouds and pay more attention to the consistency of registration data with separate parts.

VI. CONCLUSION

We propose RegiFormer, a novel GLGT-based method with an SA strategy for U-PCR. Through the GLGT, our method extracts local distinctive, global contextual, and cross-interactive features with geometry priors. Thanks to the sufficient feature representation and confidence prediction module, RegiFormer can align point cloud pairs with high accuracy even under the noise and outliers condition. Besides, we design a plug-and-play SA strategy, which can be integrated into any unsupervised cutting-edge registration methods to boost their performance.

REFERENCES

[1] Y. Shen, L. Hui, H. Jiang, J. Xie, and J. Yang, “Reliable inlier evaluation for unsupervised point cloud registration,” in *Proc. AAAI Conf. Artif. Intell.*, 2022, pp. 2198–2206.

[2] S. Agarwal, N. Snavely, I. Simon, S. M. Seitz, and R. Szeliski, “Building Rome in a day,” in *Proc. IEEE 12th Int. Conf. Comput. Vis.*, Sep. 2009, pp. 72–79.

[3] J.-E. Deschaud, “IMLS-SLAM: Scan-to-model matching based on 3D data,” in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2018, pp. 2480–2485.

[4] B. Drost, M. Ulrich, N. Navab, and S. Ilic, "Model globally, match locally: Efficient and robust 3D object recognition," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, Jun. 2010, pp. 998–1005.

[5] P. J. Besl and N. D. McKay, "A method for registration of 3-D shapes," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 14, no. 2, pp. 239–256, Feb. 1992.

[6] J. Yang, H. Li, D. Campbell, and Y. Jia, "Go-ICP: A globally optimal solution to 3D ICP point-set registration," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 38, no. 11, pp. 2241–2254, Nov. 2016.

[7] Y. Wang and J. Solomon, "Deep closest point: Learning representations for point cloud registration," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 3522–3531.

[8] Z. J. Yew and G. H. Lee, "REGTR: End-to-end point cloud correspondences with transformers," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 6667–6676.

[9] H. Wang et al., "Robust multiview point cloud registration with reliable pose graph initialization and history reweighting," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2023, pp. 9506–9515.

[10] J. Zhang, Y. Yao, and B. Deng, "Fast and robust iterative closest point," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 44, no. 7, pp. 3450–3466, Jul. 2022.

[11] X. Zhang, J. Yang, S. Zhang, and Y. Zhang, "3D registration with maximal cliques," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Vancouver, BC, Canada, Jun. 2023, pp. 17745–17754.

[12] R. B. Rusu, N. Blodow, and M. Beetz, "Fast point feature histograms (FPFH) for 3D registration," in *Proc. IEEE Int. Conf. Robot. Autom.*, May 2009, pp. 3212–3217.

[13] J. Li, C. Zhang, Z. Xu, H. Zhou, and C. Zhang, "Iterative distance-aware similarity matrix convolution with mutual-supervised point elimination for efficient point cloud registration," in *Computer Vision—ECCV 2020* (Lecture Notes in Computer Science), vol. 12369, A. Vedaldi, H. Bischof, T. Brox, and J. Frahm, Eds., Cham, Switzerland: Springer, 2020, pp. 378–394.

[14] S. Huang, Z. Gojcic, M. Usvyatsov, A. Wieser, and K. Schindler, "Predator: Registration of 3D point clouds with low overlap," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2021, pp. 4267–4276.

[15] X. Huang, G. Mei, and J. Zhang, "Feature-metric registration: A fast semi-supervised approach for robust point cloud registration without correspondences," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 11363–11371.

[16] H. Jiang, Y. Shen, J. Xie, J. Li, J. Qian, and J. Yang, "Sampling network guided cross-entropy method for unsupervised point cloud registration," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 6108–6117.

[17] X. Li, L. Wang, and Y. Fang, "Unsupervised category-specific partial point set registration via joint shape completion and registration," *IEEE Trans. Vis. Comput. Graphics*, vol. 29, no. 7, pp. 3251–3265, Jul. 2023.

[18] A. Vaswani et al., "Attention is all you need," in *Proc. Adv. Neural Inform. Process. Syst. (NIPS)*, 2017, pp. 5998–6008.

[19] K.-L. Low, "Linear least-squares optimization for point-to-plane ICP surface registration," *Chapel Hill, Univ. North Carolina*, vol. 4, no. 10, pp. 1–3, 2004.

[20] A. Segal, D. Haehnel, and S. Thrun, "Generalized-ICP," in *Proc. Robot., Sci. Syst.*, 2009, vol. 2, no. 4, p. 435.

[21] R. I. Hartley and F. Kahl, "Global optimization through rotation space search," *Int. J. Comput. Vis.*, vol. 82, no. 1, pp. 64–79, Apr. 2009.

[22] M. A. Fischler and R. C. Bolles, "Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography," *Commun. ACM*, vol. 24, no. 6, pp. 381–395, 1981.

[23] H. Yang, J. Shi, and L. Carlone, "TEASER: Fast and certifiable point cloud registration," *IEEE Trans. Robot.*, vol. 37, no. 2, pp. 314–333, Apr. 2021.

[24] S. Chen, L. Nan, R. Xia, J. Zhao, and P. Wonka, "PLADE: A plane-based descriptor for point cloud registration with small overlap," *IEEE Trans. Geosci. Remote Sens.*, vol. 58, no. 4, pp. 2530–2540, Apr. 2020.

[25] Z. Deng, Y. Yao, B. Deng, and J. Zhang, "A robust loss for point cloud registration," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, Oct. 2021, pp. 6118–6127.

[26] L. He et al., "GFOICP: Geometric feature optimized iterative closest point for 3-D point cloud registration," *IEEE Trans. Geosci. Remote Sens.*, vol. 61, 2023, Art. no. 5704217.

[27] Y. Aoki, H. Goforth, R. A. Srivatsan, and S. Lucey, "PointNetLK: Robust & efficient point cloud registration using PointNet," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 7156–7165.

[28] R. Q. Charles, H. Su, M. Kaichun, and L. J. Guibas, "PointNet: Deep learning on point sets for 3D classification and segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 77–85.

[29] Y. Wang and J. M. Solomon, "Prnet: Self-supervised learning for partial-to-partial registration," in *Proc. Adv. Neural Inf. Process. Syst.*, H. M. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. B. Fox, and R. Garnett, Eds., 2019, pp. 8812–8824.

[30] Z. J. Yew and G. H. Lee, "RPM-Net: Robust point matching using learned features," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 11821–11830.

[31] K. Fu, J. Luo, X. Luo, S. Liu, C. Zhang, and M. Wang, "Robust point cloud registration framework based on deep graph matching," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 45, no. 5, pp. 6183–6195, May 2023.

[32] W. Yuan, B. Eckart, K. Kim, V. Jampani, D. Fox, and J. Kautz, "DeepGMR: Learning latent Gaussian mixture models for registration," in *Computer Vision—ECCV 2020* (Lecture Notes in Computer Science), vol. 12350, A. Vedaldi, H. Bischof, T. Brox, and J. Frahm, Eds., Cham, Switzerland: Springer, 2020, pp. 733–750.

[33] C. Choy, W. Dong, and V. Koltun, "Deep global registration," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 2511–2520.

[34] G. D. Pais, S. Ramalingam, V. M. Govindu, J. C. Nascimento, R. Chellappa, and P. Miraldo, "3DRegNet: A deep neural network for 3D point registration," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2020, pp. 7191–7201.

[35] X. Bai et al., "PointDSC: Robust point cloud registration using deep spatial consistency," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 15859–15869.

[36] J. Lee, S. Kim, M. Cho, and J. Park, "Deep Hough voting for robust global registration," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 15974–15983.

[37] J. Yang, J. Chen, S. Quan, W. Wang, and Y. Zhang, "Correspondence selection with loose–tight geometric voting for 3-D point cloud registration," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, 2022, Art. no. 5701914.

[38] J. Wang et al., "PG-net: Progressive guidance network via robust contextual embedding for efficient point cloud registration," *IEEE Trans. Geosci. Remote Sens.*, vol. 61, 2023, Art. no. 5701712.

[39] R. Li et al., "An effective point cloud registration method based on robust removal of outliers," *IEEE Trans. Geosci. Remote Sens.*, vol. 62, 2024, Art. no. 5701316.

[40] H. Chen, Z. Wei, Y. Xu, M. Wei, and J. Wang, "ImLoveNet: Misaligned image-supported registration network for low-overlap point cloud pairs," in *Proc. SIGGRAPH*, M. Nandigjav, N. J. Mitra, and A. Hertzmann, Eds., 2022, pp. 1–29.

[41] Z. Qin, H. Yu, C. Wang, Y. Guo, Y. Peng, and K. Xu, "Geometric transformer for fast and robust point cloud registration," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 11133–11142.

[42] J. Xu, Y. Huang, Z. Wan, and J. Wei, "GLORN: Strong generalization fully convolutional network for low-overlap point cloud registration," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, 2022, Art. no. 5704814.

[43] G. Ma and H. Wei, "A novel sketch-based framework utilizing contour cues for efficient point cloud registration," *IEEE Trans. Geosci. Remote Sens.*, vol. 61, 2023, Art. no. 5703616.

[44] R. She et al., "PointDifformer: Robust point cloud registration with neural diffusion and transformer," *IEEE Trans. Geosci. Remote Sens.*, vol. 62, 2024, Art. no. 5701015.

[45] Y. Xie, J. Zhu, S. Li, N. Hu, and P. Shi, "HECPG: Hyperbolic embedding and confident patch-guided network for point cloud matching," *IEEE Trans. Geosci. Remote Sens.*, vol. 62, 2024, Art. no. 5702212.

[46] H. Jiang, J. Qian, J. Xie, and J. Yang, "Planning with learned dynamic model for unsupervised point cloud registration," in *Proc. Int. Joint Conf. Artif. Intell.*, Z. Zhou, Ed., 2021, pp. 772–778.

[47] J. Li and G. H. Lee, "USIP: Unsupervised stable interest point detection from 3D point clouds," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 361–370.

[48] L. Li, H. Fu, and M. Ovsjanikov, "WSDesc: Weakly supervised 3D local descriptor learning for point cloud registration," *IEEE Trans. Vis. Comput. Graphics*, vol. 29, no. 7, pp. 3368–3379, Jul. 2023.

[49] Y. Wang, Y. Sun, Z. Liu, S. E. Sarma, M. M. Bronstein, and J. M. Solomon, "Dynamic graph CNN for learning on point clouds," *ACM Trans. Graph.*, vol. 38, no. 5, pp. 1–12, Oct. 2019.

[50] N. Engel, V. Belagiannis, and K. Dietmayer, "Point transformer," *IEEE Access*, vol. 9, pp. 134826–134840, 2021.

[51] R. Sinkhorn, "A relationship between arbitrary positive matrices and doubly stochastic matrices," *Ann. Math. Statist.*, vol. 35, no. 2, pp. 876–879, Jun. 1964.

[52] Z. Zhang, R. Girdhar, A. Joulin, and I. Misra, "Self-supervised pretraining of 3D features on any point-cloud," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 10232–10243.

[53] A. Sanghi, "Info3D: Representation learning on 3D objects using mutual information maximization and contrastive learning," in *Computer Vision—ECCV 2020* (Lecture Notes in Computer Science), vol. 12374, A. Vedaldi, H. Bischof, T. Brox, and J. Frahm, Eds., Cham, Switzerland: Springer, 2020, pp. 626–642.

[54] H. Fan, H. Su, and L. J. Guibas, "A point set generation network for 3D object reconstruction from a single image," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jul. 2017, pp. 605–613.

[55] Z. Wu et al., "3D ShapeNets: A deep representation for volumetric shapes," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 1912–1920.

[56] M. A. Uy, Q. Pham, B. Hua, T. Nguyen, and S. Yeung, "Revisiting point cloud classification: A new benchmark dataset and classification model on real-world data," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 1588–1597.

[57] S. Choi, Q.-Y. Zhou, and V. Koltun, "Robust reconstruction of indoor scenes," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 5556–5565.

[58] J. Shotton, B. Glocker, C. Zach, S. Izadi, A. Criminisi, and A. Fitzgibbon, "Scene coordinate regression forests for camera relocalization in RGB-D images," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2013, pp. 2930–2937.

[59] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for autonomous driving? The KITTI vision benchmark suite," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2012, pp. 3354–3361.

[60] Y. Xie, B. Wang, S. Li, and J. Zhu, "Iterative feedback network for unsupervised point cloud registration," *IEEE Robot. Autom. Lett.*, vol. 9, no. 3, pp. 2327–2334, Mar. 2024.

[61] S. Ao, Q. Hu, H. Wang, K. Xu, and Y. Guo, "BUFFER: Balancing accuracy, efficiency, and generalizability in point cloud registration," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2023, pp. 1255–1264.

[62] P.-E. Sarlin, D. DeTone, T. Malisiewicz, and A. Rabinovich, "SuperGlue: Learning feature matching with graph neural networks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 4937–4946.

[63] Z. Chen et al., "UTOPIC: Uncertainty-aware overlap prediction network for partial point cloud registration," *Comput. Graph. Forum*, vol. 41, no. 7, pp. 87–98, Oct. 2022.

Chengyu Zheng is currently pursuing the Ph.D. degree with the School of Computer Science and Technology, Nanjing University of Aeronautics and Astronautics, Nanjing, China.

His research interests include deep learning, image processing, and computer vision.

Mengjiao Ma is currently pursuing the Ph.D. degree with the School of Computer Science and Technology, Nanjing University of Aeronautics and Astronautics, Nanjing, China.

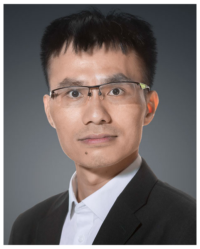
Her research interests include 3-D computer vision and large language models.

Zhilei Chen received the B.Sc. degree from Nanjing Normal University, Nanjing, China, in 2021. He is currently pursuing the M.Sc. degree with Nanjing University of Aeronautics and Astronautics, Nanjing.

His research interests include 3-D computer vision and learning-based geometry processing.

Honghua Chen received the master's degree from Nanjing Normal University, Nanjing, China, in 2017, and the Ph.D. degree from Nanjing University of Aeronautics and Astronautics (NUAA), Nanjing, in 2022.

His research interests include smart geometry processing.



Weiming Wang received the Ph.D. degree from The Chinese University of Hong Kong, Hong Kong, in 2014.

He is currently an Assistant Professor with the School of Science and Technology, Hong Kong Metropolitan University, Hong Kong. He has over 50 publications in refereed journals and conferences and led more than ten research grants. His research interests include image processing, point cloud processing, and deep learning.

Dr. Wang is an Associate Editor of *The Visual Computer*.

Mingqiang Wei (Senior Member, IEEE) received the Ph.D. degree in computer science and engineering from The Chinese University of Hong Kong (CUHK), Hong Kong, in 2014.

He is currently a Professor with the School of Computer Science and Technology, Nanjing University of Aeronautics and Astronautics (NUAA), Nanjing, China. His research interests include 3-D vision, computer graphics, and deep learning.

Dr. Wei was a recipient of the CUHK Young Scholar Thesis Awards in 2014. He is also an Associate Editor of *ACM Transactions on Multimedia Computing, Communications, and Applications* (TOMM) and *The Visual Computer* journal and a Guest Editor of IEEE TRANSACTIONS ON MULTIMEDIA.