CCAG: End-to-End Point Cloud Registration

Yong Wang, Pengbo Zhou, Guohua Geng, Li An, and Yangyang Liu

***Abstract*—Point cloud registration is a crucial task in computer vision and 3D reconstruction, aiming to align multiple point clouds to achieve globally consistent geometric structures. However, traditional point cloud registration methods face challenges when dealing with low overlap and large-scale point cloud data. To overcome these issues, we propose an end-to-end point cloud registration method called CCAG. The CCAG algorithm leverages the Cross-Convolution Attention module, which combines cross-attention mechanism and depth-wise separable convolution to capture relationships between point clouds and integrate features. Through cross-attention computation, this module establishes associations between point clouds and utilizes depth-wise separable convolution operations to extract local features and spatial relationships. Furthermore, the CCAG algorithm introduces Adaptive Graph Convolution MLP, which dynamically adjusts node representations based on the positions of nodes in the graph structure and features of neighboring nodes, enhancing the expressive power of nodes through MLP. Our algorithm demonstrates competitive performance in multiple benchmark tests, including 3DMatch/3DLoMatch, KITTI, ModelNet/ModelLoNet, and MVP-RG.**

***Index Terms*—Computer vision, cross-convolution attention, end-to-end, large-scale point cloud data, point cloud registration.**

I. Introduction

With the widespread application of 3D perception and point cloud data, point cloud registration has attracted significant attention as an important problem in the fields of computer vision and machine learning. Point cloud registration aims to align point clouds acquired from different viewpoints or at different times to obtain consistent coordinate systems or pose information. It has wide-ranging applications in various fields, including 3D modeling, augmented reality, autonomous driving, robot perception, and more. However, traditional methods for point cloud registration, relying on handcrafted feature descriptors and optimization algorithms, struggle to perform well in scenarios with low overlap and large-scale problems due to the non-structured and incomplete nature of point cloud data. Therefore, point cloud registration still faces several challenges [1].

Traditional point cloud registration methods primarily rely on manually designed feature descriptors and optimization algorithms, such as using geometric properties (e.g., normals, curvature) of points or handcrafted local feature descriptors. While these methods perform well in small-scale and locally deformable scenarios, their performance is limited when dealing with complex point cloud data, noise, occlusions, and outliers.

In recent years, the rapid development of deep learning techniques has brought new opportunities to point cloud registration. Deep learning methods can automatically learn feature representations of point clouds and perform registration through neural networks. Compared to traditional methods, deep learning methods possess stronger expressive power and robustness, enabling them to handle large-scale and complex point cloud data and learn more discriminative feature representations.

Based on deep learning, point cloud registration methods can be categorized into two types: feature learning and end-to-end. Feature learning methods aim to learn feature representations of point clouds through deep learning networks and utilize traditional optimization algorithms for registration. These methods can learn more expressive and robust feature representations from point cloud data, leading to improved performance in registration tasks. End-to-end methods directly learn the transformation parameters for registration from raw point cloud data, reducing the complexity of feature extraction and matching while providing higher computational efficiency and robustness.

Furthermore, the correlation between point clouds and the extraction of local features are crucial issues in registration. In recent years, attention mechanisms and convolution operations have gained widespread attention in the field of point cloud processing. The combination of attention mechanisms and convolution operations can enhance the feature modeling and context learning capabilities of models, handle spatial locality features, and exhibit strong model generalization. However, challenges still exist, such as point cloud data regularization and the need for parameter selection and design.

In this letter, we propose an end-to-end point cloud registration method based on CCAG. This method first extracts feature representations of input point clouds through sampling and sparsification. Then, the feature-extracted point cloud representations are constructed into a graph structure, where the points in the point clouds serve as nodes in the graph, and the relationships between points are treated as edges in the graph. Finally, by applying the Adaptive Graph Convolution MLP module and the Cross-Convolution Attention module, information propagation and feature fusion are performed on the nodes of the graph.

Manuscript received 9 August 2023; accepted 1 November 2023. Date of publication 9 November 2023; date of current version 28 November 2023. This letter was recommended for publication by Associate Editor X. Huang and Editor C. Cadena Lerma upon evaluation of the reviewers' comments. This work was supported in part by the National Natural Science Foundation of China under Grant 62271393, and in part by the National Key Research and Development Plan under Grants 2020YFC1523301 and 2020YFC1523303. *(Corresponding authors: Guohua Geng; Li An.)*

Yong Wang, Guohua Geng, Li An, and Yangyang Liu are with the School of Information Science and Technology, Northwest University, Xian 710127, China (e-mail: 1773943023@qq.com; ghgeng@nwu.edu.cn; anli18394493685@163.com; 513608191@qq.com).

Pengbo Zhou is with the School of Arts and Communication, Beijing Normal University, Beijing 100875, China (e-mail: fengye883@foxmail.com).

Digital Object Identifier 10.1109/LRA.2023.3331666

Graph convolution operations capture global and local point cloud features, while the Cross-Convolution Attention module computes the attention distribution between different input point clouds, associating relevant information across different point clouds, learning their dependencies and importance weights, and assisting in matching and aligning different point clouds.

The end-to-end CCAG-based point cloud registration method has the following advantages:

- The AGCM module incorporates shape descriptors and local features into the model, addressing the issues of point cloud data regularization, parameter selection, and design, and enhancing the model's expressive power.
- The Cross-Convolution Attention module, by introducing cross-attention computations, models the correlation between point clouds and leverages convolution operations to extract local features, further improving the accuracy and robustness of registration.
- We evaluated the proposed method on indoor, outdoor, synthetic, and incomplete synthetic datasets, demonstrating state-of-the-art performance.

II. Related Work

Feature Extraction Based on Deep Learning: In recent years, with the rapid development and advancement of deep learning, new opportunities have emerged for point cloud registration. Wang et al. [2] proposed a method using rotation-equivariant descriptors to address the issue of traditional point cloud registration methods relying on handcrafted feature descriptors. By introducing rotational equivariance in the design of feature descriptors, this method ensures their invariance to rotational transformations of point clouds. Specifically, the authors employed spherical harmonics to represent the local geometric features of point clouds and constructed rotation-equivariant descriptors using rotation-invariant spherical harmonics. Kadam et al. [3] presented an unsupervised point cloud registration method based on R-pointhop. The method first applies an adaptive sampling strategy to downsample the point cloud, reducing computational complexity. Then, it computes feature descriptors of the point cloud for local neighborhood matching and initial alignment. Next, an iterative registration method called pointhop is used to progressively improve the alignment accuracy. The pointhop method aligns the point cloud iteratively through adaptive feature selection and geometry-based strategies until achieving the final precise registration result.

Furthermore, Poiesi et al. [4] proposed an improved method for point cloud registration by learning general and distinctive 3D local depth descriptors. This approach uses deep learning to extract feature representations of point clouds and utilizes the learned descriptors for matching and registration, thereby achieving more accurate and robust point cloud registration results. In addressing the balance between accuracy, efficiency, and generalization, Ao et al. [5] introduced a point cloud registration method called BUFFER. This method leverages both point-to-point and face-to-face techniques while overcoming their inherent shortcomings. Specifically, they first introduced a point-to-point learner to enhance computational efficiency and improve feature representation by predicting key points and estimating point orientations. Subsequently, they deployed a face-to-face embedder to extract efficient and generic face features using lightweight local feature learners. Different from commonly used supervised models, Huang et al. [6] proposed an innovative unsupervised point cloud registration method. They utilized a unified Gaussian mixture model to handle sampling noise and density variations, thereby enhancing the accuracy and efficiency of point cloud registration while reducing dependence on supervised information.

End-to-end registration based on deep learning: With the rapid development of deep learning, end-to-end point cloud registration methods based on deep learning have gradually become a research hotspot. These methods directly learn the transformation parameters for registration from raw point cloud data without explicit feature extraction and matching processes. By constructing end-to-end neural network models that map input point cloud data to output transformation parameters, accurate alignment of point clouds can be achieved.

Traditional point cloud registration methods often require a high degree of overlap between point clouds to establish correspondences through matching shared features. However, in low-overlap scenarios, the shared features between point clouds are scarce, making it challenging for traditional methods to achieve accurate registration. To address this issue, the Predator method [7] introduced an autoencoder-based approach. The encoder part of the autoencoder is used to extract feature representations of the point cloud, while the decoder part is responsible for reconstructing the input point cloud. To handle point cloud registration in low-overlap scenarios, Predator incorporates an adaptive point cloud sampling method. By dynamically adjusting the point cloud's sampling density during the registration process, Predator can better handle point cloud registration problems in low-overlap situations.

On the other hand, traditional point cloud registration methods often rely on iterative optimization processes to find the optimal registration transformation. However, these iterative optimization processes often consume a significant amount of computation time and require high-quality initial alignment. To address this problem, Qin et al. [8] proposed a fast and robust point cloud registration method based on geometric transformations. This method first computes an initial coarse alignment transformation through feature extraction and matching. Then, a geometric transformation module is utilized to refine the alignment by applying nonlinear transformations based on learnable parameters, capturing the geometric relationships between point clouds. Finally, through fine alignment and registration evaluation, the final point cloud registration result is obtained.

Existing point cloud descriptors rely on structural information while ignoring texture information, yet texture information is crucial for distinguishing scene components. To address this issue, Huang et al. [9], [10] proposed a method based on multimodal fusion to generate point cloud registration descriptors that take into account both structural and texture information.

Furthermore, there are methods that leverage Transformer networks to learn end-to-end point cloud correspondences. For example, Yew et al. [11] proposed a method using Transformer

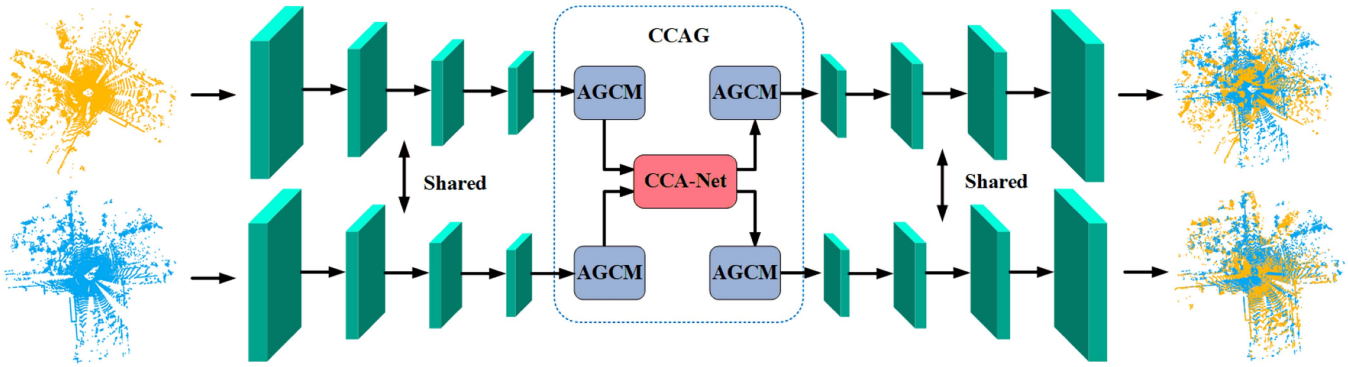

Fig. 1. Network architecture of CCAG.

networks. This method first encodes the input point cloud through an encoder network, transforming the position and feature information of the point cloud into a set of high-dimensional representations. Then, self-attention mechanisms are employed to learn the relationships between points in the point cloud, capturing both global and local feature associations in the encoded point cloud representation. Finally, a fully connected layer maps the point cloud representation to the prediction space of point cloud correspondences. This prediction space can represent the probability of correspondence between each point in the point cloud and other points. Based on the predicted correspondences, an iterative optimization algorithm is used to resolve the optimal point cloud correspondences. Additionally, there are unsupervised deep probabilistic methods, such as the method proposed by Mei et al. [12]. This method uses a deep neural network to encode partial input point clouds into low-dimensional feature vectors, which are then inputted into a Gaussian mixture model. By introducing deep probabilistic models, this method can automatically learn the matching and registration process from partial point clouds in an unsupervised manner.

In summary, deep learning-based point cloud registration methods offer more accurate, robust, and adaptable solutions for various registration scenarios through feature learning and end-to-end training. These methods can handle noise and incompleteness in point clouds and achieve satisfactory registration results. However, there are still challenges, such as point cloud registration in large-scale scenes and low-overlap scenarios, which remain hotspots and research challenges in the field.

III. Method

Our method follows a hierarchical structure, similar to D3Feat [13] and Predator [7]. The specific process is illustrated in Fig. 1.

A. Problem Setting

For two point clouds defined as the source point cloud $P = \{p_i \in \mathbb{R}^3 \mid i = 1, 2, \ldots, N\}$ and the target point cloud $Q = \{q_i \in \mathbb{R}^3 \mid i = 1, 2, \ldots, M\}$, where N and M are the number of points in point clouds P and Q, respectively, the goal of point cloud registration is to align the two point clouds using an unknown 3D rigid transformation $RT = \{R, T\}$, which consists of a rotation $R \in SO(3)$ and a translation $T \in \mathbb{R}^3$. The transformation matrix can be defined as:

$$\min_{R,T} \sum\nolimits_{(p_i,q_i)\in\vartheta} \|R \cdot p_i + T - q_i\|_2^2 \tag{1}$$

Where ϑ represents the ground truth correspondences between the points in P and Q. The notation $\| \bullet \|$ denotes the Euclidean norm.

To address this problem, we propose an end-to-end point cloud registration algorithm called CCAG, which takes a pair of point clouds as input and output the correspondence between points, and estimates rigid transformations using RANSAC.

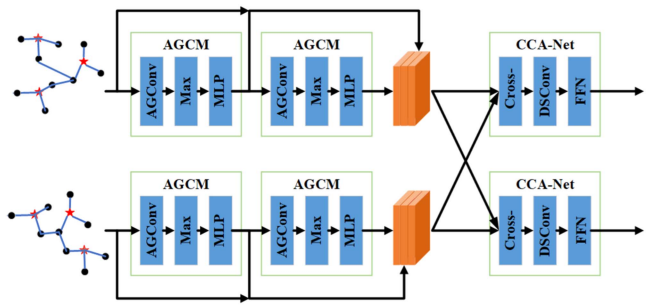

Fig. 2. Structural flow of the AGCM module and CCA-Net module.

B. Downsampling and Feature Extraction

For the denser original point clouds $P \in \mathbb{R}^{N\times 3}$ and $Q \in \mathbb{R}^{M\times 3}$, we utilize the KPConv module as the backbone, which consists of a series of residual modules and strided convolutions, to perform downsampling and reduce the number of keypoints to $P' \in \mathbb{R}^{N'\times 3}$ and $Q' \in \mathbb{R}^{M'\times 3}$, respectively (where $N > N'$ and $M > M'$). Furthermore, we employ a shared encoding method to extract relevant features, resulting in $F'_{P'} \in \mathbb{R}^{N'\times D}$ and $F'_{Q'} \in \mathbb{R}^{M'\times D}$, where D represents the feature dimension.

C. Overlap Attention Module

The structural flow of the AGCM module and CCA-Net module in the Overlapping Attention Module is shown in Fig. 2.

Adaptive Graph Convolution MLP: To address the irregularity of point cloud data and the challenges in selecting and designing network structure parameters, we propose an Adaptive Graph Convolutional MLP network, referred to as AGCM. Since the processing steps for the source point cloud and the target point cloud are identical, we will use the source point cloud as an example.

Firstly, we define the graph structure $G(V, E)$, where $V = \{1, 2, \ldots, N'\}$ represents the set of vertices and $E \subseteq |V| \times |V|$ represents the set of edges, with each vertex corresponding to a point p'_i in P'.

Next, through the adaptive graph convolution operation, the feature information of nodes is dynamically aggregated and transformed. This operation utilizes the feature information of neighboring nodes to update the feature representation of the current node. Then, the maxpooling operation is employed to extract key features. Finally, the MLP enhancement algorithm is introduced to enhance the expressiveness of point cloud data and enrich the features.

$$^{(k+1)}P' = \vartheta_m(\max(\Delta p_{ij} \cdot \Delta f_{ij})) \tag{2}$$

$$F_i^{AGCM} = \gamma_\theta(cat[^{(0)}P', ^{(1)}P', ^{(2)}P']) \tag{3}$$

Where, $\Delta p_{ij} = [p'_i, p'_j - p'_i]$ is the graph vertex and its relative position, Δf_{ij} is defined as $[f'_i, f'_j - f'_i]$ correspondingly, and $cat[\bullet, \bullet]$ is the concatenation operation. ϑ_m stands for pointwise MLP and γ_θ stands for linear layer. $\max$ stands for maxpooling.

Cross-Convolution Attention: To enhance the feature modeling capability, context relationship learning, handling spatial locality features, and possessing strong model generalization ability, we propose a Cross-Convolution Attention module, referred to as CCA-Net.

Firstly, we define a 4-layer multi-head cross-attention module that inherits the local feature information from the AGCM module and focuses on the mutual information between two point clouds.

$$MHAttn(Q, K, V) = (Head_1 \oplus \cdots \oplus Head_H)W^o \tag{4}$$

$$Head_H = Attn(QW_h^Q, KW_h^K, VW_h^V) \tag{5}$$

Where, $W_h^Q \in \mathbb{R}^{d\times d_Q}, W_h^K \in \mathbb{R}^{d\times d_K}, W_h^V \in \mathbb{R}^{d\times d_V}, W^O \in \mathbb{R}^{d_H ead\times d}$ are the learning transformation matrix, d_Q, d_K, d_V are the dimensions of Query, Key and Value, $d_Q = d_K = d_V = d_{Head} = D/H$. H is the number of heads.

Firstly, we define $F'_{P'} = (x_1^{P'}, x_2^{P'} \cdots x_{N'}^{P'})$ and $F'_{Q'} = (x_1^{Q'}, x_2^{Q'} \cdots x_{N'}^{Q'})$ as the input $MHAttn(F'_{P'}, F'_{Q'}, F'_{Q'})$ of the cross-attention module in the i-th layer, $Z'' = (z_1^{P',Q'}, z_2^{P',Q'} \cdots z_{N'}^{P',Q'})$ is the output matrix, and its formula is as follows:

$$z_i^{P',Q'} = \sum_{j=1}^{N'} soft\max(\alpha_{i,j}^{Cross-}) x_j^{Q'} W^{V,Q'} \tag{6}$$

Where, $\alpha_{i,j}^{Cross-}$ is the weight coefficient that has not been normalized, and its definition is as follows:

$$\alpha_{i,j}^{Cross-} = \frac{1}{\sqrt{d_{head}}}(x_i^{Q'} W^{Q,Q'})(x_j^{P'} W^{K,P'})^T \tag{7}$$

Next, convolution is used to further capture the local and global features of the point cloud. Its definition is as follows:

Finally, the co-global context information between the two point clouds is output, which is defined as follows:

$$F_i^{DSConv} = F'_{P'} + DSConv(z_i^{P',Q'}) \tag{8}$$

Where, $DSConv$ represents depthwise separable convolution, which consists of two components: Depthwise Convolution and Pointwise Convolution. DSConv = PWConv (DepConv(•)), DepConv, PWConv represent depthwise convolution and pointwise convolution respectively.

$$F_i^{CCA-Net} = F_i^{DSConv} + MLP(F_i^{DSConv}) \tag{9}$$

D. Decoder

The module is configured in the usual manner, with a 3-layer network structure that mainly includes upsampling, linear transformation, and skip connections.

E. Loss Function

The CCAG network we proposed is trained end-to-end and supervised using ground truth. The specific loss function is as follows:

Feature Loss: Similar to the D3Feat and Predator approaches, we use a circle loss function to evaluate the feature loss and constrain the point-wise feature descriptors during the training of 3D point clouds. The circle loss function is defined as follows:

$$\mathcal{L}_{FL}^P = \frac{1}{N_P}\sum_{i=1}^{N_P} \log\left[1 + \sum_{j\in\varepsilon_p} e^{c\beta_p^j(d_i^j - \Delta p)} \bullet \sum_{k\in\varepsilon_n} e^{\lambda\beta_p^k(\Delta n - d_i^k)}\right] \tag{10}$$

Where, d_i^j represents the Euclidean distance between features, $d_i^j = \|f_{p_i} - f_{q_j}\|_2$. ε_p and ε_n respectively represent the matching and unmatching points of the point set P_{RS} (random sampling points of the source point cloud), i.e., the positive and negative areas. Δp and Δn represent positive and negative regions, respectively. λ stands for predefined parameters. Similarly, the feature loss $\mathcal{L}_{FL}^P$ of the target point cloud is also calculated in the same way, so the total feature loss $\mathcal{L}_{FL} = \frac{1}{2}(\mathcal{L}_{FL}^P + \mathcal{L}_{FL}^Q)$.

Overlap Loss: We use a binary cross-entropy loss function for supervised training, which is defined as follows:

$$\mathcal{L}_{OL}^P = \frac{1}{N}\sum_{i=1}^{N} O_{p_i}^{label}\log(O_{p_i}) + (1 - O_{p_i}^{label})\log(1 - O_{p_i}) \tag{11}$$

Where, $O_{p_i}^{label}$ represents the overlap scores of ground truth at point p_i, which is defined as follows:

$$O_{p_i}^{label} = \begin{cases} 1, \left\|T_{P,Q}^{GT}(p_i) - NN(T_{P,Q}^{GT}(p_i), Q)\right\| < \tau_1 \\ 0, \text{otherwise} \end{cases} \tag{12}$$

Where $T_{P,Q}^{GT}$ represents the ground truth rigid transformation between the overlapping point clouds, and NN represents the nearest neighbor. τ_1 represents the overlap threshold. Similarly, the overlap loss $\mathcal{L}_{OL}^Q$ of the target point cloud is calculated in

the same way. Therefore, the total overlap loss is defined as $\mathcal{L}_{OL} = \frac{1}{2}(\mathcal{L}_{OL}^{P} + \mathcal{L}_{OL}^{Q})$.

To sum up, the overall loss function is $\mathcal{L} = \mathcal{L}_{FL} + \mathcal{L}_{OL}$.

IV. Experiments

A. Implementation Details and Evaluation Metrics

Our method is implemented using PyTorch [14] and trained on an Nvidia RTX 4090, Intel(R) Core(TM) i7-13700KF CPU @ 3.40 GHz, 128 GB RAM.

Following the evaluation metrics used in Predator [7], REGTR [11], and GMCNet [15], we evaluate our models on the datasets using Registration Recall (RR), Relative Rotation Error (RRE), and Relative Translation Error (RTE). In addition, we use the modified Chamfer Distance (CD) to evaluate the ModelNet40 dataset, and RMSE to evaluate the MVP-RG dataset. The definitions of RRE, RTE, CD are as follows:

$$RTE = \left\| t - t^{GT} \right\|_2 \tag{13}$$

$$RRE = \arccos\left(\frac{trace(R^T R^{GT}) - 1}{2} \right) \tag{14}$$

$$CD(P,Q) = \frac{1}{|P|} \sum_{p \in P} \min_{q \in Q_{raw}} \left\| T_{P,Q}^{GT}(p) - q \right\|_2^2 \tag{15}$$

$$+ \frac{1}{|Q|} \sum_{q \in Q} \min_{p \in P_{raw}} \left\| q - T_{P,Q}^{GT}(p) \right\|_2^2 \tag{16}$$

Where, t^{GT} and R^{GT} represent the ground-truth rotational error and translational error, respectively.

Registration recall is defined as the root mean square error of the transformation being less than 0.2 m. The formula is shown below:

$$RMSE = \sqrt{\frac{1}{|C_{ij}^{GT}|} \sum_{(p,q) \in C_{ij}^{GT}} \left\| T_{P,Q}^{GT}(p) - q \right\|_2^2} \tag{17}$$

Where, C_{ij}^{GT} represents the set of ground-truth correspondences.

Note that, unlike the RMSE in registration recall, where only matched points are considered, in the MVP-RG dataset, all points are involved in the computation. Therefore, to differentiate it from (18), we define the RMSE for the MVP-RG dataset as follows:

$$\mathcal{L}_{RMSE} = \frac{1}{N} \sum_{i=1}^{N} \left\| T^{GT}(p_i) - T(p_i) \right\|_2 \tag{18}$$

B. 3DMatch and 3DLoMatch

Dataset: The 3DMatch dataset contains real indoor data from 62 scenarios, with 46 scenes for training, 8 for validation, and 8 for testing. We first pretrain our model using the Predator method and then evaluate it on the 3DMatch and 3DLoMatch datasets. The overlapping areas of the 3DMatch and 3DLoMatch datasets are greater than 30% and between 10% and 30%, respectively.

TABLE I
Performance on 3DMatch and 3DLoMatch Datasets

Method	3DMatch			3DLoMatch		
	RR(%)	RRE(%)	RTE(%)	RR(%)	RRE(%)	RTE(%)
Feature-based Methods						
FCGF [16]	85.1	1.949	0.066	40.1	3.147	0.100
D3Feat [13]	81.6	2.161	0.067	37.2	3.361	0.103
OMNet [17]	90.5	4.166	0.105	8.4	7.299	0.151
Predator [7]	89.0	2.029	0.064	59.8	3.048	0.093
GeoTrans [8]	92.0	1.808	0.063	74.0	2.934	0.089
REGTR [11]	92.0	1.567	0.049	64.8	2.827	0.077
Outlier Rejection Methods						
CoFiNet [18]	89.7	2.147	0.067	67.2	3.271	0.090
Lepard [19]	93.5	2.480	0.072	69.0	3.170	0.089
UDPReg [12]	91.4	1.642	0.064	64.3	2.951	0.086
MAC [20]	93.7	1.890	0.060	59.8	3.500	0.097
SC^2 -PCR++ [21]	**94.1**	2.040	0.065	61.1	3.720	0.105
CCAG	93.7	**1.481**	**0.049**	**76.2**	**2.73**	**0.068**

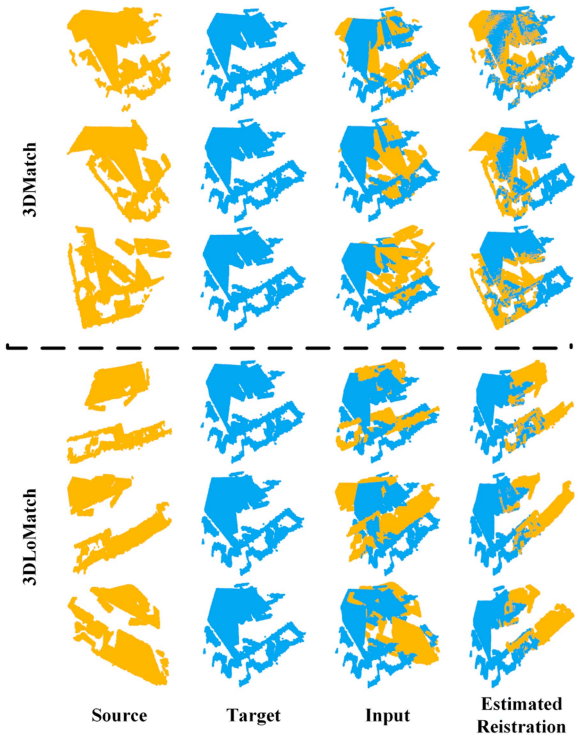

Fig. 3. Registration visualization on 3DMatch, 3DLoMatch.

Baselines: We compare our results with state-of-the-art methods such as FCGF [16], D3Feat [13], Predator [7], OMNet [17], CoFiNet [18], GeoTrans [8], REGTR [11], UDPReg [12], Lepard [19], MAC [20] and SC^2 -PCR++[21].

Registration Results: The performance comparison of different algorithms is shown in Table I, where the best-performing algorithms are indicated in bold. The data in the table are obtained from REGTR [11] and UDPReg [12]. Additionally, Fig. 3 illustrates the registration results on the 3DMatch and 3DLoMatch datasets. In the testing on the 3DMatch and 3DLoMatch datasets, our algorithm has demonstrated overall superior performance. It is noteworthy that in low-overlap validation, our algorithm has shown a remarkable improvement in RR compared to the MAC [20] and SC^2 -PCR++[21] algorithms, with increases of 16.3% and 15.1%, respectively.

TABLE II
FMR ON 3DMATCH AND 3DLOMATCH DATASETS

Method	3DMatch(FMR)	3DLoMatch(FMR)
Feature-based Methods		
FCGF [16]	95.2	76.6
D3Feat [13]	95.8	67.3
Predator [7]	96.6	78.6
GeoTrans [8]	97.9	88.3
RoITr [22]	98.0	89.6
Outlier Rejection Methods		
CoFiNet [18]	98.1	83.1
COTReg [23]	98.5	**89.5**
Lepard [19]	98.3	94.5
IMFNet [9]	98.6	80.3
DBENet [10]	98.6	80.3
CCAG	**98.7**	85.3

TABLE III
PERFORMANCE ON ODOMETRY KITTI DATASET

Method	RR(%)	RRE(%)	RTE(%)
FCGF [16]	9.5	0.30	96.6
D3Feat [13]	7.2	0.30	**99.8**
CoFiNet [18]	8.5	0.41	**99.8**
Predator [7]	6.8	0.27	**99.8**
GeoTrans [8]	7.4	0.27	**99.8**
MAC [20]	8.4	0.40	99.5
SC^2 -PCR++ [21]	7.1	0.32	99.6
CCAG	**6.1**	**0.23**	**99.8**

Following Predator [7], we evaluated the Feature Matching Recall (FMR). In the validation experiments presented in Table II , our algorithm demonstrated overall strong performance.

C. Odometry KITTI

Dataset: The Odometry KITTI dataset consists of large-scale LiDAR scanning data from 11 scenarios, with scenarios 0–5 used for training, scenarios 6–7 for validation, and scenarios 8–10 for testing.

Baselines: We select FCGF [16], D3Feat [13], Predator [7], CoFiNet [18], GeoTrans [8], MAC [20] and SC^2 -PCR++[21] as our baseline algorithms.

Registration Results: The comparison of different algorithms is shown in Table III, where the best-performing algorithms are indicated in bold. Additionally, Fig. 4 illustrates the registration results on the KITTI Odometry dataset. In the testing on the KITTI Odometry dataset, our algorithm achieves the highest average registration recall. Moreover, we also achieve the lowest RTE and RRE scores.

D. Modelnet

Data: The ModelNet40 dataset is a synthetic dataset composed of computer-aided design (CAD) models, consisting of 5112 samples for training, 1202 samples for validation, and 1266 samples for testing. We follow the training approach of RPM-Net and Predator and then evaluate our model on the ModelNet40 and ModelLoNet40 datasets. (The average overlap regions for ModelNet40 and ModelLoNet40 datasets are greater than 73.5% and 53.6%, respectively.)

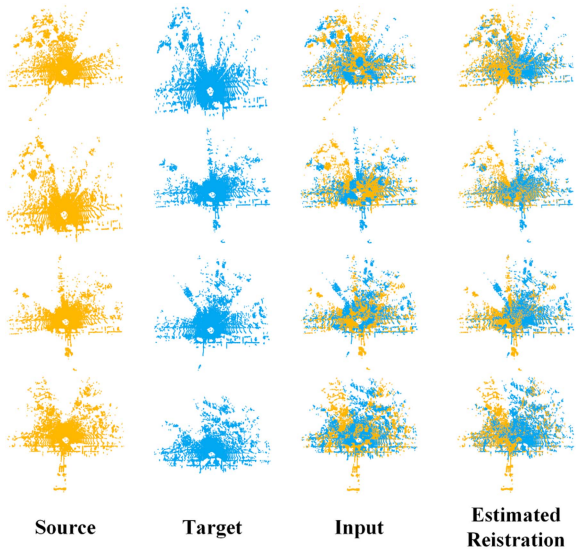

Fig. 4. Registration visualization on Odometry KITTI.

TABLE IV
PERFORMANCE ON MODELNET AND MODELLONET DATASETS

Method	ModelNet			ModelLoNet		
	CD(%)	RRE(%)	RTE(%)	CD(%)	RRE(%)	RTE(%)
RPM-Net [24]	0.0008	1.712	0.018	0.005	7.342	0.124
OMNet [17]	0.0015	2.947	0.003	0.0074	6.517	0.129
Predator [7]	0.0008	1.739	0.019	0.0083	5.235	0.132
REGTR [11]	0.0007	1.473	0.014	0.0037	3.93	0.087
UDPReg [12]	0.0306	1.331	**0.011**	0.0416	3.578	**0.069**
CCAG	**0.0006**	**1.442**	0.012	**0.0034**	**1.728**	0.074

Baselines: Based on the results of RPM-Net and Predator, we select several state-of-the-art algorithms, including RPM-Net [24], Predator [7], OMNet [17], REGTR [11] and UDPReg [12] as our baselines.

Registration Results: The comparison of different algorithms is shown in Table IV, where the best-performing algorithm is indicated in bold. The data for the table is sourced from REGTR. Additionally, Fig. 5 presents the registration results on the ModelNet/ModelLoNet datasets. Our algorithm exhibits good performance in various evaluations during the testing of the ModelNet and ModelLoNet datasets.

E. MVP-RG

Data: The MVP-RG dataset is a synthetic dataset of incomplete point clouds, consisting of 6400 pairs for training and 1200 pairs for testing. We trained our model using an approach similar to Predator and evaluated it on the MVP-RG dataset.

Baselines: We selected several state-of-the-art algorithms, including DCP [25], RPM-Net [24], GMCNet [15], IDAM [26], DeepGMR [27], Predator and DSMNet [28] as our baselines.

Registration Results: The comparison of different algorithms is shown in Table V, where the best performing algorithm is highlighted in bold, and the data is sourced from GMCNet [15]. Additionally, Fig. 6 illustrates the registration results on the MVP-RG dataset. In the testing phase on the MVP-RG dataset,

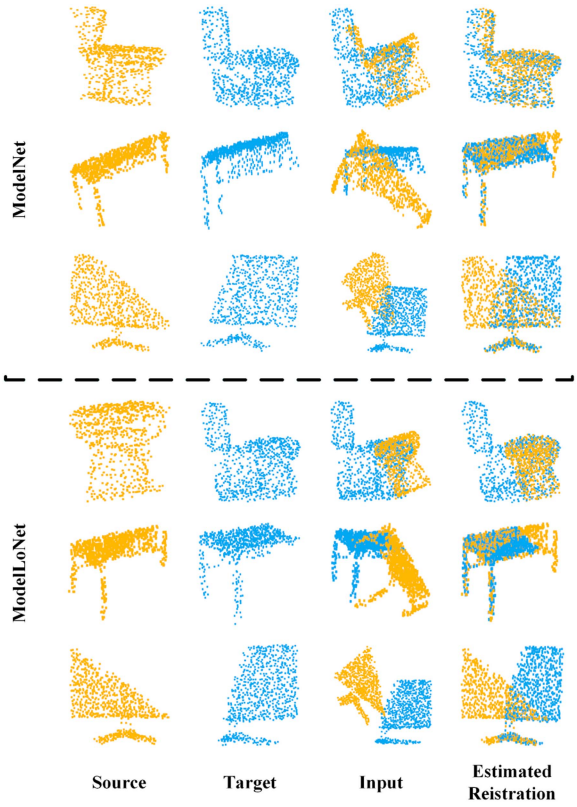

Fig. 5. Registration visualization on ModelNet and ModelLoNet.

TABLE V
PERFORMANCE ON MVP-RG DATASET

Method	RRE(%)	RTE(%)	$\mathcal{L}_{RMSE}(\%)$
DCP [25]	30.37	0.273	0.634
RPM-Net [24]	22.2	0.174	0.327
GMCNet [15]	16.57	0.174	0.246
IDAM [26]	24.35	0.28	0.344
DeepGMR [27]	49.72	0.385	0.696
Predator [7]	10.58	0.067	0.125
DSMNet [28]	14.17	0.158	-
CCAG	**7.53**	**0.051**	**0.09**

our algorithm outperforms other methods in all evaluation metrics.

F. Ablation Study

Importance of Individual Modules: To validate the effectiveness of selected modules in our model, we conducted ablation experiments on the 3DMatch/3DLoMatch datasets.

From Table VI, it can be seen that compared to the other four individual modules, our proposed CCAG algorithm demonstrates the best performance.

Loss Functions: To evaluate the impact of different loss functions and their combinations on the model, we conducted experiments on the 3DMatch/3DLoMatch datasets. Here, OL represents the overlap loss, FL represents the feature loss, and ML represents the matching loss.

From Table VII, it can be observed that all three loss functions can improve the registration accuracy, with the best performance

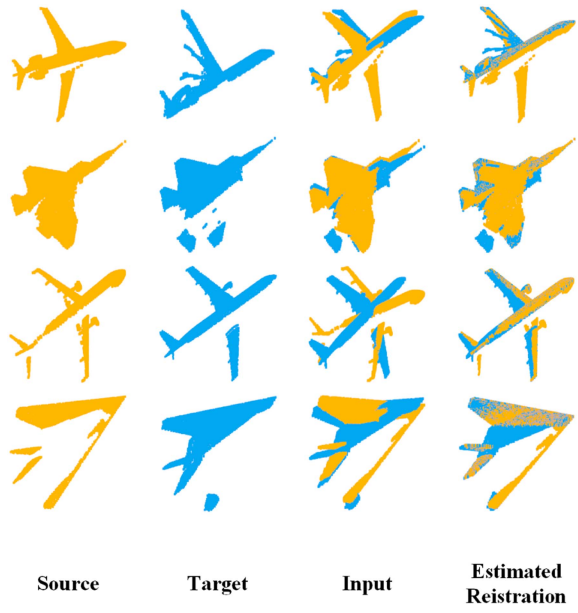

Fig. 6. Registration visualization on MVP-RG.

TABLE VI
ABLATION OF DIFFERENT MODULES ON THE 3DMATCH, 3DLOMATCH DATASET

Method	3DMatch			3DLoMatch		
	RR(%)	RRE(%)	RTE(%)	RR(%)	RRE(%)	RTE(%)
Base	58.7	2.9	0.075	17.8	5.089	0.103
+GCN	92.0	2.085	0.068	58.7	3.413	0.092
+AGCM	93.0	1.897	0.063	70.1	3.122	0.085
+Cross-Attention	88.0	2.061	0.069	55.6	3.616	0.103
+Cross-Convolution Attention	91.2	1.912	0.058	58.3	3.504	0.095
+GCN +Cross-Attention +AGCM	92.8	1.837	0.059	65.4	3.016	0.077
+Cross-Convolution Attention	**93.7**	**1.481**	**0.049**	**76.2**	**2.73**	**0.068**

TABLE VII
ABLATION OF DIFFERENT LOSS FUNCTIONS ON THE 3DMATCH, 3DLOMATCH DATASET

Method	3DMatch			3DLoMatch		
	RR(%)	RRE(%)	RTE(%)	RR(%)	RRE(%)	RTE(%)
OL	63.1	2.978	0.078	25.6	5.381	0.109
FL	92.8	1.921	0.068	67.1	3.161	0.088
OL+FL	**93.7**	**1.481**	**0.049**	**76.2**	**2.73**	**0.068**

achieved when the three loss functions are combined. Among them, in terms of single loss functions and pairwise combinations, the feature loss function or the combination involving the feature loss function shows the best performance. The overlap loss function performs well, while the matching loss function shows relatively poorer performance.

V. CONCLUSION

In response to the non-structural and incomplete nature of point clouds, as well as the challenges faced by traditional point cloud registration methods in handling low overlap and large-scale data, we propose an end-to-end point cloud registration

algorithm based on CCAG. In the CCAG algorithm, the AGCM module converts point clouds into graph structures and adaptively learns convolutional kernels based on the characteristics of point clouds, enabling effective capture of local and global features. The CCA-Net module introduces depth-wise separable convolution and cross-convolution attention computation to model the correlations between point clouds and extract local features of overlapping point clouds, thereby further improving the accuracy and robustness of registration.

Compared to traditional registration methods, our algorithm can better handle the geometric structure and local features of point clouds, providing more accurate results for point cloud registration tasks. This approach offers an effective solution for point cloud processing and the field of computer vision in point cloud registration tasks.

REFERENCES

[1] J. Li, P. Shi, Q. Hu, and Y. Zhang, "QGORE: Quadratic-time guaranteed outlier removal for point cloud registration," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 45, no. 9, pp. 11136–11151, Sep. 2023.

[2] H. Wang, Y. Liu, Z. Dong, and W. Wang, "You only hypothesize once: Point cloud registration with rotation-equivariant descriptors," in *Proc. 30th ACM Int. Conf. Multimedia*, 2022, pp. 1630–1641.

[3] P. Kadam, M. Zhang, S. Liu, and C.-C. J. Kuo, "R-pointhop: A green, accurate, and unsupervised point cloud registration method," *IEEE Trans. Image Process.*, vol. 31, pp. 2710–2725, 2022.

[4] F. Poiesi and D. Boscaini, "Learning general and distinctive 3D local deep descriptors for point cloud registration," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 45, no. 3, pp. 3979–3985, Mar. 2023.

[5] 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.*, 2023, pp. 1255–1264.

[6] X. Huang, S. Li, Y. Zuo, Y. Fang, J. Zhang, and X. Zhao, "Unsupervised point cloud registration by learning unified Gaussian mixture models," *IEEE Robot. Automat. Lett.*, vol. 7, no. 3, pp. 7028–7035, Jul. 2022.

[7] 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.*, 2021, pp. 4267–4276.

[8] 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.*, 2022, pp. 11143–11152.

[9] X. Huang, W. Qu, Y. Zuo, Y. Fang, and X. Zhao, "IMFNet: Interpretable multimodal fusion for point cloud registration," *IEEE Robot. Automat. Lett.*, vol. 7, no. 4, pp. 12323–12330, Oct. 2022.

[10] M. Yuan, X. Huang, K. Fu, Z. Li, and M. Wang, "Boosting 3D point cloud registration by transferring multi-modality knowledge," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2023, pp. 11734–11741.

[11] 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.*, 2022, pp. 6677–6686.

[12] G. Mei et al., "Unsupervised deep probabilistic approach for partial point cloud registration," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2023, pp. 13611–13620.

[13] X. Bai, Z. Luo, L. Zhou, H. Fu, L. Quan, and C.-L. Tai, "D3Feat: Joint learning of dense detection and description of 3D local features," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 6359–6367.

[14] A. Paszke et al., "PyTorch: An imperative style, high-performance deep learning library," in *Proc. 33rd Int. Conf. Neural Inf. Process. Syst.*, 2019, pp. 8026–8037.

[15] L. Pan, Z. Cai, and Z. Liu, "Robust partial-to-partial point cloud registration in a full range," 2021, *arXiv:2111.15606*.

[16] C. Choy, J. Park, and V. Koltun, "Fully convolutional geometric features," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2019, pp. 8958–8966.

[17] H. Xu, S. Liu, G. Wang, G. Liu, and B. Zeng, "OMNet: Learning overlapping mask for partial-to-partial point cloud registration," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2021, pp. 3132–3141.

[18] H. Yu, F. Li, M. Saleh, B. Busam, and S. Ilic, "CoFinet: Reliable coarse-to-fine correspondences for robust point cloud registration," in *Proc. Adv. Neural Inf. Process. Syst.*, 2021, pp. 23872–23884.

[19] Y. Li and T. Harada, "Lepard: Learning partial point cloud matching in rigid and deformable scenes," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2022, pp. 5554–5564.

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

[21] Z. Chen, K. Sun, F. Yang, L. Guo, and W. Tao, "SC^2-PCr : Rethinking the generation and selection for efficient and robust point cloud registration," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 45, no. 10, pp. 12358–12376, Oct. 2023.

[22] H. Yu et al., "Rotation-invariant transformer for point cloud matching," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2023, pp. 5384–5393.

[23] G. Mei, X. Huang, L. Yu, J. Zhang, and M. Bennamoun, "COTReg: Coupled optimal transport based point cloud registration," 2021, *arXiv:2112.14381*.

[24] Z. J. Yew and G. H. Lee, "RPM-Net: Robust point matching using learned features," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 11824–11833.

[25] Y. Wang and J. M. Solomon, "Deep closest point: Learning representations for point cloud registration," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2019, pp. 3523–3532.

[26] 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 *Proc. Comput. Vis. ECCV: 16th Eur. Conf.*, 2020, pp. 378–394.

[27] W. Yuan, B. Eckart, K. Kim, V. Jampani, D. Fox, and J. Kautz, "DeepGMR: Learning latent Gaussian mixture models for registration," in *Proc. Comput. Vis., 16th Eur. Conf.*, 2020, pp. 733–750.

[28] C. Qiu, Z. Wang, X. Lin, Y. Zang, C. Wang, and W. Liu, "DSMNet: Deep high-precision 3D surface modeling from sparse point cloud frames," *IEEE Geosci. Remote Sens. Lett.*, vol. 20, pp. 1–5, 2023.