Robust Partial-to-Partial Point Cloud Registration in a Full Range
Liang Pan, Zhongang Cai, Graduate Student Member, IEEE, and Ziwei Liu

Abstract—Registration of 3D objects from point clouds is a challenging task due to sparse and noisy measurements, incomplete observations, and large transformations. In this work, we propose the Graph Matching Consensus Network (GMCNet) to estimate faithful correspondences for full-range Partial-to-Partial point cloud registration (PPR) in object-level registration scenarios. To encode robust point descriptors, we employ a novel Transformation-robust Point Transformer (TPT) module to adaptively aggregate local features with respect to the structural relations, taking advantage of both handcrafted rotation-invariant (RI) features and noise-resilient spatial coordinates. Based on the synergy of hierarchical graph networks and graphical modeling, we propose the Hierarchical Graphical Modeling (HGM) architecture to encode robust descriptors comprising of i) a unary term learned from RI features, and ii) multiple smoothness terms encoded from neighboring point relations at different scales through our TPT modules. Extensive experiments show that GMCNet outperforms previous state-of-the-art methods for PPR.

Index Terms—Deep Learning for Visual Perception, Visual Learning.

I. INTRODUCTION

POINT cloud registration (PCR) usually estimates a rigid transformation (i.e., a 3D rotation and a 3D translation) to align a source point cloud to a target point cloud, which serves as a fundamental research task for a wide range of robot vision applications, such as localization [1], 3D reconstruction [2] and 6D object pose estimation [3]. For instance, PCR has the potential to empower robots to accurately determine their current position and pose, thereby enhancing their interactions with the 3D world. In the context of object 6D pose estimation, this involves registering the observed partial point cloud of a certain object with the corresponding points sampled from its template CAD model surfaces. Subsequently, a robot arm can intelligently choose an optimal method for grasping the object.

A series of learning-based PCR methods [4], [5], [6], [7], [8] have been proposed. In the object-centric PCR scenario, Wang et al. [4], [5] learn to encode point features for estimating a mapping between the source point cloud and the target from ModelNet40 dataset [9]. Instead of directly taking 3D coordinates as inputs, recent PCR methods [6], [7], [10] provide better registration results with the help of rotation-invariant (RI) features. However, they are mostly validated under restricted source-target transformations (i.e., rotations in [0, 45°]), while large transformations could largely increase their registration error.

In this work, we take one step further and focus on partial-to-partial point cloud registration (PPR) in the full-range (i.e., global transformations in SE(3)). Encoding robust and discriminative point feature descriptors against large transformations and noisy measurements could be the major difficulty in estimating correct correspondences for global registration. Motivated by the success of using RI features for unconstrained arbitrary rotations in 3D object classification [11], [12] and 3D molecule property prediction [13], we find RI features promising, given carefully designed network architectures, to outperform existing methods [6], [7], [10] for full-range PPR.

Equipped with RI features, we further represent point cloud as a set of features with graph structures, and the correspondence estimation for PPR is reformulated as a maximum common subgraph prediction problem. Specifically, we propose a one-shot paradigm, Graph Matching Consensus Network (GMCNet), which estimates faithful correspondences features and multi-scale graph structures for full-range PPR. In order to encode large-scale distribution features, we enlarge the receptive fields by consecutively sub-sampling point clouds in a hierarchical architecture. To tackle the uncertainty introduced in the sub-sampling process, e.g Farthest Point Sampling (FPS) that leads to inconsistent encoded features, we employ a novel Transformation-robust Point Transformer (TPT) module, which adaptively aggregates neighboring features from sampled points with the help of both RI feature graphs and noise-resilient spatial relations. Following the idea of graphical modeling, we further propose the Hierarchical Graphical Modeling (HGM) architecture that encodes robust descriptors for each point consisting of 1) a unary term learned from RI features, and 2) multiple smoothness terms to encourage spatially-smooth correspondences with multi-scale geometric distribution-level features.

Experimental results on ModelNet40 (CAD models), MVP-RG (virtual scans), and SONN-RG (real scans), show that GMCNet outperforms previous state-of-the-art (SoTA) methods.
Qualitative results for full-range PPR on ModelNet40 by various methods are shown in Fig. 1, where GMCNet achieve accurate registration despite noisy and partial observation. We highlight that GMCNet employs a one-shot paradigm for PPR, and it encodes robust descriptors for each point cloud individually without requiring cross-contextual information.

Our key contributions are summarized as: 1) We provide a comprehensive analysis of transformation-robustness and noise-resilience of different geometric features, paving the way to encode robust descriptors for global PPR. 2) Employing graphical modeling in the context of deep learning, we propose GMCNet with novel modules, such as TPT and HGM, to achieve robust correspondence estimation for registration. 3) Extensive experimental results show that GMCNet could achieve much better registration results than previous SoTA methods, especially for full-range PPR.

II. RELATED WORK

Rotation-Invariant (RI) Features: Learning RI features has been extensively studied. By defining local reference frames (LRF), many descriptors [14, 15] accumulate measurements into histograms according to spatial coordinates, surfaces, and normals, and curvatures. However, it is challenging to achieve rotation-invariance by defining unambiguous LRFs. Without relying on LRFs, researchers designed handcrafted RI features, such as PPF [11], PFH [16] and FPFH [17]. With the help of deep learning, robust RI descriptors can be encoded based on the handcrafted features for downstream tasks [18], [19], [20]. For example, networks designed with RRI features show impressive performance for classification [12] and registration [10].

Point Cloud Registration (PCR): Conventional PCR methods mainly use the ICP [21] and its variants [22], [23]. Recently, Yang et al. introduced TEASER++ [24], which could efficiently solve semidefinite programs, and achieve fast and certifiable PCR. Many learning PCR methods could also achieve impressive PCR results. USIP [25] shows that it is possible to detect stable feature points from point clouds. A few learning-based methods, D3Feat [26], PointDSC [27], and HRegNet [28] utilize the learned feature detectors and descriptors to achieve PCR for large-scale scenes. The recent work GeoTransformer [29] could efficiently resolve PCR by using super-point matching. As for object-centric PCR, DCP [4] and PRNet [5] generate the estimated transformation by predicting soft assignments between two point sets with features encoded by DGCNN [30]. Other follow-up works [6], [7] iteratively resolve PPR with robust descriptors encoded by RI features. Storm [31] introduces an overlap prediction module with differentiable sampling to detect points in overlaps. Nonetheless, most existing methods [4], [5], [6], [7], [31], [32], [33], [34], [35], [36], [37], [38], [39] focus on local PCR, while overlooking global PCR (i.e., rotations in [0, 180°]). Additionally, they prefer to learn point descriptors without using hierarchies for sparse object-centric PCR. DeepGMR [10] addresses full-range PCR for complete point clouds but is inapplicable for PPR.

III. PROBLEM ANALYSIS

PPR Problem: Given a source point cloud (N points) \( \mathcal{X} = \{x_1, \ldots, x_N\} \subset \mathbb{R}^3 \) and a target point cloud (M points) \( \mathcal{Y} = \{y_1, \ldots, y_M\} \subset \mathbb{R}^3 \), PCR targets at aligning \( \mathcal{X} \) to \( \mathcal{Y} \), i.e. mapping each \( x_i \) to \( y_{m(x_i)} \), by finding a rigid transformation \( T_{\mathcal{X}\mathcal{Y}} \in \text{SO}(3) \) and a translation \( t_{\mathcal{X}\mathcal{Y}} \in \mathbb{R}^3 \). If \( \mathcal{X} \) and \( \mathcal{Y} \) are two different partial point clouds with overlapped areas, it becomes the more challenging PPR problem, as more outliers are introduced by their different partial observations.

Pose-Invariant Features: RRI (Fig. 2(a)) achieves pose-invariance for complete point clouds by aligning the same shape center. However, different partial point clouds mostly have inconsistent shape centers caused by different missing parts, which challenges the pose-invariance of RRI. In contrast, PPF (Fig. 2(b)) and FPFH [17] encodes pose-invariant geometric properties relying on the presence of consistent surface normal estimations. Although we can estimate surface normals on the fly, the surface normal direction ambiguities (red boxes) and 3D shape incompleteness (yellow boxes) make the normal estimation inconsistent (Fig. 2(c)).

Transformation-Robustness: We conduct preliminary studies (Fig. 3) on the ModelNet40 test set with 2,468 complete point clouds (2,048 points) to evaluate the transformation-robustness and noise-resilience of different features. For rotation-robustness, a rotation axis (e.g., Z-axis) is fixed, and then we gradually increase the rotation magnitude from 0° to 180°. RRI, PPF and FPFH are invariant against arbitrary rotations (see red line), but the absolute position “XYZ” and the relative position “\( \Delta XYZ \)” are highly influenced by the
increment in the rotation magnitude. Furthermore, concatenating the \( R \) features, such as \( PPF \), with “\( XYZ \)" and “\( \Delta XYZ \)" can highly increase the rotation-robustness (yellow dash line).

However, if adding an arbitrary translation, the similarity scores “\( PPF-SE(3) \)" and “\( RRI-SE(3) \)" will be influenced (i.e. < 1) due to ambiguous surface normal directions and mismatched shape centers, respectively. \( FPFH \) is similar with \( PPF \), and hence we omitted it in Fig. 3(a).

Nonetheless, handcrafted \( RI \) features can still describe \( RI \) geometric properties for the corresponding observation, because “\( PPF-SE(3) \)" and “\( RRI-SE(3) \)" are invariant to arbitrary rotations (two horizontal lines in Fig. 3(a)).

Noise-Resilience: As for noise-resilience test (Fig. 3(b)), we add a random Gaussian noise \( \mathcal{N}(0, \sigma^2) \) to each 3D point, and gradually increase the noise magnitude upper bound from 0 to 0.1. The mean absolute error indicates that both “\( XYZ \)" and “\( \Delta XYZ \)" features may exhibit greater noise resilience compared to handcrafted \( RI \) features, suggesting the robustness of structural relations against noise.

Consequently, it is promising to leverage both structural relations (e.g., “\( XYZ \)"") and handcrafted \( RI \) features (e.g., \( PPF \)) to encode transformation-robust and noise-resilient descriptors for global \( PPR \).

IV. OUR APPROACH

We represent point cloud as a list of neighborhood graphs \( \{\mathcal{N}\} \), and each \( \mathcal{N} = (\mathcal{V}, \mathcal{E}, \mathbf{X}) \) consists of a node set \( \mathcal{V} \) for spatial coordinates, an edge set \( \mathcal{E} \) for neighboring connections and a node feature set \( \mathbf{X} \) for encoded descriptors. Consequently, the two partial point clouds \( \mathcal{X} \) and \( \mathcal{Y} \) are represented as two lists of neighborhood graphs \( \{\mathcal{N}_s = (\mathcal{V}_s, \mathcal{E}_s, \mathbf{X}_s)\} \) and \( \{\mathcal{N}_t = (\mathcal{V}_t, \mathcal{E}_t, \mathbf{X}_t)\} \), and estimating faithful correspondences between \( \mathcal{X} \) and \( \mathcal{Y} \) can be reformulated as a maximal common graph estimation problem between \( \{\mathcal{N}_s\} \) and \( \{\mathcal{N}_t\} \) by learning a mapping \( m(\mathcal{X}, \mathcal{Y}) \) based on the encoded structural-aware features \( \Theta_Y \in \mathbb{R}^{N \times C} \) and \( \Theta_X \in \mathbb{R}^{M \times C} \): \( m(x_i, \mathcal{Y}) = \mathcal{F}_{map}[\text{softmax}(\Theta_Y \cdot \Theta_X^T)] \), where \( \mathcal{F}_{map}[\cdot] \) denotes a learnable mapping, and \( C \) is channel size.

A. Transformation-Robust Point Transformer

Recently, many research works have learned to utilize attention-based or transformation operations for point cloud analysis, which could show great potential towards robustness to various corruptions, such as noise and incompleteness [40]. To encode robust point feature descriptors, we propose a novel Transformation-robust Point Transformer (TPT) module. Similar to previous methods [41], [42], TPT adaptively aggregates local features by considering both point feature similarities and their spatial positions for each neighborhood graph \( \mathcal{N} \).

In contrast to those transformer operations, TPT emphasizes the pose-robustness for the aggregated point features by using handcrafted \( RI \) features in the positional encoding. Notably, the neighborhood relationships \( \mathcal{E} \) in point clouds are also pose-invariant. The attention-based aggregation layer of TPT (Fig. 4) can be formulated as follows:

\[
\begin{align*}
\mathbf{x}'_j = \sum_{i \in \mathcal{N}(i)} (\beta(\mathbf{x}_j) + \eta(\rho_{ij})) \odot \alpha \left( \mathbf{x}_{\mathcal{N}(i)}, \eta(\rho_{\mathcal{N}(i)}) \right)_j,
\end{align*}
\]

where \( \beta(\cdot) \) and \( \eta(\cdot) \) are shared Multi-Layer-Perceptrons (MLPs), \( \mathcal{N}(i) \) is the K-Nearest Neighboring (K-NN) point graph constructed in the spatial coordinates for the center point \( x_i \), \( \beta(x_i) \) is the transformed features, \( \eta(\rho_{ij}) \) is the positional embedding, the attention prediction function \( \alpha(\mathbf{x}_{\mathcal{N}(i)}, \eta(\rho_{\mathcal{N}(i)})) \) could compute the attention weight \( w_{ij} \) between \( x_i \) and \( x_j \) as \( w_{ij} = \eta(\delta(\mathbf{x}_{\mathcal{N}(i)}, \eta(\rho_{\mathcal{N}(i)}))) \) and \( \odot \) is element-wise product.

In view that vector attention operators [43] often achieve better performance than scalar attention operators, we compute vector attention (denoted as \( \alpha(\cdot) \)) that adapts to different feature channels, which consists of a relation operator \( \delta(\cdot) \) and a mapping function \( \gamma(\cdot) \):

\[
\begin{align*}
\delta(\mathbf{x}_{\mathcal{N}(i)}, \eta(\rho_{\mathcal{N}(i)})) = \left[ [\zeta(\mathbf{x}_i) - \zeta(x_j) + \eta(\rho_{ij})]_{ij \in \mathcal{N}(i)} \right],
\end{align*}
\]

where \( \zeta(\cdot) \) and \( \xi(\cdot) \) are shared MLPs, and \( \delta \) combines all feature vectors by concatenation, which is equivalent to the “Reshape” operation in Fig. 4. Afterward, \( \gamma(\cdot) \) further transforms the set of vectors and then maps to the right dimensionality. Particularly, we carefully design the positional embedding for describing
the positional relations between pairwise connected points by concatenating the absolute positions, relative positions, and handcrafted \(RI\) features (e.g. \(RRI\) and \(PPF\)). Although adding absolute and relative positions may influence the pose-invariance, it further increases PPR performance against noise.

### B. Hierarchical Graphical Modeling

Many existing object-centric PCR methods typically eschew the utilization of a hierarchical architecture due to large sparsity and subsampling randomness. In contrast, we propose a Hierarchical Graphical Modeling (HGM) architecture to encode multi-scale distribution features for constructing robust point feature descriptors for the source and the target individually. As shown in Fig. 5, the constructed descriptors for each point consist of 1) a unary term focusing on each node learned by handcrafted \(RI\) feature graphs; 2) multiple smoothness terms focusing on 3D geometric distributions encoded from multi-scale structural relations using the proposed TPT modules. Consequently, the encoded descriptors for a point \(x_i\) can be formulated as:

\[
\Theta_{x_i} = \left[ \Omega \left( \mathbf{x}^{RI}_{N_1(i)} \right), \Lambda^1 \left( \mathbf{x}^{N_1(i)} \right), \Lambda^2 \left( \mathbf{x}^{N_2(i)} \right), \Lambda^3 \left( \mathbf{x}^{N_3(i)} \right) \right],
\]

where \(\Theta_{x_i}\) concatenates the unary term \(\Omega(\cdot)\) encoded by \(RI\) features \(\mathbf{x}^{RI}_{N_1(i)}\), and the smoothness terms \(\Lambda^1(\cdot), \Lambda^2(\cdot)\) and \(\Lambda^3(\cdot)\). \(N_1(i), N_2(i)\) and \(N_3(i)\) are neighborhood graphs at different scales. Using the defined point descriptors in (3) for the source and the target point clouds, the mapping function \(m\) estimation can be formulated as an assignment problem with the following objective function:

\[
E(m) = \sum_{x_i \in X} \left( \lambda_u E_u \left( \Omega(\mathbf{x}^{RI}_{N_1(i)}), \Omega(\mathbf{y}^{RI}_{N_1(i)}) \right) \right) + \sum_{n} \lambda_{s_n} E_{s_n} \left( \Lambda^n(\mathbf{x}^{N_n(i)}), \Lambda^n(\mathbf{y}^{N_n(i)}) \right),
\]

where \(E_u(\cdot)\) denotes the unary penalty term, \(E_{s_n}(\cdot)\) denotes each smoothness penalty term, and \(\lambda(\cdot)\) is the corresponding weights for each term.

Unlike previous methods [7], [18] that only use the nearest neighbor as the paired points to encode handcrafted \(RI\) features, we consider each point as a centroid that is paired with more neighboring points for \(\Omega(\cdot)\). We highlight that our unary term \(\Omega(\cdot)\) rigorously preserves rotation-invariance, which can resolve full-range PCR without smoothness terms if the measurements are clean and consistent. The smoothness terms \(\Lambda^n(\cdot)\) is transformation-robust with the help of \(RI\) features, which highly increases noise-resilience for optimizing the mapping function \(m\).

### C. Optimization

**Mapping Function Optimization:** We penalize the mismatching loss by encouraging correspondences with similar feature descriptors. For each potential mapping \(m(x_i, y_j)\), we compute element-wise feature distances between corresponding terms from \(\Theta_x\) and \(\Theta_y\), and its objective function (shown in (4)) can be formulated as:

\[
E(m_{ij}) = \sum_{C} \left( \lambda_u \left( \|\Omega(\mathbf{x}^{RI}_{N_1(i)}) - \Omega(\mathbf{y}^{RI}_{N_1(j)})\|_2^2 / \sqrt{|\mathbf{C}_u|} \right) \right.
\]

\[
+ \sum_{n} \left( \lambda_{s_n} \left( \|\Lambda^n(\mathbf{x}^{N_n(i)}) - \Lambda^n(\mathbf{y}^{N_n(j)})\|_2^2 / \sqrt{|\mathbf{C}_{s_n}|} \right) \right),
\]

where \(|\mathbf{C}_u|\) and \(|\mathbf{C}_{s_n}|\) are the feature size for each corresponding feature item, which adjusts their importance. Hence, \(m(x_i, y_j)\) is initialized as \(\exp(-E(m_{ij}))\). Thereafter, we use Sinkhorn normalization layers for outlier rejection [7], [45], [46]. With the optimized matching matrix \(m^*\), the cross-covariance matrix \(\mathbf{H} = \sum_{i=1}^{N} \sum_{j=1}^{M} c_i \cdot (x_i - \bar{x}) \cdot (y[m^*(x_i)] - \bar{y})^\top\), where \(\bar{x} = \sum_{i=1}^{N} c_i \cdot x_i\), \(\bar{y} = \sum_{i=1}^{N} \sum_{j=1}^{M} c_i \cdot y[m^*(x_i)]\), \(m^*(x_i) = \sum_{j=1}^{M} c_j \cdot m^*(x_i, y_j)\), and \(c_i = \sum_{j=1}^{M} m^*(x_i, y_j)\), which denotes the confidence for point \(x_i\) is an inlier. In the end, \(\mathbf{R}_{XY}\) and \(\mathbf{t}_{XY}\) can be resolved by a SVD. Consequently, our GMCNet framework overview can be illustrated in Fig. 6.

**Training Loss:** GMCNet is trained end-to-end. Following [7], the training loss consists of two parts, a registration loss \(\mathcal{L}_{\text{reg}}\) and an auxiliary loss \(\mathcal{L}_{\text{aux}}\) to encourage inliers. In addition, we further improve registration estimation by using a cycle loss function that is formulated as:

\[
\mathcal{L} = \mathcal{L}_{\text{reg}} + \omega \mathcal{L}_{\text{aux}} + \mathcal{L}_{\text{cycle}} = \mathcal{L}_{\text{cycle}} + \omega(\mathcal{L}_{\text{inlier}} + \mathcal{L}_{\text{cyclic}}),
\]

Fig. 5. Hierarchical Graphical Modeling for Encoding Transformation-Robust Feature Descriptors. Following the graphical modeling structure, each point descriptor consists of 1) a “Unary Term” learned by the rotation-invariant feature graphs; 2) and multiple “Smoothness Terms” encoded from neighboring geometric features at different scales.
ON and We evaluate GMCNet on partial point clouds that evaluates reg-
L Following previous methods, we add Gaussian
1 [48] (L | to minimize the distance
Y−m 7 [10] Following
L M = RRR·(I
M [10] rotations (i.e. [0, 45°]) and
II
L M m L and clipped
M−
[7] PPF,
[49] I T restricted
X EXPERIMENTS
(1
1 L X− is our
L (1 = N contains CAD models of 40 categories,
L [4] 1 to implement normal esti-
N with an
N I =
[47] L [0, 180°) PPR.
U FPF features,
38x50] is isotropic rotation errors
most experiments.
38x73] initial learning rate 0.001 on an NVIDIA TITAN Xp GPU for
38x109 towards the coordinate origin of the posed partial point cloud.
38x121 PyTorch. We use Open3D library
38x157 and a random rotation magnitude in [0, 180°] defined by an arbitrary axis
38x169 unrestricted rotations (i.e. SO(3)) defined by an arbitrary axis
38x181 datasets. We evaluate both
38x193 from the ModelNet40
38x284 Our networks are implemented using
38x297 the root mean square error
38x313 where
38x326 inlier
38x339 T
38x345 ⋆ (x
38x347 y
38x349 j )+
38x352 m (x
38x354 y
38x356 j ),
38x360 m (x
38x362 y
38x364 j )
38x366 and
38x373 two HGM share weights.
38x382 dences using the robust feature descriptors encoded by our HGM. Note that the
38x391 Fig. 6. Framework Overview. GMCNet estimates pose-invariant correspon-
38x393 TABLE I
38x429 TABLE I
38x431 A. Registration on ModelNet40
38x454 ModelNet40 [9] contains CAD models of 40 categories, which is split into 9,843 shapes for training and 2,468 for testing. For each 3D shape, we uniformly sample 1,024 points from its surfaces. To imitate partial scans, incomplete point clouds are generated by selecting the nearest 768 points for a randomly placed point in the space [5]. Following previous works [4], [7], [10], we use four different data settings: 1) “Clean”, 2) “Unseen”, 3) “Noisy” and “Noisy & Unseen”, which are reported in Table I and Table II, respectively. The maximal translation between the paired point clouds is 0.5 along an arbitrary direction. To achieve fair comparisons, we use the same rotation augmentations - that is [0, 45°] for restricted rotations and [0, 180°] for unrestricted, for all reported methods. For all the other settings, we use their official configurations on ModelNet40, which are provided by the authors. Qualitative registration results are shown in Fig. 7.

Unseen Shape PPR: We use point clouds from all 40 categories of ModelNet40 for training and testing. Note that unseen 3D objects are used for evaluation. As reported in Table I, GMCNet using different RI features, RRI, FPFH and PPF, achieves nearly perfect registration results, even for full-range (i.e. [0, 180°]) PPR.

Unseen Category PPR: In this setting, we train our network by using 3D point clouds from the first 20 categories and use the other 20 category point clouds for evaluation. Similar to the “Clean” setting, GMCNet significantly outperforms existing SoTA methods in Table I.

Noisy PPR: Following previous methods, we add Gaussian noise that is randomly sampled from \( N(0, 0.01) \) and clipped
TABLE II

|                      | Noisy | Noisy & Unseen |
|----------------------|-------|----------------|
|                      | $\ell_2$ | $\ell_4$ | $\ell_{\text{BASE}}$ | $\ell_2$ | $\ell_4$ | $\ell_{\text{BASE}}$ | $\ell_2$ | $\ell_4$ | $\ell_{\text{BASE}}$ |
| PRNet$^1$ [5]        | 4.37* | 0.034 | 0.045 | 95.80* | 0.319 | 0.542 | 8.47* | 0.061 | 0.081 |
| IDAM$^2$ (GGN) [6]  | 4.46* | 0.029 | 0.039 | 57.85* | 0.253 | 0.374 | 4.25* | 0.025 | 0.037 |
| IDAM$^2$ (FPFH) [6] | 9.60* | 0.052 | 0.084 | 71.06* | 0.217 | 0.430 | 9.50* | 0.051 | 0.087 |
| RGM$^2$ [8]         | 2.21* | 0.013 | 0.018 | 23.58* | 0.111 | 0.156 | 2.62* | 0.025 | 0.030 |
| Predator$^2$ [44]   | 3.33* | 0.018 | 0.026 | 40.64* | 0.110 | 0.207 | 2.91* | 0.017 | 0.024 |
| DCP [4]             | 9.33* | 0.070 | 0.097 | 73.61* | 0.185 | 0.441 | 10.58* | 0.072 | 0.109 |
| DeepGMR (RRI) [10]  | 16.96* | 0.068 | 0.120 | 68.68* | 0.248 | 0.419 | 17.67* | 0.067 | 0.131 |
| DeepGMR (XY2) [10]  | 6.48* | 0.049 | 0.064 | 70.26* | 0.246 | 0.428 | 7.55* | 0.052 | 0.071 |
| RPMNet (PPF) [7]    | 3.52* | 0.214 | 0.029 | 37.82* | 0.132 | 0.250 | 3.80* | 0.022 | 0.032 |
| PointNetLK (32)     | 5.29* | 0.037 | 0.055 | 82.30* | 0.265 | 0.510 | 4.86* | 0.036 | 0.053 |
| PointNetLK Revisited [35] | 8.63* | 0.060 | 0.086 | 82.81* | 0.267 | 0.519 | 8.55* | 0.059 | 0.085 |
| GeoTransformer [29] | 3.04* | 0.014 | 0.024 | 45.87* | 0.154 | 0.272 | 3.07* | 0.014 | 0.024 |
| GMCNet (RRI)        | 1.28* | 0.009 | 0.011 | 50.62* | 0.206 | 0.315 | 1.47* | 0.010 | 0.012 |
| GMCNet (FPFH)       | 3.18* | 0.008 | 0.010 | 25.95* | 0.115 | 0.184 | 3.33* | 0.009 | 0.011 |
| GMCNet (PPF)        | 0.94* | 0.007 | 0.008 | 18.13* | 0.093 | 0.132 | 1.12* | 0.008 | 0.010 |

$^1$ denotes ground truth correspondences or cross-contextual information are used during training. NC denotes Not-Converge, and it means the network does not converge during training.

Fig. 7. Qualitative Registration Results on ModelNet40 [9].

to $[-0.05, 0.05]$ to each 3D point. By adding random noise, raw handcrafted RI features become no longer transformation-invariant, which mostly influences the global registration performance. In Table II, GMCNet could resolve local PPR with small registration error, and also achieves better global registration with the help of PPF than previous SoTA methods. Those symmetric shapes and repetitive structures (e.g., vases and tables) of 3D shapes from the ModelNet40 dataset make the full-range PPR challenging to be perfectly resolved.

Unseen Category and Noisy PPR: The “Unseen & Noisy” setting is a combination of “Unseen” and “Noisy”: Novel category evaluation on noisy partial point clouds. Similar to the “Noisy” setting, the proposed GMCNet could resolve local registration very well, but also struggles with global registration. Even though, GMCNet achieves better PPR than previous SoTA methods for most evaluated metrics.

Different RI Features: In Tables I and II, we evaluated RRI, FPFH and PPF in GMCNet. Although RRI features are no longer consistent for clean partial point clouds under different SE(3) transformations due to center shifting, GMCNet estimate faithful correspondences between different partial point clouds for PPR. We think the reason is that the inconsistency caused by translations influences the encoded RRI features for all points, which is compensated during the feature matching optimization if perfect correspondences exist. Together with random noise, local registration with RRI features can be resolved mainly by using spatial coordinates graphs. However, it is too challenging to use RRI for global PPR with noisy data, while those local RI features, FPFH and PPF, can lead to better registration results.

Other Methods Using RI Features: Methods [6], [7], [10] also utilize RI features, but they do not achieve good performance for full-range PPR. DeepGMR [10] encounters challenges when registering point cloud pairs with different incompleteness, as the RRI features lose their invariance in the presence of different missing parts (please refer to our analysis in Sec. III). We speculate that IDAM [6] does not account for both rotation invariance and noisy resilience in their encoded features, which could significantly impact their correspondence estimation in cases involving substantial relative rotations and noisy measurements. RPMNet [7] requires a few iterations for registration, and we notice it converges very slowly, especially for full-range PPR. In contrast, we consider both rotation-invariance and noisy-resilience properties for encoding descriptors. Moreover, we utilize transformer-based operations and hierarchical graph architectures to achieve efficient PCR in a full range.

Inf. Time and Model Size: In Table IV, we report model details of GMCNet and previous representative methods. To achieve fair comparisons, we evaluate all the reported methods by using the same setting on an NVIDIA 3090 GPU. Among all the listed methods, GMCNet shows 2nd shortest inference time, 2nd smallest run-time memory, 4th smallest model size, and 4th least number of parameters.
TABLE III
ABLATION STUDIES FOR PPR ON MODELNET40

| K-NN TPT HGM | Clean (0°, 45°) | Noisy (0°, 180°) |
|-------------|----------------|-----------------|
|             | $L_{\text{RMSE}}$ | $L_{\text{RMSE}}$ | $L_{\text{RMSE}}$ |
| ✔           | 36.93° | 0.197 | 0.246 | 55.32° | 0.208 | 0.358 |
| ✔           | 5.45° | 0.086 | 0.068 | 23.74° | 0.123 | 0.173 |
| ✔           | 0.016° | <0.0001 | 0.001 | 18.13° | 0.093 | 0.132 |

TABLE IV
INFERENGE TIME AND MODEL SIZE

| Inference Time (ms) | Model Size (KB) | Runtime Memory (MB) | Number of Parameters (M) |
|---------------------|-----------------|---------------------|--------------------------|
| DCP [4]             | 15.04           | 21791               | 10192                    | 5.569 |
| DeepGMR (ORF) [10]  | 69.25           | 600                 | 3098                     | 1.527 |
| IDAM (GMM) [4]      | 27.48           | 400                 | 5072                     | 0.088 |
| PointNetLK [32]     | 46.22           | 615                 | 5992                     | 0.152 |
| PointNetLK Revisited [35] | 50.09        | 576                 | 3804                     | 0.143 |
| GeoTransformer [29] | 55.49           | 20448              | 5190                     | 5.210 |
| GMCNet (PPF)        | 23.09           | 1148               | 3586                     | 0.283 |

B. Registration on MVP-RG

We further evaluate GMCNet on the MVP-RG dataset consisting of view-based partial point clouds. In the MVP-RG dataset, each partial point cloud is generated by projecting the scanned depth map to the image coordinate frame and followed by an FPS to 2,048 points. Afterward, we select paired partial point clouds for the same object if sufficient overlapping areas are detected. In total, our MVP-RG dataset consists of 7,600 partial point cloud pairs from 16 categories, which is split into a training set (6,400 samples) and a testing set (1,200 samples). Furthermore, we randomly transform the 3D object for each observation, hence making it a full-range PPR problem. Note that the partial point cloud pairs do not have perfect correspondences since different partial point clouds are generated by virtual scans from different camera views. To achieve fair comparisons, we use the same rotation augmentations for all methods and use their best settings that previously were used for PPR with noisy data from ModelNet40. Similarly, it is very challenging to register noisy partial point clouds from MVP-RG. In Table V, GMCNet also outperforms previous SoTA methods on MVP-RG. Qualitative registration results are shown in Fig. 8.

C. Registration on SONN-RG

To evaluate object-centric registration on real-scanned data, we establish a registration dataset (SONN-RG) based on the ScanObjectNN dataset [47] consisting of point clouds scanned by an RGBD camera (Fig. 9(a)). The training set of SONN-RG consists of 11,481 pairs of point clouds, and the test set has 2,894 pairs. Each point cloud has 2048 points. The initial relative transformations are randomly generated similar to the settings on ModelNet40. As shown in Fig. 9(b), the source (object points only) and the target (w/background points) point clouds are generated by separate sampling processes, which does not guarantee perfect correspondences. Experimental results (in Table VI) show that GMCNet still surpasses previous SoTA methods.

VI. CONCLUSION

In this letter, we propose a comprehensive paradigm GMCNet, which utilizes a synergy of hierarchical graph networks and graphical modeling for object-centric PPR. In particular, we propose novel modules, such as TPT and HGM modules to adaptively aggregate neighboring features. Extensive experiments show that GMCNet outperforms previous SoTA methods, especially for full-range registration.
REFERENCES

[1] L. Ding and C. Feng, “Deepmapping: Unsupervised map estimation from multiple point clouds,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2019, pp. 8650–8659.

[2] W. Lu, Y. Zhou, G. Wan, S. Hou, and S. Song, “L3-Net: Towards learning based lidar localization for autonomous driving,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2019, pp. 6389–6398.

[3] M. Tian, L. Pan, M. H. Ang, and G. H. Lee, “Robust 6D object pose estimation by learning RGB-D features,” in Proc. IEEE Int. Conf. Robot. Automat., 2020, pp. 6218–6224.

[4] Y. Wang and J. M. Solomon, “Deep closest point: Learning representations for point cloud registration,” in Proc. IEEE Int. Conf. Comput. Vis., 2019, pp. 3523–3532.

[5] Y. Wang and J. M. Solomon, “PRNet: Self-supervised learning for partial-to-partial registration,” in Adv. Neural Inf. Process. Syst., 2019, pp. 8812–8824.

[6] 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. Eur. Conf. Comput. Vis., 2020, pp. 378–394.

[7] 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.

[8] K. Fu, S. Liu, X. Luo, and M. Wang, “Robust point cloud registration framework based on deep graph matching,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit., 2021, pp. 8893–8902.

[9] Z. Wu et al., “3D ShapeNets: A deep representation for volumetric shapes,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2015, pp. 1912–1920.

[10] W. Yuan, B. Eckart, K. Kim, V. Jammpani, D. Fox, and J. Kautz, “DeepGMRF: Learning latent gaussian mixture models for registration,” in Proc. Eur. Conf. Comput. Vis., 2020, pp. 733–750.

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

[12] C. Chen, G. Li, R. Xu, T. Chen, M. Wang, and L. Lin, “Clusternet: Deep hierarchical cluster network with rigorously rotation-invariant representation for point cloud analysis,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit., 2019, pp. 4994–5002.

[13] F. Fuchs, D. Worrall, V. Fischer, and M. Welling, “Se(3)-transformers: 3D rototranslation equivariant attention networks,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit., 2021, pp. 12763–12772.

[14] D. Bauer, T. Patten, and M. Vincze, “Reagent: Point cloud registration using imitation and reinforcement learning,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit., 2021, pp. 14586–14594.

[15] J. Xu, Y. Zhang, Y. Zou, and P. X. Liu, “Point cloud registration with zero overlap rate and negative overlap rate,” IEEE Robot. Automat. Lett., vol. 8, no. 3, pp. 6643–6650, Oct. 2023.

[16] H. Zhao, Z. Liang, Y. He, C. Wang, and M. Yang, “Cy-CNN: Cylinder convolution based rotation-invariant neural network for point cloud registration,” Sci. China Inf. Sci., vol. 66, no. 5, pp. 1–15, 2023.

[17] Z. Wang, Z. Qi, Q. Peng, Z. Wu, and Z. Zhu, “Robust point cloud registration using geometric spatial refinement,” IEEE Robot. Automat. Lett., vol. 8, no. 7, pp. 4171–4178, Jul. 2023.

[18] J. Ren, L. Pan, and Z. Liu, “Benchmarking and analyzing point cloud classification under corruptions,” in Proc. Int. Conf. Mach. Learn., 2022, pp. 18559–18575.

[19] R. Santen, C. Stroud, A. Viefers, and P. H. Torr, “Point cloud classification under corruptions,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit., 2022, pp. 18524–18533.

[20] H. Zhao, L. Jiang, J. Jia, P. H. Torr, and V. Koltun, “Point transformer,” in Proc. IEEE/CVF Int. Conf. Comput. Vis., 2021, pp. 16259–16268.

[21] H. Zhao, J. Jia, and V. Koltun, “Exploring self-attention for image recognition,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit., 2020, pp. 10076–10085.

[22] S. Huang, Z. Goeic, M. Usyvetsov, 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.

[23] R. Santa Cruz, B. Fernando, A. Cheriaim, and S. Gould, “Deeppermut: Visual permutation learning,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2017, pp. 3949–3957.

[24] F. E. Sarlin, D. DeTone, T. Malisiewicz, and A. Rabinovich, “Superpoint: Learning feature matching with graph neural networks,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit., 2020, pp. 4938–4947.

[25] M. A. Uy, Q.-H. Pham, B.-S. Hua, T. Nguyen, and S.-K. Yeung, “Revisiting point cloud classification: A new benchmark dataset and classification model on real-world data,” in Proc. IEEE/CVF Int. Conf. Comput. Vis., 2019, pp. 1588–1597.

[26] Q.-Y. Zhou, J. Park, and V. Koltun, “Open3D: A modern library for 3D data processing,” 2018, arXiv:1801.09847.

[27] D. Kingma and J. Ba, “Adam: A method for stochastic optimization,” in Proc. Int. Conf. Learn. Representations, San Diego, CA, USA, Dec. 2015.