Overlap-guided Gaussian Mixture Models for Point Cloud Registration

Guofeng Mei
University of Technology Sydney
Sydney, Australia
guofeng.mei@student.uts.edu.au

Fabio Poiesi
Fondazione Bruno Kessler
Trento, Italy
poiesi@fbk.eu

Cristiano Saltori
University of Trento
Trento, Italy
cristiano.saltori@unitn.it

Jian Zhang
University of Technology Sydney
Sydney, Australia
jian.zhang@uts.edu.au

Elisa Ricci
University of Trento & FBK
Trento, Italy
e.ricci@unitn.it

Nicu Sebe
University of Trento
Trento, Italy
niculae.sebe@unitn.it

Abstract

Probabilistic 3D point cloud registration methods have shown competitive performance in overcoming noise, outliers, and density variations. However, registering point cloud pairs in the case of partial overlap is still a challenge. This paper proposes a novel overlap-guided probabilistic registration approach that computes the optimal transformation from matched Gaussian Mixture Model (GMM) parameters. We reformulate the registration problem as the problem of aligning two Gaussian mixtures such that a statistical discrepancy measure between the two corresponding mixtures is minimized. We introduce a Transformer-based detection module to detect overlapping regions, and represent the input point clouds using GMMs by guiding their alignment through overlap scores computed by this detection module. Experiments show that our method achieves superior registration accuracy and efficiency than state-of-the-art methods when handling point clouds with partial overlap and different densities on synthetic and real-world datasets. https://github.com/gfmei/ogmm

1. Introduction

With the rise of inexpensive 3D sensors for both indoor (e.g. Microsoft Kinect) and outdoor (e.g. LiDAR) scenes, the point cloud has become an important data source, which presents rich 3D spatial information efficiently. To produce large-scale point clouds, 3D point cloud registration has been widely investigated. The point cloud registration refers to the problem of finding a rigid relative pose transformation that aligns a pair of point clouds into the same coordinate frame [18, 28]. The registration quality can directly impact applications such as robotics [38, 55], augmenting reality [3], autonomous driving [30, 32], and radiotherapy [26, 27]. However, sensor noise, varying point densities, outliers, occlusions, and partial views are challenges that still affect the performance of these applications in the real world [21].

Point cloud registration approaches can be broadly categorized into correspondence-free and correspondence-based [28]. *Correspondence-free registration approaches* aim at minimizing the difference between the global features extracted from two input point clouds [28, 18, 1]. These global features are typically computed based on all the points of a point cloud, making correspondence-free approaches inadequate to handle scenes with partial overlaps, such as those captured in the real world [51, 6]. *Correspondence-based registration approaches* rely on point-level correspondences between two input point clouds [10, 39, 2]. Despite showing promising results, these approaches suffer from two major challenges: i) real-world point clouds do not contain exact point-level correspondences due to sensor noise and density variations [17, 49, 36]; ii) the size of the correspondence search space increases quadratically with the number of points of the two point clouds [49]. An alternative to finding point-to-point correspondences is using distribution-to-distribution matching through probabilistic models [17, 49]. These probabilistic registration techniques showed greater robustness to noise and density variations than their point-to-point counterpart [36], however, they typically require their inputs to share the same distribution parameters (e.g., Gaussian Mixture Models). Due to this, they can only handle complete-to-complete [49] or partial-to-complete [36] point cloud registration setups. Partial-to-partial setups, which are typical in real-world applications, may have disjoint distribution parameters. Therefore, when state-of-the-art approaches are used in these setups are highly likely to underperform.

In this paper, we propose an overlap-guided GMM-based registration method, named OGMM, to mitigate the limitations of partial-to-partial setups without using exact point-level correspondences. We reformulate the problem of registering point cloud pairs as the alignment of two Gaussian mixtures by minimizing a statistical discrepancy measure between the two corresponding mixtures. We introduce an overlap score to measure how likely points locate in the overlapping areas between source and target point clouds via a Transformer-based deep neural network. The input point cloud is modeled through GMMs under the guidance of this overlap score. The self-attention or cross-attention of Transformer networks come with computational and memory requirements that scale quadratically with the size of point clouds (N^2), hindering their applicability to large-scale point cloud datasets. Therefore, we introduce the idea of *clustered attention*, which is a fast approximation of self-attention. Clustered attention groups a set of points into J clusters and compute the attention for these clusters only, making the complexity linear with the number of clusters, i.e., $N \cdot J$, where $J << N$. OGMM is inspired by DeepGMR [49], but it differs from it in two ways. First, our probabilistic paradigm can handle partial-to-partial point cloud registration problems through the overlap score constraint. Second, our network learns a consistent GMM representation across feature and geometric space rather than fitting a GMM in a single feature space. We evaluate our approach on ModelNet40 [41], 7Scene [35], and ICL-NUIM [14], comparing our approach against traditional and deep learning-based point cloud registration approaches. We use the typical evaluation protocol for point cloud registration that is used in this literature [10]. OGMM achieves state-of-the-art results and largely outperforms DeepGMR on all the benchmarks.

In summary, the main contributions of this work are:

- We propose a learning-based probabilistic registration framework under the guidance of overlap scores;
- We propose a cluster-based overlap Transformer module to embed cross point cloud information that enables the detection of overlap regions;
- We introduce a cluster-based loss to ensure that our network learns a consistent GMM representation across feature and geometric space rather than fitting a GMM in a single feature space;
- We achieve state-of-the-art accuracy and efficiency on a comprehensive set of experiments, including synthetic and real-world datasets.

2. Related Work

We review both correspondence-free and correspondence-based point cloud registration methods, and Transformer-related works as it is a major component in our approach.

2.1. Correspondence-free Registration

The core idea of the correspondence-free registration approaches, such as PointNetLK [1] and FMR [18], is to first extract the global features from the source and target point clouds, and then regress the rigid motion parameters by minimizing the difference between global features of two input point clouds. Such methods do not require point-point correspondences and are robust to density variations. These methods use extracted global features to reduce the point cloud's dimension so that the algorithm's time complexity does not increase as the number of points increases. However, they highly depend on fairly high overlaps (more than 90%) between two point clouds and suffer from performance degradation in the case of partially-overlapping point clouds, which are typical in real-world scenarios [51, 6].

2.2. Correspondence-based Registration

2.2.1 Point-level Correspondence-Based Registration.

The most popular point-to-point registration algorithm is ICP [2], which alternates between rigid motion estimation and the correspondences searching [39, 19] by solving a L_2-optimization. However, ICP converges to spurious local minima due to the non-convexity nature of the problem. Based on this algorithm, many variants have been proposed. For example, the Levenberg-Marquardt ICP [9] uses a Levenberg-Marquardt algorithm to yield the transformation by replacing the singular value decomposition with gradient descent and Gaussian-Newton approaches, accelerating the convergence while ensuring high accuracy. Go-ICP [46] solves the point cloud alignment problem globally by using a Branch-and-Bound (BnB) optimization framework without prior information on correspondence or transformation. RANSAC-like algorithms are widely used for robust finding the correct correspondences for registration [24]. FGR [53] optimizes a Geman-McClure cost-induced correspondence-based objective function in a graduated nonconvex strategy and achieves high performance. TEASER [45] reformulates the registration problem as an intractable optimization and provides readily checkable conditions to verify the optimal solution. These methods still face challenges when point clouds are affected by noise, outlier, and density variations [51].

Recently, several deep features [18, 51] are proposed to accurately estimate the point-level correspondences. For instance, DCP [39] employs DGCNN [40] for feature extraction and an attention module to generate soft matching pairs. RPMNet [47] proposes a method to solve the point cloud partial visibility by integrating the Sinkhorn algorithm into a network to get soft correspondences from local features. Soft correspondences can increase the robustness of registration accuracy. RGM [10] utilizes a Transformer to aggregate information by generating soft graph edges for point-wise matching. IDAM [25] incorporates both geometric and distance features into the iterative matching process. RIENet [34] propose a method to identify the inlier based on the graph-structure difference between the neighborhoods. Although achieving remarkable performance, most of these approaches rely on point-to-point correspondences, and thus they are still sensitive to noise and density variations [17].

2.2.2 Probabilistic Registration.

The probabilistic registration methods model the distribution of point clouds as a density function often via the use of GMMs and perform alignment either by employing a correlation-based approach or using an EM-based optimization framework [49, 23]. A commonly used formulation, such as CPD [29] and FilterReg [12], represents the geometry of the target point cloud using a GMM distribution over 3D Euclidean space. The source point cloud is then fitted to the GMM distribution under the maximum likelihood estimation (MLE) framework. Another statistical approach, including GMMReg [20], JRMPC [8], and DeepGMR [49], builds GMM probability distribution on both the source and the target point clouds. These methods show robustness to outliers, noise, and density variation [49]. These models usually assume that both source and target point clouds share the same GMM parameters or that one of the point clouds is supposed to be "perfect", thus leading to biased solutions as long as both point clouds contain noise and outliers (such as partial-to-partial registration) [8]. Considering the above factors, we propose an overlap-guided 3D point cloud registration algorithm based on Gaussian Mixture Model (GMM) that can be applied to partial-to-partial registration problems.

2.3. Transformers in 3D Point Clouds

Due to the inherent permutation invariance and strong global feature learning ability, 3D Transformers are well suited for point cloud processing and analysis. They have surpassed state-of-the-art non-Transformer algorithm performance. In particular, A-SCN [42] is the first such example of a Transformer-based approach for point cloud learning. Later, PCT [13], which is a pure global Transformer network, generates positional embedding using 3D coordinates of points and adopts a transformer with an offset attention module to enrich features of points from its local neighborhood. PointTransformer [52] to construct self-attention networks for general 3D tasks with nearest neighbor search. However, they suffer from the fact that as the size of the feature map increases, the computing and memory overheads of the original Transformer increase quadratically. Efforts to reduce the quadratic complexity of attention mainly focused on self-attention. For instance, PatchFormer [50] reduces the size of the attention map by first splitting the raw point cloud into patches, and then aggregating the local feature in each patch to generate an attention matrix. FastPointTransformer proposes centroid-aware voxelization and devoxelization techniques to reduce space complexity. However, these works are less suitable for feature matching, where we are required to perform self- and cross-attention on features within and between point clouds, respectively. To this end, we propose clustered attention which groups a set of points into J clusters and computes the attention only for these clusters, resulting in linear complexity for a fixed number of clusters.

3. Our approach

Rigid point cloud registration aims to recover a rigid transformation matrix $T \in SE(3)$, which consists of rotation $R \in SO(3)$ and translation $\boldsymbol{t} \in \mathbb{R}^3$, that optimally aligns the source point cloud $\boldsymbol{\mathcal{P}} = \{\boldsymbol{p}_i \in \mathbb{R}^3 \big| i = 1, 2, ..., N\}$ to the target point cloud $\boldsymbol{\mathcal{Q}} = \{\boldsymbol{q}_j \in \mathbb{R}^3 \big| j = 1, 2, ..., M\}$, where N and M represent the number of points in $\boldsymbol{\mathcal{P}}$ and $\boldsymbol{\mathcal{Q}}$, respectively. Fig. 1 illustrates our framework that consists of three modules: feature extraction, overlap region detection, and overlap-guided GMM for registration. The shared weighted encoder first extracts point-wise features $\boldsymbol{\mathcal{F}}_p$ and $\boldsymbol{\mathcal{F}}_q$ from point clouds $\boldsymbol{\mathcal{P}}$ and $\boldsymbol{\mathcal{Q}}$, respectively. The clustered self-attention module then updates the point-wise features $\boldsymbol{\mathcal{F}}_p$ and $\boldsymbol{\mathcal{F}}_q$ to capture global context. Next, the overlap region detection module projects the updated features $\boldsymbol{\mathcal{P}}$ and $\boldsymbol{\mathcal{Q}}$ to overlap scores $\boldsymbol{o}_p, \boldsymbol{o}_q$, respectively. $\boldsymbol{\mathcal{F}}_p$, $\boldsymbol{\mathcal{F}}_q$, $\boldsymbol{o}_p$ and $\boldsymbol{o}_q$ are then used to estimate the distributions (GMMs) of $\boldsymbol{\mathcal{P}}$ and $\boldsymbol{\mathcal{Q}}$. Finally, weighted SVD is adopted to estimate the rigid transformation T based on the estimated distributions.

3.1. Feature Extraction

The feature extraction network consists of a Dynamic Graph Convolutional Neural Network (DGCNN), positional encoding, and a clustered self-attention network. Given a point cloud pair $\boldsymbol{\mathcal{P}}$ and $\boldsymbol{\mathcal{Q}}$, DGCNN extracts their associated features $\boldsymbol{\mathcal{F}}_p = \{\boldsymbol{f}_{p_i} \in \mathbb{R}^d \big| i = 1, 2, ..., N\}$ and $\boldsymbol{\mathcal{F}}_q = \{\boldsymbol{f}_{q_j} \in \mathbb{R}^d \big| j = 1, 2, ..., M\}$. Here, $d = 512$.

3.1.1 Attention module

Transformer training and inference in previous works can be computationally expensive due to the quadratic complexity of self-attention over a long sequence of representations, especially for high-resolution correspondences prediction tasks. To mitigate this limitation, our novel cluster-based Transformer architecture operates after local feature extraction: $\boldsymbol{\mathcal{F}}_p$ and $\boldsymbol{\mathcal{F}}_q$ are passed through the attention module to extract context-dependent point-wise features. Intuitively, the self-attention module transforms the DGCNN features into more informative representations to facilitate matching.

Spherical positional encoding. Transformers are typically fed with only high-level features, which may not explicitly encode the geometric structure of the point cloud [39, 16]. This makes the learned features geometrically less discriminative, causing severe matching ambiguity and numerous outlier matches, especially in low-overlap cases [33]. A straightforward recipe is to explicitly inject positional encoding of 3D point coordinates, which assigns intrinsic geometric properties to the per-point feature by adding unique positional information that enhances distinctions among point features in indistinctive regions [52]. However, the resulting coordinate-based attentions are naturally transformation-variant [33], while registration requires transformation invari-

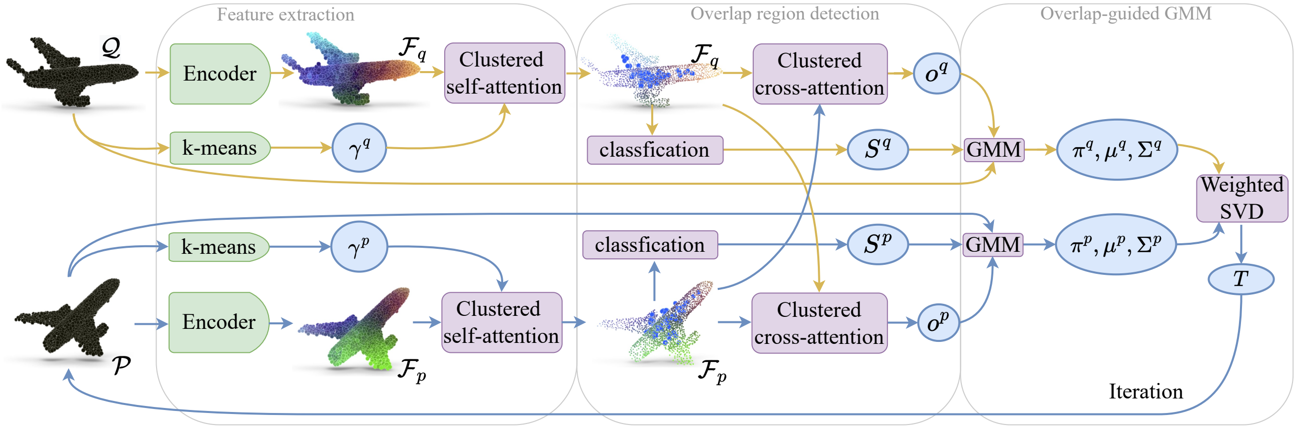

Figure 1. OGMM consists of three modules: feature extraction, overlap region detection, and overlap-guided GMM for registration. The shared weighted encoder extracts point-level features $\boldsymbol{\mathcal{F}}_p$ and $\boldsymbol{\mathcal{F}}_q$ from point clouds $\boldsymbol{\mathcal{P}}$ and $\boldsymbol{\mathcal{Q}}$, respectively. The self-attention module updates the point-wise features $\boldsymbol{\mathcal{F}}_p$ and $\boldsymbol{\mathcal{F}}_q$. The overlap region detection module projects the updated features $\boldsymbol{\mathcal{P}}$ and $\boldsymbol{\mathcal{Q}}$ to overlap scores $\boldsymbol{o}_p, \boldsymbol{o}_q$, respectively. $\boldsymbol{\mathcal{F}}_p$, $\boldsymbol{\mathcal{F}}_q$, $\boldsymbol{o}_p$ and $\boldsymbol{o}_q$ are used to estimate GMMs of $\boldsymbol{\mathcal{P}}$ and $\boldsymbol{\mathcal{Q}}$. The weighted SVD is used to estimate the rigid transformation T based on the estimated distributions.

ance since the input point clouds can be in arbitrary poses. To this end, we design spherical positional encoding, which leverages the distances and angles computed with the points, to encode the transformation-invariant geometric information of the points. Specifically, given a point $\boldsymbol{p}_i \in \boldsymbol{\mathcal{P}}$, we select the $k > 0$ nearest neighbors $\mathcal{K}_i$ of $\boldsymbol{p}_i$ and compute the centroid $\boldsymbol{p}_c = \sum_{i=1}^{N} \boldsymbol{p}_i$ of mass of $\boldsymbol{\mathcal{P}}$. For each $\boldsymbol{p}_x \in \mathcal{K}_i$, we denote the angle between the vectors $\boldsymbol{p}_i - \boldsymbol{p}_c$ and $\boldsymbol{p}_x - \boldsymbol{p}_c$ as α_{ix}. The positional encoding $\boldsymbol{f}_{\boldsymbol{p}_i}^{pos} \in \mathbb{R}^d$ of $\boldsymbol{p}_i$ is as

$$\boldsymbol{f}_{\boldsymbol{p}_i}^{pos} = \varphi\left(\|\boldsymbol{p}_i - \boldsymbol{p}_c\|_2\right) + \max_{x \in \mathcal{K}_i}\{\phi\left(\alpha_{ix}\right)\}, \tag{1}$$

where φ and ϕ are two MLPs, and each MLP consists of a linear layer and one ReLU nonlinearity function [43]. Our positional encoding is invariant to rigid transformation since the distance and angles are invariant to transformation. We then updates features $\boldsymbol{\mathcal{F}}_{\boldsymbol{p}}$ of $\boldsymbol{\mathcal{P}}$ by $\boldsymbol{\mathcal{F}}_{\boldsymbol{p}} = \{\boldsymbol{f}_{\boldsymbol{p}_i}^{pos} + \boldsymbol{f}_{\boldsymbol{p}_i}\}$. The same operation is also applied in $\boldsymbol{\mathcal{Q}}$.

Cluster-based self-attention. The extracted local features have a limited receptive field which may hinder the distinctiveness of regions. Instead, humans can find correspondences between these regions not only based on the local neighborhood structures but also by exploiting context. Self-attention is thus introduced to model global structures by establishing long-range dependencies. We exploit this idea to improve the computational complexity of self-attention. Specifically, we first use Wasserstein K-Means [11] to cluster point cloud $\boldsymbol{\mathcal{P}}$ (or $\boldsymbol{\mathcal{Q}}$) into J non-overlapping clusters in geometric space by $\boldsymbol{\gamma}^p \in \{0,1\}^{N \times J}$ such that, $\gamma_{ij}^p = 1$ (or $\gamma_{ij}^q = 1$), if the i-th point of $\boldsymbol{\mathcal{P}}$ (or $\boldsymbol{\mathcal{Q}}$) belongs to the j-th cluster (denoted as $\bar{\boldsymbol{p}}_j$ or $\bar{\boldsymbol{q}}_j$) and 0 otherwise. Using this partitioning, we can now compute the clustered attention. First, we calculate the cluster centroids $\boldsymbol{f}_{\bar{p}_j}$ and $\boldsymbol{f}_{\bar{q}_j}$ of the points in each of these J clusters in feature space as follows,

$$\boldsymbol{f}_{\bar{p}_j} = \sum_{i=1}^{N} \frac{\gamma_{ij}^p \boldsymbol{f}_{p_i}}{\sum_k^N \gamma_{kj}^p}, \quad \boldsymbol{f}_{\bar{q}_j} = \sum_{i=1}^{M} \frac{\gamma_{ij}^q \boldsymbol{f}_{q_j}}{\sum_k^M \gamma_{kj}^q}. \tag{2}$$

We then use a multi-attention layer with four parallel attention heads [39] to update $\boldsymbol{\mathcal{F}}_{\boldsymbol{p}}$ in parallel via

$$\boldsymbol{f}_{\boldsymbol{p}_i} \leftarrow \boldsymbol{f}_{\boldsymbol{p}_i} + \mathrm{MLP}\left(\sum_{j=1}^{J} \alpha_{ij}^{\boldsymbol{p}} W_V^s \boldsymbol{f}_{\bar{\boldsymbol{p}}_j}\right), \tag{3}$$

where $\alpha_{ij}^{\boldsymbol{p}}$ is the element of matrix $\boldsymbol{\alpha}^{\boldsymbol{p}} = \mathrm{Softmax}(\boldsymbol{S})$ with $\boldsymbol{S} = (W_Q^s \boldsymbol{f}_{\boldsymbol{p}_i})^\top W_K^s \boldsymbol{f}_{\bar{\boldsymbol{p}}_j}$. Here, $W_Q^s \in \mathbb{R}^{N \times d}, W_K^s \in \mathbb{R}^{J \times d}$ and $W_V^s \in \mathbb{R}^{J \times d}$ are the query, key and value matrices. The self-attention features for $\boldsymbol{\mathcal{Q}}$ are updated in the same way. MLP$(\cdot)$ denotes a three-layer fully connected network with instance normalization [37] and ReLU [43] activations after the first two layers.

3.1.2 Overlap Region Detection

Cluster-based cross-attention. Cross-attention is a typical module for point cloud registration tasks, which performs feature exchange between two input point clouds. Given the self-attention feature matrices $\boldsymbol{\mathcal{F}}_{\boldsymbol{p}}$ and $\boldsymbol{\mathcal{F}}_{\boldsymbol{q}}$ for $\boldsymbol{\mathcal{P}}$ and $\boldsymbol{\mathcal{Q}}$ respectively, we first update the cluster centroids $\boldsymbol{f}_{\bar{p}_j}$ and $\boldsymbol{f}_{\bar{q}_j}$ based on Eq. (2). We denote the transformed features as $\boldsymbol{\mathcal{F}}_p^t$ and $\boldsymbol{\mathcal{F}}_p^t$ attained by cross-attention via

$$\boldsymbol{f}_{\boldsymbol{p}_i}^t \leftarrow \boldsymbol{f}_{\boldsymbol{p}_i} + \mathrm{MLP}\left(\sum \beta_{ij}^{\boldsymbol{p}} W_V^c \boldsymbol{f}_{\bar{\boldsymbol{q}}_j}\right), \tag{4}$$

where $\beta_{ij}^{\boldsymbol{p}}$ is the element of matrix $\boldsymbol{\beta}^{\boldsymbol{p}} = \mathrm{SoftMax}(\boldsymbol{C})$ with $\boldsymbol{C} = (W_Q^c \boldsymbol{f}_{\boldsymbol{p}_i})^\top W_K^c \boldsymbol{f}_{\bar{\boldsymbol{q}}_j}$. Here, $W_Q^c \in \mathbb{R}^{N \times d}, W_K^c \in \mathbb{R}^{J \times d}$ and $W_V^c \in \mathbb{R}^{J \times d}$ are the query, key and value matrices. MLP$(\cdot)$ denotes a three-layer fully connected network with instance normalization [37] and ReLU [43] activations after the first two layers. The same cross-attention block is also applied in reverse direction, so that information flows in both directions, $\boldsymbol{\mathcal{P}} \to \boldsymbol{\mathcal{Q}}$ and $\boldsymbol{\mathcal{Q}} \to \boldsymbol{\mathcal{P}}$.

Overlap score prediction. To deal with non-overlapping points, an overlap score prediction block is proposed. After obtaining the conditioned features $\boldsymbol{\mathcal{F}}_p^t$ and $\boldsymbol{\mathcal{F}}_q^t$, the per-point overlap score $\boldsymbol{o}_{p_i} \in [0,1]$ can be computed by

$$w_{ij} = \sigma\left({\boldsymbol{f}_{p_i}^t}^\top \boldsymbol{f}_{q_j}^t / \tau\right), \boldsymbol{o}_{p_i} = g_\beta\left(\text{cat}\left[\boldsymbol{f}_{q_i}^t, \boldsymbol{w}_i^\top g_\alpha\left(\boldsymbol{\mathcal{F}}_q^t\right)\right]\right),$$

where σ is a Softmax function, and $\tau > 0$ is a learned parameter that controls the soft assignment. When $\tau \to 0$, w_{ij} converges to a hard nearest-neighbor assignment. $g_\alpha(\cdot) : \mathbb{R}^d \to [0,1]$ and $g_\beta(\cdot) : \mathbb{R}^{d+1} \to [0,1]$ are linear layers followed by an instance normalization layer and a sigmoid activation with different parameters α and β, respectively.

3.2. Overlap-guided GMM for Registration

A Gaussian mixture model (GMM) establishes a multimodal generative probability distribution over 3D space as a weighted sum of L Gaussian densities [49], with the form

$$p(\boldsymbol{x}) = \sum_{j=1}^{L} \pi_j \mathcal{N}\left(\boldsymbol{x} | \boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j\right), \boldsymbol{x} \in \mathbb{R}^3. \tag{5}$$

Each density $\mathcal{N}\left(\boldsymbol{x} | \boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j\right)$ is referred to as a component of the GMM and parameterized by a mean $\boldsymbol{\mu}_j$ and covariance $\boldsymbol{\Sigma}_j$. The components are combined through a set of normalized mixing coefficients $\{\pi_1, \pi_2, \cdots, \pi_L\}$ that represent the prior probability of selecting the j-th component.

Learning posterior. We first apply a classification head ϕ_θ that takes as input $\boldsymbol{\mathcal{F}}_p$ and $\boldsymbol{\mathcal{F}}_q$ and outputs joint log probabilities, and a softmax operator that acts on log probabilities to generate a probability matrix $\boldsymbol{S}^p$ and $\boldsymbol{S}^q$, respectively. The GMM parameters $\boldsymbol{\Theta}_p$ for point cloud $\boldsymbol{\mathcal{P}}$ consists of L triples $(\pi_j^p, \boldsymbol{\mu}_j^p, \boldsymbol{\Sigma}_j^p)$, where π_j^p is a scalar mixture weight, $\boldsymbol{\mu}_j^p$ is a 3×1 mean vector and $\boldsymbol{\Sigma}_j^p$ is a 3×3 covariance matrix of the j-th component. For partial overlapping registration tasks, given the outputs $\boldsymbol{S}^p$ of ϕ_θ together with the point coordinates $\boldsymbol{\mathcal{P}}$, and overlap scores $\boldsymbol{o}_p$, the GMM parameters can be written as

$$n_p = \sum_{i=1}^{N} \boldsymbol{o}_{p_i}, \pi_j^p = \sum_{i=1}^{N} \frac{\boldsymbol{o}_{p_i} \boldsymbol{s}_{ij}^p}{\epsilon + n_p}, \boldsymbol{\mu}_j^p = \sum_{i=1}^{N} \frac{\boldsymbol{o}_{p_i} \boldsymbol{s}_{ij}^p \boldsymbol{p}_i}{\epsilon + n_p \pi_j^p},$$

$$\boldsymbol{\Sigma}_j^p = \frac{1}{\epsilon + n_p \pi_j^p} \sum_{i=1}^{N} \boldsymbol{o}_{p_i} \boldsymbol{s}_{ij}^p \left(\boldsymbol{p}_i - \boldsymbol{\mu}_j^p\right)\left(\boldsymbol{p}_i - \boldsymbol{\mu}_j^p\right)^\top,$$

where $\epsilon = 1e-4$ is used to avoid zero in the denominator. By the same way, we can get the $\pi_j^q, \boldsymbol{\mu}_j^q$ and $\boldsymbol{\Sigma}_j^q$ for target point cloud, when giving $\boldsymbol{S}^q$, $\boldsymbol{o}_q$ and $\boldsymbol{\mathcal{Q}}$. The GMMs of point set $\boldsymbol{\mathcal{P}}$ and $\boldsymbol{\mathcal{Q}}$ are then given as

$$\begin{aligned} \mathbf{G}_{\mathcal{P}}(\boldsymbol{x}) &= \sum_{i=1}^{L} \pi_j^p \mathcal{N}\left(\boldsymbol{x} | \boldsymbol{\mu}_j^p, \boldsymbol{\Sigma}_j^p\right), \\ \mathbf{G}_{\mathcal{Q}}(\boldsymbol{x}) &= \sum_{i=1}^{L} \pi_j^q \mathcal{N}\left(\boldsymbol{x} | \boldsymbol{\mu}_j^q, \boldsymbol{\Sigma}_j^q\right). \end{aligned} \tag{6}$$

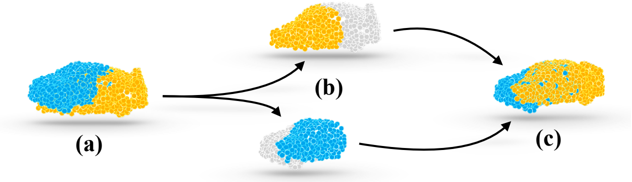

Figure 2. Given (a) input partial point clouds, OGMM detects (b) the overlap regions that are then used for the estimation of (c) the transformation that aligns the two point clouds. The non-overlap regions in (b) are shown in grey. Our approach focuses on the geometric information in the overlap regions to perform the point cloud registration.

We replace $\boldsymbol{p}_i$ and $\boldsymbol{q}_i$ with $\boldsymbol{f}_{p_i}$ and $\boldsymbol{f}_{q_i}$ to calculate the feature centroids $\{\boldsymbol{\nu}_j^p\}_{j=1}^L$ and $\{\boldsymbol{\nu}_j^q\}_{j=1}^L$ of $\boldsymbol{\mathcal{P}}$ and $\boldsymbol{\mathcal{Q}}$, respectively.

Estimating the transformation. Given the estimated GMMs parameters estimated through Eq. (6) as well as feature centroids $\{\boldsymbol{\nu}_j^p\}_{j=1}^L$ and $\{\boldsymbol{\nu}_j^q\}_{j=1}^L$, we first calculate the cluster-level matching matrix Γ by solving the following optimal transport (OT) problem [31] as

$$\begin{aligned} &\min_{\Gamma} \sum_{i=1}^{L} \sum_{j=1}^{L} \Gamma_{ij} \|\boldsymbol{\nu}_i^p - \boldsymbol{\nu}_j^q\|_2^2, \\ &\text{s.t.}, \Gamma \mathbf{1}_M = \boldsymbol{\pi}^p, \Gamma^\top \mathbf{1}_N = \boldsymbol{\pi}^q, \Gamma_{ij} \in [0,1], \end{aligned} \tag{7}$$

where $\boldsymbol{\pi}^t = (\pi_1^t, \pi_1^t, \cdots, \pi_L^t), t \in \{p, q\}$. The minimization of Eq. (7) can be solved in polynomial time as a linear program, and we address this issue by adopting an efficient version of the Sinkhorn-Knopp algorithm [7]. After obtaining the Γ, we then calculate the transformation by

$$\min_{T} L \sum_{i=1}^{L} \sum_{j=1}^{L} \Gamma_{ij} \|T(\boldsymbol{\mu}_i^p) - \boldsymbol{\mu}_j^q\|. \tag{8}$$

Finally, we can solve T in a closed-form using a weighted version of the SVD solution [49].

Fig. 2 shows an example of registration where the overlap regions that our approach can automatically determine are colored with the respective point cloud colors (non-overlap regions are in grey). This example depicts a case where the overlap between the two point clouds is 50%.

3.3. Training

We jointly optimize three tasks, finding overlap regions, clustering point clouds, and estimating rigid transformations.

Overlap score loss. The goal of the overlap score loss is to detect the overlap region between $\boldsymbol{\mathcal{P}}$ and $\boldsymbol{\mathcal{Q}}$. Given ground-truth $\bar{\boldsymbol{T}}$, the ground-truth $\bar{\boldsymbol{o}}_{p_i}$ of $\boldsymbol{p}_i$ is defined

$$\bar{\boldsymbol{o}}_{p_i} = \begin{cases} 1, & \left(\min_{q_j \in \boldsymbol{\mathcal{Q}}} \|\bar{\boldsymbol{T}}(\boldsymbol{p}_i) - \boldsymbol{q}_j\|\right) < \eta \\ 0, & \text{otherwise} \end{cases}, \tag{9}$$

$\bar{\boldsymbol{o}}_{q_j}$ is calculated in the same way. The overlap score loss is defined as $\mathcal{L}_O = \frac{1}{2}\left(\mathcal{L}_{\boldsymbol{\mathcal{P}}} + \mathcal{L}_{\boldsymbol{\mathcal{Q}}}\right)$, where

$$\mathcal{L}_{\boldsymbol{\mathcal{P}}} = -\frac{1}{|\boldsymbol{\mathcal{P}}|}\sum_{i}\left(\bar{\boldsymbol{o}}_{\boldsymbol{p}_i}\log \boldsymbol{o}_{\boldsymbol{p}_i} + \left(1-\bar{\boldsymbol{o}}_{\boldsymbol{p}_i}\right)\log\left(1-\boldsymbol{o}_{\boldsymbol{p}_i}\right)\right).$$

By the same operation, we can attain $\mathcal{L}_{\boldsymbol{\mathcal{Q}}}$.

Global registration loss. We first exploit the registration error-based loss function to train our model. To handle the partially-overlapping problem well, we adopt a robust error metric that does not incur high computational overhead. Specifically, we formulate the registration loss as

$$\mathcal{L}_g = \sum_{\hat{p}\in\hat{\boldsymbol{\mathcal{P}}}} \psi_\nu\left(\mathcal{D}\left(\hat{\boldsymbol{p}}, m\left(\bar{\boldsymbol{T}}(\boldsymbol{p}), \boldsymbol{\mathcal{Q}}\right)\right)\right).$$

Here, $m(\boldsymbol{x}, \boldsymbol{\mathcal{Q}})$ maps point $\boldsymbol{x}$ to its nearest point in $\boldsymbol{\mathcal{Q}}$. $\hat{\boldsymbol{\mathcal{P}}}$ denotes the transformed $\boldsymbol{\mathcal{P}}$ using the estimated transformation. $\mathcal{D}(\cdot,\cdot)$ defines as the Euclidean distance of two vectors. ψ_ν is the Welsch's function [15] as $\psi_\nu(x) = 1 - \exp\left(-\frac{x^2}{2\nu^2}\right)$. $\nu > 0$ is a user-specified parameter.

Clustering-based loss. To ensure that our network learns a consistent GMM representation across feature and geometric space rather than fit a GMM in a single feature space. We introduce the following loss.

$$\begin{aligned}\mathcal{L}_c = &-\sum_{ij}\gamma_{ij}^{p}\log\frac{\exp\left(-\mathcal{D}\left(\boldsymbol{p}_i, \boldsymbol{\mu}_{\boldsymbol{j}}^{\boldsymbol{p}}\right)\right)}{\sum_l \exp\left(-\mathcal{D}\left(\boldsymbol{p}_i, \boldsymbol{\mu}_{\boldsymbol{l}}^{\boldsymbol{p}}\right)\right)} \\ &-\sum_{ij}\gamma_{ij}^{q}\log\frac{\exp\left(-\mathcal{D}\left(\boldsymbol{q}_i, \boldsymbol{\mu}_{\boldsymbol{j}}^{\boldsymbol{q}}\right)\right)}{\sum_l \exp\left(-\mathcal{D}\left(\boldsymbol{q}_i, \boldsymbol{\mu}_{\boldsymbol{l}}^{\boldsymbol{q}}\right)\right)}.\end{aligned} \tag{10}$$

4. Experiments

We perform extensive experiments and ablation studies on both synthetic and real-world point cloud datasets, including the ModelNet40 [41], 7Scenes [35] and ICL-NUIM [14]. Unless otherwise specified, $J = 72, K = 5, \eta = 0.1$. Our implementation is built on the PyTorch library. We used AdamW as optimizer with a base learning rate of 0.001. The batch size is 32, and the learning rate was reduced by a factor of 0.7 every 20 epochs. We trained our model for 200 epochs. All of our models were trained on two Tesla V100-PCI-E-32G GPUs. We set GMM components $L = 48$. To alleviate the local minima clustering solution, we initialize the centroids of Wasserstein K-Means using the farthest point sampling strategy and the equal partition constraint.

4.1. Comparisons

We compare our approach with methods from state-of-the-art by running their code on the selected datasets. Because OGMM belongs to the category of learning-based probabilistic registration, we compare its performance with probabilistic approaches: CPD [29], GMMReg [20], SVR [4], DeepGMR [49], and FilterReg [12]. For these traditional methods, we use the implementations provided by probreg [22]. We improve the DeepGMR by replacing its encoder of DeepGMR with our encoder for a fair comparison. We also include point-level correspondence methods: traditional methods (i.e., ICP [2], FGR [53]), using implementations from Open3D [54], and the learning-based state-of-the-art methods RGM [10], OMNET [44], RIENet [34], and REGTR [48] (using authors' implementation).

4.2. Evaluation metrics

We evaluate the registration quality by using the *Mean Absolute Error (MAE)* between the estimated rotation angle θ_{est} and ground truth θ_{gt}, and the estimated translation t_{est} and ground truth t_{gt} [10, 39]. Rotation metrics are in *degrees*, while translation metrics are in *cm*. We also use *Clip Chamfer Distance (CCD)*, which measures how close the two point clouds are aligned with each other [10]. To avoid the influence of outliers in partial-to-partial registration, each pair of points whose distance is larger than 0.1 is discarded. This is implemented by setting the threshold $d = 0.1$.

4.3. Datasets

ModelNet40 [41] contains 12,311 meshed CAD models from 40 categories. Following [17], we split the dataset and create two setups, namely *same-category* and *cross-category*. The same-category setup contains 20 categories for training and testing. The cross-category setups consist of 20 categories that are disjoint between training and testing for the generalization performance evaluation. Following RGM [10], each category consists of official train/test splits. To select models for evaluation, we take 80%, and 20% of the official train split as the training set and validation set, respectively, and the official test split for testing. In realistic scenes, the points in $\boldsymbol{\mathcal{P}}$ have no exact correspondences in $\boldsymbol{\mathcal{Q}}$ [44]. We thus uniformly sample 1,024 points from each CAD model two times with different random seeds to generate $\boldsymbol{\mathcal{P}}$ and $\boldsymbol{\mathcal{Q}}$, breaking exact correspondences between the input point clouds, which is different from previous works. Following previous works [18, 10], we randomly draw a rigid transformation along each axis to generate the target points; the rotation along each axis is sampled in $[0, 45°]$ and translation is in $[-0.5, 0.5]$. $\nu = 0.1$.

7Scenes [35] is a 3D dataset captured in seven indoor scenes by using a Kinect RGB-D camera and consists of seven scenes: Chess, Fires, Heads, Office, Pumpkin, Red kitchen, and Stairs. 7Scenes is split into two parts, one is for training with 296 scans and one is for testing with 57 scans. 7Scenes is widely used to assess registration performance with real-world data. The point clouds are also uniformly sampled two-time from the original point clouds and a rigid transformation for one of the two samples to simulate the pose difference. We randomly draw a rigid transformation along each axis to generate the target points; the rotation along each axis is sampled within the interval $[0, 45°]$, while the translation is sampled within the interval $[-0.5, 0.5]$. $\nu = 0.5$.

ICL-NUIM [14] is derived from RGB-D scans in the Aug-

Table 1. Partial-to-Partial Registration results on ModelNet40.

Method	Same-category setup			Cross-category setup		
	MAE(R)	MAE(t)	CCD	MAE(R)	MAE(t)	CCD
ICP [2]	10.333	0.1034	0.1066	11.499	0.1084	0.1142
FGR [53]	22.103	0.1273	0.1108	21.928	0.1281	0.1175
RGM [10]	0.8211	0.0094	0.0729	1.3249	0.0164	0.0832
OMNET [44]	2.5944	0.0249	0.0936	3.6001	0.0355	0.1063
RIENet [34]	4.9586	0.0152	0.0735	5.5074	0.0425	0.1072
REGTR [48]	0.7836	**0.0066**	0.0676	0.9105	0.0071	0.0645
CPD [29]	11.033	0.1139	0.1110	12.681	0.1153	0.1154
GMMReg [20]	13.677	0.1344	0.1180	14.899	0.1440	0.1268
SVR [4]	11.857	0.1162	0.1170	13.120	0.1225	0.1237
FilterReg [12]	20.363	0.1558	0.1182	20.531	0.1646	0.1302
DeepGMR [49]	6.8043	0.0683	0.1182	7.3139	0.0718	0.1207
OGMM (ours)	**0.5892**	0.0079	**0.0493**	**0.6309**	**0.0055**	**0.0548**

Table 2. Registration results on ModelNet40 with jittering noise or density variation.

Method	Jittering			Density variation		
	MAE(R)	MAE(t)	CCD	MAE(R)	MAE(t)	CCD
ICP [2]	10.609	0.1058	0.1087	5.6654	0.0638	0.0908
FGR [53]	24.534	0.1413	0.1168	16.867	0.0295	0.0890
RGM [10]	1.3536	0.0261	0.0447	1.5948	0.0103	0.0672
OMNET [44]	2.6996	0.0259	0.0977	3.1372	0.0313	0.1010
RIENet [34]	5.8212	0.0635	0.1003	5.5928	0.0725	0.1059
REGTR [48]	1.0984	0.0080	0.0732	3.6409	0.0237	0.0946
CPD [29]	11.049	0.1141	0.1120	8.5349	0.0793	0.0941
GMMReg [20]	13.763	0.1343	0.1187	8.1247	0.0853	0.1024
SVR [4]	12.174	0.1192	0.1178	5.0848	0.0594	0.0949
FilterReg [12]	19.921	0.1548	0.1184	20.098	0.1016	0.1109
DeepGMR [49]	8.8242	0.0699	0.1210	7.1273	0.0689	0.1206
OGMM (ours)	**0.9111**	**0.0071**	**0.0645**	**1.3523**	**0.0100**	**0.0534**

mented ICL-NUIM dataset [5]. Following [49], we split the scenes in ICL-NUIM into 1,278 and 200 scans for training and testing, respectively. $\nu = 0.5$.

4.4. Evaluation on ModelNet40

Same-category setup. We follow the protocol in [48] to generate partial-to-partial point cloud pairs, which are closer to real-world applications. We first generate a half-space with a random direction for each point cloud and shift it to retain approximately 70% of the points, i.e., 717 points. Tab. 1 reports the results. OGMM significantly outperforms both traditional and deep learning-based methods. Unlike DeepGMR, OGMM can better detect overlapping regions thanks to the newly introduced overlap scores. Fig. 3 shows various successful and unsuccessful registration results. We can observe that cases with low overlap (e.g., guitar) and with repetitive structures (e.g., net) can be handled by OGMM. The most frequent unsuccessful cases involve point clouds of symmetric objects that can have multiple correct registration solutions along the symmetry axis that do not match with the ground-truth one. This is an intrinsic problem of symmetric cases [44]. The second most frequent unsuccessful cases involve point cloud pairs with repetitive local geometric structures. These make GMM clustering underperform because features of similar structures in different locations have a small distance in the feature space.

Cross-category setup. To evaluate the generalization ability of our method, we train our model and all the other deep learning-based methods on 20 categories (i.e., airplane, bathtub, bed, bench, bookshelf, bottle, bowl, car, chair, cone, cup, curtain, desk, door, dresser, flower pot, glass box, guitar, keyboard, and lamp), and test them on other different 20 categories (i.e., laptop, mantel, monitor, nightstand, person, piano, plant, radio, range hood, sink, sofa, stairs, stool, table, tent, toilet, TV stand, vase, wardrobe, and Xbox). The data pre-processing is the same as that of the first experiment. Tab. 1 shows that the performances of all learning-based methods are worse than those trained in the same categories. OGMM still outperforms the other baselines in this setup.

Noisy shapes. To evaluate the robustness of OGMM, Gaussian noise sampled from $\mathcal{N}(0, 0.01)$ and clipped to $[-0.05, 0.05]$ is independently added to each point coordinate. We train and test our OGMM on the noisy data of ModelNet40. From the results listed in the left part of Tab. 2 (Jittering), we can observe that OGMM can effectively handle noise. This occurs thanks to our cluster-based network that can extract more robust features.

Density variation. We aim to evaluate the robustness of methods in handling variations of point cloud densities. We randomly discard points of one point cloud and repeat the remaining points to keep the same number of points to generate a pair of point clouds with density differences. Tab. 2 (Density variation) reports the results when the point density is reduced to 50%. OGMM can outperform the other methods when source and target point clouds have different densities. This occurs mainly because the source and target point clouds are modeled as GMMs, therefore the alignment between the two GMMs becomes less sensitive to density, as opposed to establishing the explicit point correspondence.

4.5. Evaluation on 7Scenes and ICL-NUIM

We conduct experiments on two indoor scenes: real-world 7Scenes [35] and ICL-NUIM. The point clouds are uniformly sampled from the original point clouds to generate source and target counterparts. Following [17], we resample 2,048 points from each model two times with different random seeds to generate $\mathcal{P}$ and $\mathcal{Q}$, then generate a half-space with a random direction for each point cloud and shift it to retain approximately 70% of the points (1,433) to generate the partial data. As shown in Tab. 3, OGMM achieves the lowest errors on all the metrics on both datasets. OGMM outperforms DeepGMR and GMMReg thanks to the clustered attention network to detect overlapping regions and produce more distinctive features.

4.6. Ablation Studies

Components of OGMM. We analyze the effectiveness of OGMM components in the case of partial-to-partial registration (same-category setup). We assess the three key novel components of OGMM: Spherical Positional Encoding (SPE), Cluster-based Self-Attention (CSA), and Overlap

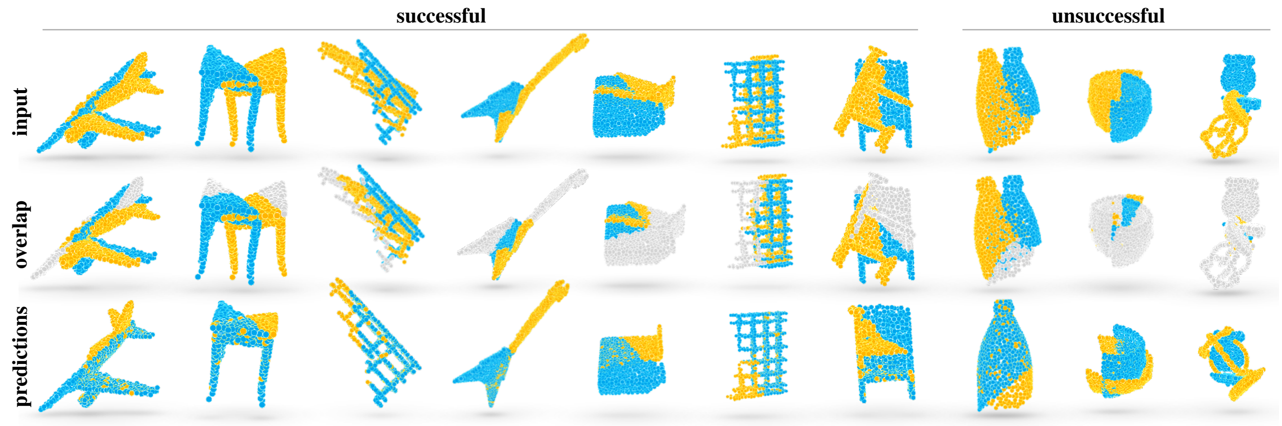

Figure 3. Successful and unsuccessful registration results on ModelNet40 using OGMM. The non-overlap regions are shown in grey.

Table 3. The registration results on 7Scenes and ICL-NUIM.

Method	7Scenes			ICL-NUIM		
	MAE(R)	MAE(t)	CCD	MAE(R)	MAE(t)	CCD
ICP [2]	18.266	0.2346	0.0793	10.539	0.3301	0.1410
FGR [53]	1.1736	0.0198	0.0270	0.8792	0.0332	0.0835
RGM [10]	3.0334	0.0445	0.0425	1.3279	0.0416	0.0840
OMNET [44]	9.8499	0.1416	0.1879	17.177	0.4587	0.1947
REGTR [48]	4.6143	0.0827	0.1733	3.2503	0.1044	0.0902
CPD [29]	4.9897	0.1056	0.0900	9.8322	0.3828	0.1581
GMMReg [20]	11.081	0.1213	0.1374	6.5411	0.1700	0.1530
SVR [4]	10.729	0.1152	0.1388	6.3229	0.1946	0.1528
FilterReg [12]	18.113	0.2521	0.0636	28.317	0.7930	0.1693
DeepGMR [49]	8.8478	0.1534	0.0541	6.4600	0.1899	0.1269
OGMM (ours)	**0.5764**	**0.0088**	**0.0214**	**0.6279**	**0.0305**	**0.0732**

Table 4. Ablation study on ModelNet40.

SPE	CSA	OSP	MAE(R)	MAE(t)	CCD
	✓	✓	0.9534	0.0096	0.0589
✓		✓	1.9060	0.0169	0.0889
✓	✓		6.7087	0.0729	0.1155
✓	✓	✓	**0.5892**	**0.0079**	**0.0493**

Table 5. Loss function analysis on ModelNet40.

GR	CL	OS	MAE(R)	MAE(t)	CCD
✓			3.7284	0.0272	0.0958
	✓		7.0198	0.0553	0.1133
		✓	4.4373	0.0392	0.1006
✓	✓		3.2335	0.0262	0.0939
	✓	✓	2.6470	0.0250	0.0751
✓		✓	0.7828	0.0085	0.0515
✓	✓	✓	**0.5892**	**0.0079**	**0.0493**

Score Prediction (OSP). Tab. 4 shows that OGMM underperforms when the overlap scores are not used. CSA and SPE modules help achieving a higher registration accuracy, as they allow for more distinctive features to be produced.

Loss functions. We train our model with different combinations of the Global Registration loss (GR), the Clustering-based loss (GS), and the Overlap Score loss (OS). Experiments are conducted on ModelNet40 (same-category setup). Tab. 5 shows that the combination of GR and OS losses provides the major contribution.

For the ablation study results of overlapping ratios and clusters, please refer to the supplementary material.

Table 6. Comparisons of the average inference time.

Method	CPD	GMMReg	RGM	DeepGMR	OGMM (f)	OGMM (c)
Time(s)	4.347	2.536	0.057	**0.002**	0.047	0.006

Inference time. We evaluate the efficiency of OGMM and compare it to other approaches on ModelNet40 (same-category setup). We average the inference time of OGMM using a single Tesla V100 GPU (32G) and two Intel(R) 6226 CPUs. f and c represent the full and cluster-based attention. Tab. 6 reports the results. Compared to RGM, OGMM utilizes similar network architecture but different matching strategies and attention modules. OGMM (c) outperforms RGM and reduces about 9× the computation time. Compared with full attention, our cluster-based attention can speed up 8× times, which verifies the effectiveness of our clustered strategy in reducing the computing complexity. Our method is inferior to DeepGRM because the overlap detection module is introduced to handle partial overlap cases.

5. Conclusions

We presented a learning-based probabilistic registration approach for partially overlapping point clouds, named OGMM. We use a clustered attention-based network to detect overlap regions between two point clouds and formulate the registration problem as the minimization of the discrepancy between Gaussian mixtures under the guidance of the overlap information. Experiments show that OGMM outperforms prior traditional and deep learning-based registration approaches across a variety of data setups. OGMM is robust to noise and can effectively generalize to new object categories and on real-world data. OGMM features a novel way to integrate 3D neural networks inside a probabilistic registration paradigm. Future research directions include the development of an unsupervised method to detect the overlap regions to reduce the dependence on labeled data.

References

[1] Yasuhiro Aoki, Hunter Goforth, Rangaprasad Arun Srivatsan, and Simon Lucey. Pointnetlk: Robust & efficient point cloud registration using pointnet. In *CVPR*, pages 7163–7172, 2019.

[2] Paul J Besl and Neil D McKay. A method for registration of 3-d shapes. *TPAMI*, 14(2):239–256, 1992.

[3] Dorit Borrmann, Andreas Nuechter, and Thomas Wiemann. Large-scale 3d point cloud processing for mixed and augmented reality. In *ISMAR-Adjunct*, 2018.

[4] Dylan Campbell and Lars Petersson. An adaptive data representation for robust point-set registration and merging. In *ICCV*, pages 4292–4300, 2015.

[5] Sungjoon Choi, Qian-Yi Zhou, and Vladlen Koltun. Robust reconstruction of indoor scenes. In *CVPR*, pages 5556–5565, 2015.

[6] Christopher Choy and et al. Deep global registration. In *CVPR*, pages 2514–2523, 2020.

[7] Marco Cuturi. Sinkhorn distances: Lightspeed computation of optimal transport. *NeurIPS*, 26:2292–2300, 2013.

[8] Georgios Dimitrios Evangelidis and Radu Horaud. Joint alignment of multiple point sets with batch and incremental expectation-maximization. *TPAMI*, 40(6):1397–1410, 2017.

[9] Andrew W Fitzgibbon. Robust registration of 2d and 3d point sets. *Image Vis Comput*, 21(13-14):1145–1153, 2003.

[10] Kexue Fu and et al. Robust point cloud registration framework based on deep graph matching. In *CVPR*, pages 8893–8902, 2021.

[11] Takumi Fukunaga and Hiroyuki Kasai. Wasserstein k-means with sparse simplex projection. In *ICPR*, pages 1627–1634. IEEE, 2021.

[12] Wei Gao and Russ Tedrake. Filterreg: Robust and efficient probabilistic point-set registration using gaussian filter and twist parameterization. In *CVPR*, pages 11095–11104, 2019.

[13] Meng-Hao Guo, Jun-Xiong Cai, Zheng-Ning Liu, Tai-Jiang Mu, Ralph R Martin, and Shi-Min Hu. Pct: Point cloud transformer. *Comput Vis Media (Beijing)*, 7(2):187–199, 2021.

[14] Ankur Handa, Thomas Whelan, John McDonald, and Andrew J Davison. A benchmark for rgb-d visual odometry, 3d reconstruction and slam. In *ICRA*, pages 1524–1531. IEEE, 2014.

[15] Paul W Holland and Roy E Welsch. Robust regression using iteratively reweighted least-squares. *Commun. Stat. Theory Methods*, 6(9):813–827, 1977.

[16] Shengyu Huang and et al. Predator: Registration of 3d point clouds with low overlap. In *CVPR*, pages 4267–4276, 2021.

[17] Xiaoshui Huang, Sheng Li, Yifan Zuo, Yuming Fang, Jian Zhang, and Xiaowei Zhao. Unsupervised point cloud registration by learning unified gaussian mixture models. *RA-L*, 2022.

[18] Xiaoshui Huang, Guofeng Mei, and Jian Zhang. Feature-metric registration: A fast semi-supervised approach for robust point cloud registration without correspondences. In *CVPR*, pages 11366–11374, 2020.

[19] Xiaoshui Huang, Jian Zhang, Qiang Wu, Lixin Fan, and Chun Yuan. A coarse-to-fine algorithm for matching and registration in 3d cross-source point clouds. *TCSVT*, 28(10):2965–2977, 2017.

[20] Bing Jian and Baba C Vemuri. Robust point set registration using gaussian mixture models. *TPAMI*, 33(8):1633–1645, 2010.

[21] Pranav Kadam, Min Zhang, Shan Liu, and C-C Jay Kuo. R-pointhop: A green, accurate, and unsupervised point cloud registration method. *TIP*, 31:2710–2725, 2022.

[22] Kenta-Tanaka. Probreg, 2019-9-29.

[23] Felix Järemo Lawin, Martin Danelljan, Fahad Shahbaz Khan, Per-Erik Forssén, and Michael Felsberg. Density adaptive point set registration. In *CVPR*, pages 3829–3837, 2018.

[24] Jiayuan Li, Qingwu Hu, and Mingyao Ai. Point cloud registration based on one-point ransac and scale-annealing biweight estimation. *TGRS*, 59(11):9716–9729, 2021.

[25] Jiahao Li, Changhao Zhang, and et al. Iterative distance-aware similarity matrix convolution with mutual-supervised point elimination for efficient point cloud registration. In *ECCV*, 2019.

[26] Weifeng Li, Zhuoran Jiang, Kaiyue Chu, Jianhua Jin, Yun Ge, and Jing Cai. A noninvasive method to reduce radiotherapy positioning error caused by respiration for patients with abdominal or pelvic cancers. *TCRT*, 18, 2019.

[27] Zhengfei Ma et al. Point reg net: Invariant features for point cloud registration using in image-guided radiation therapy. *JCC*, 6(11):116, 2018.

[28] Guofeng Mei. Point cloud registration with self-supervised feature learning and beam search. In *DICTA*, pages 01–08. IEEE, 2021.

[29] Andriy Myronenko and Xubo Song. Point set registration: Coherent point drift. *TPAMI*, 32(12):2262–2275, 2010.

[30] Balázs Nagy and Csaba Benedek. Real-time point cloud alignment for vehicle localization in a high resolution 3d map. In *ECCVW*, 2018.

[31] Gabriel Peyré, Marco Cuturi, et al. Computational optimal transport: With applications to data science. *Foundations and Trends® in Machine Learning*, 11(5-6):355–607, 2019.

[32] F. Poiesi and D. Boscaini. Learning general and distinctive 3D local deep descriptors for point cloud registration. *TPAMI*, 2022.

[33] Zheng Qin, Hao Yu, Changjian Wang, Yulan Guo, Yuxing Peng, and Kai Xu. Geometric transformer for fast and robust point cloud registration. In *CVPR*, pages 11143–11152, 2022.

[34] Yaqi Shen, Le Hui, Haobo Jiang, Jin Xie, and Jian Yang. Reliable inlier evaluation for unsupervised point cloud registration. *AAAI*, 2022.

[35] Jamie Shotton, Ben Glocker, Christopher Zach, Shahram Izadi, Antonio Criminisi, and Andrew Fitzgibbon. Scene coordinate regression forests for camera relocalization in rgb-d images. In *CVPR*, pages 2930–2937, 2013.

[36] Zhiliang Sun, Rongguo Zhang, Jing Hu, and Xiaojun Liu. Probability re-weighted 3d point cloud registration for missing correspondences. *Multimed. Tools. Appl.*, 81(8):11107–11126, 2022.

[37] Dmitry Ulyanov, Andrea Vedaldi, and Victor Lempitsky. Instance normalization: The missing ingredient for fast stylization. *arXiv preprint arXiv:1607.08022*, 2016.

[38] Y. Wang and A. Del Bue. Where to Explore Next? ExHistCNN for History-aware Autonomous 3D Exploration. In *ECCV*, 2020.

[39] Yue Wang and Justin M Solomon. Deep closest point: Learning representations for point cloud registration. In *ICCV*, pages 3523–3532, 2019.

[40] Yue Wang, Yongbin Sun, Ziwei Liu, Sanjay E Sarma, Michael M Bronstein, and Justin M Solomon. Dynamic graph cnn for learning on point clouds. *ACM TOG*, 38(5):1–12, 2019.

[41] Zhirong Wu, Shuran Song, Aditya Khosla, Fisher Yu, Linguang Zhang, Xiaoou Tang, and Jianxiong Xiao. 3d shapenets: A deep representation for volumetric shapes. In *CVPR*, pages 1912–1920, 2015.

[42] Saining Xie, Sainan Liu, Zeyu Chen, and Zhuowen Tu. Attentional shapecontextnet for point cloud recognition. In *CVPR*, pages 4606–4615, 2018.

[43] Bing Xu, Naiyan Wang, Tianqi Chen, and Mu Li. Empirical evaluation of rectified activations in convolutional network. *arXiv preprint arXiv:1505.00853*, 2015.

[44] Hao Xu, Shuaicheng Liu, Guangfu Wang, Guanghui Liu, and Bing Zeng. Omnet: Learning overlapping mask for partial-to-partial point cloud registration. In *CVPR*, pages 3132–3141, 2021.

[45] Heng Yang, Jingnan Shi, and Luca Carlone. Teaser: Fast and certifiable point cloud registration. *T-RO*, 37(2):314–333, 2020.

[46] Jiaolong Yang, Hongdong Li, Dylan Campbell, and Yunde Jia. Go-icp: A globally optimal solution to 3d icp point-set registration. *TPAMI*, 38(11):2241–2254, 2015.

[47] Zi Jian Yew and Gim Hee Lee. Rpm-net: Robust point matching using learned features. In *CVPR*, pages 11824–11833, 2020.

[48] Zi Jian Yew and Gim Hee Lee. Regtr: End-to-end point cloud correspondences with transformers. In *CVPR*, pages 6677–6686, 2022.

[49] Wentao Yuan, Benjamin Eckart, Kihwan Kim, Varun Jampani, Dieter Fox, and Jan Kautz. Deepgmr: Learning latent gaussian mixture models for registration. In *ECCV*, pages 733–750. Springer, 2020.

[50] Cheng Zhang, Haocheng Wan, Xinyi Shen, and Zizhao Wu. Patchformer: An efficient point transformer with patch attention. In *CVPR*, pages 11799–11808, 2022.

[51] Zhiyuan Zhang, Yuchao Dai, and Jiadai Sun. Deep learning based point cloud registration: an overview. *VRIH*, 2(3):222–246, 2020.

[52] Hengshuang Zhao, Li Jiang, Jiaya Jia, Philip HS Torr, and Vladlen Koltun. Point transformer. In *CVPR*, pages 16259–16268, 2021.

[53] Qian-Yi Zhou, Jaesik Park, and Vladlen Koltun. Fast global registration. In *ECCV*, pages 766–782, 2016.

[54] Qian-Yi Zhou, Jaesik Park, and Vladlen Koltun. Open3d: A modern library for 3d data processing. *arXiv preprint arXiv:1801.09847*, 2018.

[55] Y. Zhou, Y. Wang, F. Poiesi, Q. Qin, and Y. Wan. Loop closure detection using local 3D deep descriptors. *R-AL*, 2022.