Deep Conditional Clustering for Point Cloud Registration

Anonymous ICCV submission

Paper ID ****

Abstract

This paper proposes a clustering-based point cloud registration approach under the constraint of overlap scores learning by a network. Specifically, our CluReg first introduces a geometric transformer-based network to extract pointwise features with their associated overlap scores. Then, the clustering is implemented in the coordinate space under the constraints of overlap scores to generate cluster centers with associated cluster probabilities, which can be translated into solving a weighted Wasserstein K-Means problem. After that, the probabilities are used to calculate feature centers in feature space. Finally, the transformation are estimated using both coordinate and feature centers.

1. Introduction

- 23/02 Intro
- 24/02 Related work
- 26/02 method
- 28/02 finish partial exps
- 05/03 finish all experiments and a draft version.

Point cloud registration aims to seek a relative spatial transformation that aligns two point clouds with each other, which is a crucial aspect in various applications, including but not limited to 3D printing [22], robotics, and autonomous driving [3]. The state-of-art registration pipelines commonly involve first acquiring local descriptors and detecting overlap regions. These descriptors in the overlap regions are then matched to identify a set of possible correspondences, which are subsequently used to estimate the transformation. If any step is unsuccessful, it will result in inaccurate estimation of transformation, leading to unsatisfactory registration performance. Especially learning-based methods have recently dominated recent registration advances, showing significant improvements in accuracy and efficiency compared to traditional methods. However, the presence of noise, repetitive patterns, and varying density levels challenges the registration accuracy.

all-pair similarity matrix, which may result in a large combinatorial search space and vulnerability to over-fitting.

OGMM [16] applies a cluster head (MLP) to assign each point in a point cloud with soft cluster labels, which corporate the learning overlap scores to calculate the cluster centers in both coordinate and feature space. The centers are then used to estimate the transformation based on optimal transport. To our best knowledge, OGMM is the first work incorporated with overlap scores to deal with partial overlap registration. However, it underperforms in registration tasks when the point cloud contains multiple objects, since using a network to learn all possible clusters is unreasonable in multi-objective scenes. Besides, points in different regions tend to group into the same clusters when low-texture regions or repetitive patterns dominate the field of view. This issue is especially prominent in indoor environments.

- Using a network to learn all possible clusters is unreasonable in the multi-objective scenes.
- It means all points of the source or target will be assigned corresponding points without distinguishing inliers and outliers
- the main problem is that they require the inputs to have distinctive geometric structures to promote sparse matched points. However, not all regions are distinctive, resulting in a limited number of matches or poor distributions.

Putting fewer weights on these non-overlapped points can potentially improve the clustering algorithm.

Contributions

- We provide a soft clustering-based point cloud registration.
- We provide a conditional clustering method, which can be solved by translating it into an optimal transport problem.

2. Related Work

We review correspondence-based registration, including point-level and distribution-level methods, since our work follows the line of correspondence-based methods. As unsupervised learning is a major component in our proposed learning framework, we also review work on this topic.

Point-Level Methods. Point-level approaches first extract point-wise features, then establish point-to-point correspondences through feature matching, followed by outlier rejection and robust estimation of the rigid transformation. Numerous works, such as FCGF [5] and RGM [9], focus on extracting discriminative features for geometric correspondences. For the correspondence prediction, DCP [19], RPMNet [23], and REGTR [24] perform feature matching by integrating the Sinkhorn algorithm or Transformer [?] into a network to generate soft correspondences from local features. IDAM [14] incorporates both geometric and distance features into the iterative matching process. To reject outliers, DGR [4] and 3DRegNet [18] use networks to estimate the inliers. Predator [11] and PRNet [20] focus on detecting points in the overlap region and utilizing their features to generate matches. Keypoint-free methods [15, 28, 25] first downsample the point clouds into super-points and then match them by examining whether their neighborhoods (patch) overlap. Though achieving remarkable performance, most of these methods rely

Cluster-Level Methods. Cluster-level methods model the point clouds as clusters, often via the use of GMMs, and perform alignment either by employing a correlation-based or an EM-based optimization framework. The correlation-based methods [12, 26] first build GMM probability distributions for both the source and target point clouds. Then, the transformation is estimated by minimizing a metric or divergence between the distributions. However, these methods lead to nonlinear optimization problems with nonconvex constraints [13]. Unlike correlation-based methods, the EM-based approaches, such as JRMPC [7], CPD [17], and FilterReg [10], represent the geometry of one point cloud using a GMM distribution over 3D Euclidean space. The transformation is then calculated by fitting another point cloud to the GMM distribution under the maximum likelihood estimation (MLE) framework. These methods are robust to noise and density variation [26]. Most of them utilize robust discrepancies to reduce the influence of outliers by greedily aligning the largest possible fraction of points while being tolerant of a small number of outliers. However, if outliers dominate, the greedy behavior of these methods easily emphasizes outliers, leading to degraded registration results [7]. Considering these factors, we formulate registration in a novel partial cluster matching framework, where

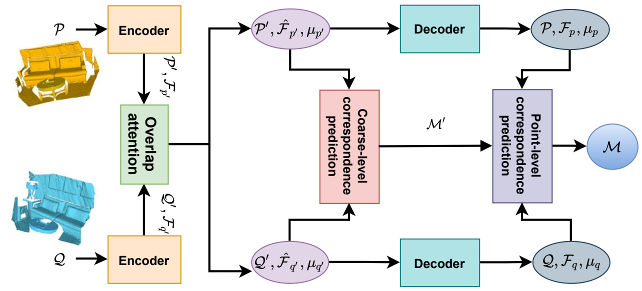

Figure 1. Overview of the proposed method.Please ignore this figure. NEED TO BE REPLACED

we only seek to partially match the distributions.

3. The Proposed Methods

3.1. Problem formulation

Point cloud registration refers to recover a transformation $\boldsymbol{T} \in SE(3)$ that aligns the source set $\boldsymbol{\mathcal{P}} = \{\boldsymbol{p}_i \in \mathbb{R}^3 | i = 1, 2, ..., N\}$ to the target set $\boldsymbol{\mathcal{Q}} = \{\boldsymbol{q}_j \in \mathbb{R}^3 | j = 1, 2, ..., M\}$. N and M represent the number of points in $\boldsymbol{\mathcal{P}}$ and $\boldsymbol{\mathcal{Q}}$, respectively. $\boldsymbol{T}$ can be calculated using correspondences between $\boldsymbol{\mathcal{P}}$ and $\boldsymbol{\mathcal{Q}}$. Our work focuses on correspondence estimation. The pipeline of our CluReg is illustrated in Fig. 1, which is a shared weighted two-stream encoder-decoder network. Given a point cloud pair $\boldsymbol{\mathcal{P}}$ and $\boldsymbol{\mathcal{Q}}$, the encoder aggregates the raw points into super-points $\bar{\boldsymbol{\mathcal{P}}}$ and $\boldsymbol{\mathcal{Q}}'$, while jointly learning the associated features $\boldsymbol{\mathcal{F}}_{\bar{p}}$ and $\boldsymbol{\mathcal{F}}_{q'}$. The attention block updates the features as $\hat{\boldsymbol{\mathcal{F}}}_{\bar{p}}$ and $\hat{\boldsymbol{\mathcal{F}}}_{q'}$, and projects them to super-point overlap scores $\boldsymbol{\mu}_{\bar{p}}, \boldsymbol{\mu}_{q'}$. After that, the decoder transforms the features and super-point overlap scores to per-point features $\boldsymbol{\mathcal{F}}_p, \boldsymbol{\mathcal{F}}_q$ and overlap scores $\boldsymbol{\mu}_p, \boldsymbol{\mu}_q$.

3.2. Feature extraction

Encoder. A KPConv-FPN, which consists of a series of ResNet-like blocks and stridden convolutions, simultaneously down-samples the raw points clouds $\boldsymbol{\mathcal{P}}$ and $\boldsymbol{\mathcal{Q}}$ into super-points $\bar{\boldsymbol{\mathcal{P}}} = \{\bar{\boldsymbol{p}}_i \in \mathbb{R}^3 | i = 1, 2, ..., \bar{N}\}$ and $\boldsymbol{\mathcal{Q}}' = \{\boldsymbol{q}'_j \in \mathbb{R}^3 | j = 1, 2, ..., \bar{M}\}$ and extracts associated point-wise features $\boldsymbol{\mathcal{F}}_{\bar{p}} = \{\boldsymbol{f}_{\bar{p}} \in \mathbb{R}^b | i = 1, 2, ..., \bar{N}\}$ and $\boldsymbol{\mathcal{F}}_{q'} = \{\boldsymbol{f}_{q'_j} \in \mathbb{R}^b | j = 1, 2, ..., \bar{M}\}$, respectively.

Geometry-aware overlap attention module. The geometry aware overlap attention module, which estimates the probability (overlap score) of whether a point is in the overlapping area, consists of positional encoding, self-attention, and cross-attention. To better leverage the 3D geometric structures of point clouds, we introduce positional encoding that assigns intrinsic geometric properties to per-point features, thus enhancing distinctions among point features in indistinctive regions. Self-attention models the long-range dependencies. And cross attention exploits the intra-relationship within the source and target point clouds, which models the potential overlap regions.

Specifically, given a superpoint $\bar{\boldsymbol{p}}_i$ of $\bar{\boldsymbol{\mathcal{P}}}$, we first select k ($k = 5$ in our experiments) nearest neighbors Ω_i of $\bar{\boldsymbol{p}}_i$. Its associated covariance matrix Σ_i in the local region is calculated as

$$\begin{aligned} \Sigma_i &= \sum_{\boldsymbol{x}_j \in \Omega_i} \omega_{x_j} (\boldsymbol{x}_j - \boldsymbol{p}_i)(\boldsymbol{x}_j - \boldsymbol{p}_i)^\top, \\ \omega_{x_j} &= \frac{\phi - \|\boldsymbol{x}_j - \boldsymbol{p}_i\|_2}{\sum_{\boldsymbol{x}_j \in \Omega_i} (\phi - \|\boldsymbol{x}_j - \boldsymbol{p}_i\|_2)}, \end{aligned} \tag{1}$$

where $\phi = \max_{\boldsymbol{x}_j \in \Omega_i} \|\boldsymbol{x}_j - \boldsymbol{p}_i\|_2$. The global centroid of $\bar{\boldsymbol{\mathcal{P}}}$ is $\bar{\boldsymbol{p}}_c = \frac{1}{N} \sum_{i=1}^{\bar{N}} \bar{\boldsymbol{p}}_i$. The positional encoding $\boldsymbol{f}_{\bar{p}}^e$ of $\bar{\boldsymbol{p}}_i$ is defined as follow:

$$\boldsymbol{f}_{\bar{p}_i}^e = \varphi \left(\mathrm{cat} \left[\frac{\|\bar{\boldsymbol{p}}_i - \bar{\boldsymbol{p}}_c\|_2}{\max_j \|\bar{\boldsymbol{p}}_j - \bar{\boldsymbol{p}}_c\|_2}, \mathrm{vec}\,(\Sigma_i) \right] \right), \tag{2}$$

where φ is an MLP consisting of a linear layer and a ReLU. Let $\boldsymbol{\mathcal{F}}_{\bar{p}}^l$ be the intermediate representation for $\bar{\boldsymbol{\mathcal{P}}}$ at layer l and let $\boldsymbol{\mathcal{F}}_{\bar{p}}^0 = \{\boldsymbol{f}_{\bar{p}} + \boldsymbol{f}_{\bar{p}}^e\}_{i=1}^{\bar{N}}$. The multi-attention with four parallel attention head updates $\boldsymbol{\mathcal{F}}_{\bar{p}}^l$ via

$$\begin{aligned} \boldsymbol{S}_{\bar{p}} &= \boldsymbol{W}_1^l \boldsymbol{\mathcal{F}}_{\bar{p}}^l + \boldsymbol{b}_1^l, \boldsymbol{K}_{x'} = \boldsymbol{W}_2^l \boldsymbol{\mathcal{F}}_{x'}^l + \boldsymbol{b}_2^l, \\ \boldsymbol{V}_{x'} &= \boldsymbol{W}_3^l \boldsymbol{\mathcal{F}}_{x'} + \boldsymbol{b}_3^l, \boldsymbol{A} = \sigma \left(\boldsymbol{S}_{\bar{p}}^\top \boldsymbol{K}_{x'} / \sqrt{b} \right), \\ \boldsymbol{\mathcal{F}}_{\bar{p}}^{l+1} &= \boldsymbol{\mathcal{F}}_{\bar{p}} + g^l \left(\left[\boldsymbol{\mathcal{F}}_{\bar{p}}^l \| \boldsymbol{A} \boldsymbol{V}_{x'} \right] \right). \end{aligned} \tag{3}$$

Here, σ is a softmax function. If $x' = \bar{p}$ represents self-attention, and if $x' = q'$ indicates cross-attention. $[\cdot\|\cdot]$ denotes concatenation, and $g^l(\cdot)$ is a three-layer fully connected network consisting of a linear layer, instance normalization, and a LeakyReLU activation. The same attention module is also simultaneously performed for all points in point cloud $\boldsymbol{\mathcal{Q}}'$. A fixed number of layers $L = 2$ with different parameters are chained and alternatively aggregate along the self- and cross- attention. As such, starting from $l = 0, x' = \bar{p}$ if l is even and $x' = q'$ if l is odd. The final outputs of attention module are $\hat{\boldsymbol{\mathcal{F}}}_{\bar{p}} = \boldsymbol{\mathcal{F}}_p^3$ for $\bar{\boldsymbol{\mathcal{P}}}$ and $\hat{\boldsymbol{\mathcal{F}}}_{q'} = \boldsymbol{\mathcal{F}}_{q'}^3$ for $\boldsymbol{\mathcal{Q}}'$. By doing this, the latent features $\hat{\boldsymbol{\mathcal{F}}}_{\bar{p}}$ has the knowledge of $\hat{\boldsymbol{\mathcal{F}}}_{q'}$ and vice versa. After obtaining the conditioned features $\hat{\boldsymbol{\mathcal{F}}}_{\bar{p}}$ and $\hat{\boldsymbol{\mathcal{F}}}_{q'}$, the overlap score $\mu_{\bar{p}} \in [0, 1]$ of super-point $\bar{\boldsymbol{p}}_i$, which is proposed to detect the overlap regions, can be computed by

$$w_{ij} = \sigma \left(\hat{\boldsymbol{f}}_{\bar{p}}^\top \hat{\boldsymbol{f}}_{q'_j} \right), \mu_{\bar{p}} = g_\beta \left(\mathrm{cat} \left[\hat{\boldsymbol{f}}_{\bar{p}}, \boldsymbol{w}_i^\top g_\alpha \left(\hat{\boldsymbol{\mathcal{F}}}_{q'} \right) \right] \right),$$

where $g_\alpha(\cdot) : \mathbb{R}^b \to [0, 1]$ and $g_\beta(\cdot) : \mathbb{R}^{b+1} \to [0, 1]$ are linear layers followed by an instance normalization layer and a sigmoid activation with different parameters α and β, respectively. As a shorthand, we denote $\boldsymbol{\mu}_{\bar{p}} = \{\mu_{\bar{p}}\}_{i=1}^{\bar{N}}$. The same operator is applied to calculate $\boldsymbol{\mu}_{q'}$.

Decoder. The decoder, which consists of several KPConv layers, starts from the super-points $\bar{\boldsymbol{\mathcal{P}}}$ and the concatenations of $\hat{\boldsymbol{\mathcal{F}}}_{\bar{p}}$ and $\boldsymbol{\mu}_{\bar{p}}$, and outputs raw point cloud $\boldsymbol{\mathcal{P}}$ with associated features $\boldsymbol{\mathcal{F}}_p \in \mathbb{R}^{N \times 32}$ and overlap scores $\boldsymbol{\mu}_p \in [0, 1]^N$. The raw point cloud $\boldsymbol{\mathcal{Q}}$ and its associated features $\boldsymbol{\mathcal{F}}_q \in \mathbb{R}^{M \times 32}$ and overlap scores $\boldsymbol{\mu}_q \in [0, 1]^M$ is obtained in the same way.

3.3. Conditional clustering registration

The goal of the conditional clustering algorithm is to partition given a set of data $\boldsymbol{X} = \{\boldsymbol{x}_1, \cdots, \boldsymbol{x}_K\}$ with associated weight $\boldsymbol{\mu} = \{\boldsymbol{\mu_1}, \cdots, \boldsymbol{\mu_K}\}$ into L separated groups, i.e., L clusters, as $\boldsymbol{C} = \{c_1, \cdots, c_L\}$ with associated clustering probability matrix $\boldsymbol{\gamma} = \{\gamma_{ij}\}$ such that the following cost function is minimized:

$$\begin{aligned} &\min_{\boldsymbol{C}, \boldsymbol{\gamma}} \sum_{k=1}^{K} \sum_{l=1}^{L} \gamma_{kl} \|\boldsymbol{x}_k - c_l\|_2^2, \\ &\text{s.t.,} \boldsymbol{\gamma}^\top \boldsymbol{1}_N = \frac{1}{J} \boldsymbol{1}_J, \boldsymbol{\gamma} \boldsymbol{1}_J = \mathrm{softmax}\,(\boldsymbol{\mu}). \end{aligned} \tag{4}$$

The minimization of Eq. (4) can be solved in polynomial time as a linear program. However, the linear program involves millions of data points and thousands of classes and traditional algorithms hardly scale to large problems [6]. We address this issue by adopting an efficient version of the Sinkhorn-Knopp algorithm [6].

The operator performing on point clouds $\boldsymbol{\mathcal{P}}$ and $\boldsymbol{\mathcal{Q}}$ to get cluster centers $\boldsymbol{\mathcal{P}}^c = \{\boldsymbol{p}_i^c\}_{j=1}^L$ and $\boldsymbol{\mathcal{Q}}^c = \{\boldsymbol{q}_j^c\}_{j=1}^L$, respectively. Then, we calculate the cluster centroids $\boldsymbol{f}_{p_j}^c$ and $\boldsymbol{f}_{q_j}^c$ of the points in each of these J clusters in feature space as follows,

$$\boldsymbol{f}_{p_j}^c = \sum_{i=1}^{N} \frac{\gamma_{ij}^p \boldsymbol{f}_{p_i}}{\sum_k^N \gamma_{kj}^p}, \quad \boldsymbol{f}_{q_j}^c = \sum_{i=1}^{M} \frac{\gamma_{ij}^q \boldsymbol{f}_{q_j}}{\sum_k^M \gamma_{kj}^q}. \tag{5}$$

Extracting point correspondences is to match two smaller corresponded scale point clouds ($\boldsymbol{\mathcal{P}}^c$, $\boldsymbol{\mathcal{Q}}^c$) by solving an optimization problem

$$\min_{\boldsymbol{\Gamma}} \langle \boldsymbol{D}, \boldsymbol{\Gamma} \rangle, \tag{6}$$

where $\boldsymbol{\Gamma} = [\boldsymbol{\Gamma}]_{ij}$ represents an assignment matrix and $\boldsymbol{D} = [\boldsymbol{D}]_{ij}$ with $\boldsymbol{D}_{ij} = \| \frac{\boldsymbol{\mathcal{F}}_{p_i}^c}{\|\boldsymbol{\mathcal{F}}_{p_i}^c\|_2} - \frac{\boldsymbol{\mathcal{F}}_{q_j}^c}{\|\boldsymbol{\mathcal{F}}_{q_j}^c\|_2} \|_2$. The picked point correspondences from ($\boldsymbol{\mathcal{P}}_p^c$, $\boldsymbol{\mathcal{Q}}_q^c$) are defined as

$$\mathcal{M} = \{(\boldsymbol{p}_{\hat{i}}^c \in \boldsymbol{\mathcal{P}}_p^c, \boldsymbol{q}_{\hat{j}} \in \boldsymbol{\mathcal{Q}}_q^c) | \hat{j} = \arg\max_k \boldsymbol{\Gamma}_{\hat{i},k}\}. \tag{7}$$

Following [2, 25], a variant of RANSAC [8] that is specialized to 3D registration takes as an input $\mathcal{M}$ to estimate the transformation.

3.4. Loss Function and Training

Our model is an end-to-end learning framework, using the ground truth correspondences as supervision. The loss function $\mathcal{L} = \mathcal{L}_C + \mathcal{L}_F + \mathcal{L}_{CO} + \mathcal{L}_{FO}$ is composed of an coarse-level loss $\mathcal{L}_C$ for superpoint matching, a point matching loss $\mathcal{L}_F$ for point matching, a binary classification loss $\mathcal{L}_{CO}$ for coarse-level overlap scores, and a classification loss $\mathcal{L}_{FO}$ for fine-level overlap scores.

3.4.1 Coarse-Level Loss

Superpoint Matching Loss. Existing methods [25, 9] usually formulate superpoint matching as a multilabel classification problem and adopt a cross-entropy loss with optimal transport. Doing this requires unfolding the Sinkhorn layer to compute gradients in the training stage. To address this issue, we adopt a circle loss [**?**] to optimize the superpoint-wise feature descriptors. As there is not direct supervision for superpoint matching, we leverage the overlap ratio r_i^j of points in $G_{\bar{p}_i}$ that have correspondences in $G_{\bar{q}_j}$ to depict the matching probability between superpoints $\bar{p}_i$ and $\bar{q}_j$. r_i^j is defined as:

$$r_i^j = \frac{1}{|G_{\bar{p}_i}|} |\{\boldsymbol{p} \in G_{\bar{p}_i} | \min_{\boldsymbol{q} \in G_{\bar{q}_j}} \|\hat{\boldsymbol{T}}(\boldsymbol{p}) - \boldsymbol{q}\|_2 < r_p\}|.$$

where $\hat{\boldsymbol{T}}$ is the ground-truth transformation and r_p is a set threshold. For circle loss, a pair of superpoints are positive if their corresponded patches share at least 10% overlap, and negative if they do not overlap. All other pairs are omitted. We select the superpoints in $\bar{\boldsymbol{\mathcal{P}}}$ which have at least one positive superpoint in $\bar{\boldsymbol{\mathcal{Q}}}$ to form a set of anchor superpoints, $\tilde{\boldsymbol{\mathcal{P}}}$. For each anchor $\tilde{\boldsymbol{p}}_i \in \tilde{\boldsymbol{\mathcal{P}}}$, we denote the set of its positive superpoints in $\bar{\boldsymbol{\mathcal{Q}}}$ as $\mathcal{N}_p^{\tilde{\boldsymbol{p}}_i}$, and the set of its negative patches as $\mathcal{N}_n^{\tilde{\boldsymbol{p}}_i}$. The superpoint matching loss (circle loss) $\mathcal{L}_C^{\bar{\boldsymbol{\mathcal{P}}}}$ on $\bar{\boldsymbol{\mathcal{P}}}$ is then defined as:

$$\begin{aligned} \mathcal{L}_C^{\bar{\boldsymbol{\mathcal{P}}}} &= \frac{1}{|\bar{\boldsymbol{\mathcal{P}}}|} \sum_{\tilde{\boldsymbol{p}}_i \in \bar{\boldsymbol{\mathcal{P}}}} \log\left[1 + \zeta_i\right], \\ \zeta_i &= \sum_{\tilde{\boldsymbol{q}}_k \in \mathcal{N}_p^{\tilde{\boldsymbol{p}}_i}} e^{r_i^k \beta_p^{ik}(d_i^k - \Delta p)} \cdot \sum_{\tilde{\boldsymbol{q}}_l \in \mathcal{N}_n^{\tilde{\boldsymbol{p}}_i}} e^{\beta_n^{il}(\Delta n - d_i^l)}, \end{aligned} \tag{8}$$

where $d_i^k = \mathcal{D}_f(\boldsymbol{f}_{\tilde{p}_i}, \boldsymbol{f}_{\tilde{q}_k})$ is the distance in the feature space. The weights $\beta_p^{ik} = \gamma d_i^k$ and $\beta_n^{il} = \gamma(2.0 - d_i^l)$ are determined individually for each positive and negative example, using the empirical margins $\Delta p = 0.1$ and $\Delta n = 1.4$ with a learned scale factor $\gamma \geq 1$. The circle loss reweights the loss values on $\mathcal{N}_{p^i}$ based on the overlap ratio so that the patch pairs with higher overlap are given more importance. The same goes for the loss $\mathcal{L}_C^{\bar{\boldsymbol{\mathcal{Q}}}}$ on $\bar{\boldsymbol{\mathcal{Q}}}$. The overall superpoint matching loss is

$$\mathcal{L}_C = \frac{1}{2}(\mathcal{L}_C^{\bar{\boldsymbol{\mathcal{P}}}} + \mathcal{L}_C^{\bar{\boldsymbol{\mathcal{Q}}}}). \tag{9}$$

Coarse-Level Overlap Loss. We use the ratio of points in $G_{\bar{p}_i}$ that are visible in $\mathcal{Q}$ to depict the ground-truth overlap scores $\bar{\boldsymbol{\mu}}_{\bar{p}_i}$ of the superpoint $\bar{p}_i$. It is calculated by

$$\bar{\boldsymbol{\mu}}_{\bar{p}_i} = \frac{1}{|G_{\bar{p}_i}|} |\{\boldsymbol{p} \in G_{\bar{p}_i} | \min_{\boldsymbol{q} \in \mathcal{Q}} \|\hat{\boldsymbol{T}}(\boldsymbol{p}) - \boldsymbol{q}\|_2 < r_o\}|, \tag{10}$$

with overlap threshold. If $\bar{\boldsymbol{\mu}}_{\bar{p}_i}$ is close to 1, $\bar{p}_i$ tends to locate in the overlap regions. $\bar{\boldsymbol{\mu}}_{\bar{q}_j}$ is calculated in the same way. The predicted overlap scores for $\bar{\boldsymbol{\mathcal{P}}}$ are thus supervised using the binary cross entropy loss, i.e.,

$$\mathcal{L}_{\bar{\boldsymbol{\mathcal{P}}}} = -\frac{1}{\bar{N}} \sum_i \bar{\boldsymbol{\mu}}_{\bar{p}_i} \log \boldsymbol{\mu}_{\bar{p}_i} + (1 - \bar{\boldsymbol{\mu}}_{\bar{p}_i}) \log(1 - \boldsymbol{\mu}_{\bar{p}_i}). \tag{11}$$

The loss $\mathcal{L}_{\bar{\boldsymbol{\mathcal{Q}}}}$ for $\bar{\boldsymbol{\mathcal{Q}}}$ is calculated in the same way. The loss for coarse-level overlap scores is

$$\mathcal{L}_{CO} = \frac{1}{2}(\mathcal{L}_{\bar{\boldsymbol{\mathcal{P}}}} + \mathcal{L}_{\bar{\boldsymbol{\mathcal{Q}}}}).$$

3.4.2 Fine-Level Loss

Point Matching Loss. We apply circle loss again to supervise the point matching. Consider a pair of matched superpoints $\bar{\boldsymbol{p}}_i$ and $\bar{\boldsymbol{q}}_j$ with associated patches $G_{\bar{p}_i}$ and $G_{\bar{q}_j}$, we first extract a set of anchor points $\tilde{G}_{\bar{p}_i} \subseteq G_{\bar{p}_i}$ satisfying that each $\boldsymbol{g}_{\bar{p}_i}^k \in \tilde{G}_{\bar{p}_i}$ has at least one (possibly multiple) correspondence in $G_{\bar{q}_j}$, i.e.,

$$\tilde{G}_{\bar{p}_i} = \{\boldsymbol{g}_{\bar{p}_i}^k \in \tilde{G}_{\bar{p}_i} | \min_{\boldsymbol{g}_{\bar{q}_j}^l \in G_{\bar{q}_j}} \|\hat{\boldsymbol{T}}(\boldsymbol{g}_{\bar{p}_i}^k) - \boldsymbol{g}_{\bar{q}_j}^l\|_2 < r_p\}.$$

For each anchor $\boldsymbol{g}_{\bar{p}_i}^k \in \tilde{G}_{\bar{p}_i}$, we denote the set of its positive points in $G_{\bar{q}_j}$ as $\mathcal{N}_p^{\boldsymbol{g}_{\bar{p}_i}^k}$. All points of $\mathcal{Q}$ outside a (larger) radios r_n form the set of its negative patches as $\mathcal{N}_n^{\boldsymbol{g}_{\bar{p}_i}^k}$. The fine-level matching loss $\mathcal{L}_F^{\mathcal{P}}$ on $\mathcal{P}$ is calculated as:

$$\begin{aligned} \mathcal{L}_F^{\mathcal{P}} &= \frac{1}{|\bar{\boldsymbol{\mathcal{P}}}|} \sum_{\tilde{\boldsymbol{p}}_i \in \bar{\boldsymbol{\mathcal{P}}}} \frac{1}{|\tilde{G}_{\bar{p}_i}|} \sum_{\boldsymbol{g}_{\bar{p}_i}^s \in \tilde{G}_{\bar{p}_i}} \log\left[1 + \xi_s\right], \\ \xi_s &= \sum_{\boldsymbol{g}_{\bar{q}_j}^k \in \mathcal{N}_p^{\boldsymbol{g}_{\bar{p}_i}^s}} e^{r_s^k \beta_p^{sk}(d_s^k - \Delta p)} \cdot \sum_{\boldsymbol{g}_{\bar{q}_j}^l \in \mathcal{N}_n^{\boldsymbol{g}_{\bar{p}_i}^s}} e^{\beta_n^{sl}(\Delta n - d_s^l)}, \end{aligned} \tag{12}$$

where $d_s^k = \mathcal{D}_f(\boldsymbol{f}_{\boldsymbol{g}_{\bar{p}_i}^s}, \boldsymbol{f}_{\boldsymbol{g}_{\bar{q}_j}^s})$ is the distance in the feature space. The weights $\beta_p^{sk} = \omega d_s^k$ and $\beta_n^{sl} = \omega(2.0 - d_s^l)$ are determined individually for each positive and negative example with a learned scale factor $\omega \geq 1$. $\Delta p = 0.1$ and $\Delta n = 1.4$. The same goes for the loss $\mathcal{L}_F^{\boldsymbol{\mathcal{Q}}}$ on $\boldsymbol{\mathcal{Q}}$. The overall superpoint matching loss writes as

$$\mathcal{L}_F = \frac{1}{2}(\mathcal{L}_F^{\mathcal{P}} + \mathcal{L}_F^{\mathcal{Q}}). \tag{13}$$

Fine-Level Overlap Loss. The overlap score loss is $\mathcal{L}_{FO} = -\frac{1}{2}\left(\frac{1}{|\bar{\mathcal{P}}|}\sum_{\bar{p}_i}\mathcal{L}_{\bar{p}_i} + \frac{1}{|\bar{\mathcal{Q}}|}\sum_{\bar{q}_j}\mathcal{L}_{\bar{q}_j}\right)$ with

$$\mathcal{L}_{\bar{p}_i} = \frac{1}{|\tilde{G}_{\bar{p}_i}|}\sum_{\boldsymbol{g}^k_{\bar{p}_i}}\left(\bar{\boldsymbol{\mu}}_{\boldsymbol{g}^k_{\bar{p}_i}}\log\boldsymbol{\mu}_{\boldsymbol{g}^k_{\bar{p}_i}} + \left(1-\bar{\boldsymbol{\mu}}_{\boldsymbol{g}^k_{\bar{p}_i}}\right)\log\left(1-\boldsymbol{\mu}_{\boldsymbol{g}^k_{\bar{p}_i}}\right)\right).$$

The ground-truth label $\bar{\boldsymbol{\mu}}_{\boldsymbol{g}^k_{\bar{p}_i}}$ of the point $\boldsymbol{g}^k_{\bar{p}_i} \in \tilde{G}_{\bar{p}_i}$ is defined as

$$\bar{\boldsymbol{\mu}}_{\boldsymbol{g}^k_{\bar{p}_i}} = \begin{cases} 1, & \left(\min_{q_j\in\boldsymbol{\mathcal{Q}}}\|\hat{\boldsymbol{T}}(\boldsymbol{g}^k_{\bar{p}_i}) - \boldsymbol{q}_j\|\right) < r_o \\ 0, & \text{otherwise} \end{cases}, \quad (14)$$

where $\mathcal{L}_{\bar{q}_j}$ is calculated in the same way.

4. Experiments

We conduct extensive experiments to evaluate the performance of our method on the real datasets 3DMatch [27] and 3DLoMatch [11], as well as on the synthetic datasets ModelNet [21] and ModelLoNet [11].

4.1. Implementation Details

Our method is implemented in PyTorch and was trained on one Quadro GV100 GPU (32G) and two Intel(R) Xeon(R) Gold 6226 CPUs. We used the AdamW optimizer with an initial learning rate of $1e{-}4$ and a weight decay of $1e{-}4$. We adopted the same encoder and decoder architectures used in [**?**]. For the 3DMatch dataset, we trained for 200 epochs with a batch size of 1, halving the learning rate every 70 epochs. We trained on the ModelNet for 400 epochs with a batch size of 1, halving the learning rate every 100 epochs. On 3DMatch and 3DLoMatch, we set $J{=}128$ with truncated patch size $K{=}32$. On ModelNet and ModelLoNet, we set $J{=}32$ with truncated patch size $K{=}32$. The cluster head MLP consists of 3 fully connected layers. Each layer is composed of a linear layer followed by batch normalization. The hidden layer and the final linear layer output dimension are 512 and 256, respectively. Except for the final layer, each layer has a LeakyReLU activation.

4.2. Evaluation on 3DMatch and 3DLoMatch

Datasets and Metrics. 3DMatch [27] and 3DLoMatch [11] are two widely used indoor datasets with more than 30% and 10%$\sim$30% partially overlapping scene pairs, respectively. 3DMatch contains 62 scenes, from which we use 46 for training, 8 for validation, and 8 for testing. The test set contains 1,623 partially overlapping point cloud fragments and their corresponding transformation matrices. We used training data preprocessed by [11] and evaluated with both the 3DMatch and 3DLoMatch protocols. Each input point cloud contains an average of about 20,000 points. We performed training data augmentation by applying small rigid perturbations, jittering the point locations, and shuffling points. Following Predator [11], we evaluated the Relative Rotation Errors (RRE) and Relative Translation Errors (RTE) that measure the accuracy of successful registrations. We also assessed Registration Recall (RR), the fraction of point cloud pairs whose transformation error is smaller than a threshold (i.e., 0.2m).

Table 1. Results on both 3DMatch and 3DLoMatch datasets. The best results for each criterion are labeled in bold, and the best results of unsupervised methods are underlined.

Method	3DMatch RR↑	3DMatch RRE↓	3DMatch RTE↓	3DLoMatch RR↑	3DLoMatch RRE↓	3DLoMatch RTE↓
	Point-level Methods					
FCGF[5]	85.1%	1.949	0.066	40.1%	3.147	0.100
D3Feat[1]	81.6%	2.161	0.067	37.2%	3.361	0.103
OMNet [**?**]	35.9%	4.166	0.105	8.4%	7.299	0.151
DGR [4]	85.3%	2.103	0.067	48.7%	3.954	0.113
Predator1K [11]	89.0%	2.062	0.068	62.4%	3.159	0.096
CoFiNet[25]	89.7%	2.147	0.067	67.2%	3.271	0.090
GeoTrans [**?**]	92.0%	1.808	0.063	**74.0%**	2.934	0.089
REGTR [24]	**92.0%**	**1.567**	**0.049**	64.8%	**2.827**	**0.077**
	Cluster-level Methods					
CluReg (Ours)	<u>91.4%</u>	<u>1.642</u>	<u>0.064</u>	<u>64.3%</u>	<u>2.951</u>	<u>0.086</u>

Baselines. We chose supervised state-of-the-art (SOTA) methods: FCGF [5], D3Feat [1], SpinNet [**?**], Predator [11], REGTR [24], CoFiNet [25], and GeoTransformer[**?**], as well as unsupervised PPFFoldNet [**?**] and SGP [**?**] as our baselines.

Registration Results. The results of various methods are shown in Table 1, where the best performance is highlighted in bold while the best-unsupervised results are marked with an underline. For both 3DMatch and 3DLoMatch, our method outperforms all unsupervised methods and achieves the lowest average rotation (RRE) and translation (RTE) errors across scenes. Our method also achieves the highest average registration recall, which reflects the final performance on point cloud registration (91.4% on 3DMatch and 64.3% on 3DLoMatch). Specifically, CluReg largely exceeds the previous winner and our closest competitor, SGP, (85.5% RR on 3DMatch) by about 5.9% and (39.4% RR on 3DLoMatch) by 24.9%. Interestingly, our method also exceeds some supervised methods, *e.g.* FCGF, D3Feat, DGR, and Predator1K, showing its efficacy in both high- and low-overlap scenarios. Even compared with recent supervised SOTA methods, our method achieves competitive results.

4.3. Generalization on Cross-source Dataset

The generalization ability of learning-based registration algorithms is highly required when the point cloud is acquired from different sensors. To validate the generalizability of our model, we experiment on our own Cross Source Dataset (3DCSR) [**?**]. 3DCSR is a challenging dataset for registration due to a mixture of noise, outliers, density difference, partial overlap, and scale variation.

4.3.1 3DCSR

This dataset contains two folders: Kinect Lidar and Kinect SFM. Kinect lidar contains 19 scenes from both the Kinect and Lidar sensors, where each scene is cropped into different parts. Kinect SFM consists of 2 scenes from both Kinect and RGB sensors. The RGB images have already been constructed into a point cloud by using the software VSFM. We use the model trained on 3DMatch since the cross-source dataset is captured in an indoor environment. RR is the percentage of successful alignment whose rotation error and translation error are below set thresholds (i.e., RRE $< 15°$ and RTE $< 6m$).

Table 2. Registration results on Cross Source Datasets. Best performance is highlighted in bold.

Method	Estimator	RRE (°) ↓	RTE (cm) ↓	RR(%) ↑
FCGF [5]	RANSAC	7.47	**0.21**	49.6
D3Feat [1]	RANSAC	6.41	0.26	52.0
SpinNet [?]	RANSAC	6.56	0.24	53.5
Predator [11]	RANSAC	6.26	0.27	54.6
CoFiNet [25]	RANSAC	5.76	0.26	57.3
GeoTrans [?]	RANSAC	5.60	0.24	60.2
CluReg (Ours)	RANSAC	**5.49**	**0.21**	**63.4**

4.3.2 Registration Results

We use FCGF [5], D3Feat [1], SpinNet [?], Predator [11], CoFiNet [25], and GeoTransformer [?], as the baselines. Table 2 shows that our method obtains the highest accuracies in generalizing the registration ability to real-world cross-source dataset. Specifically, it outperforms the second-best, GeoTransformer, by more than 3.2% in terms of registration recall (63.4% vs 60.2%). However, the recall is not high enough, showing that registration challenges on 3DCSR remain.

4.4. Ablation Study

To fully understand CluReg, we conduct an ablation study on 3DMatch and 3DLoMatch to investigate the contribution of each part. First, we replace the overlap scores with a uniform distribution, i.e., treating the points in overlap and non-overlap regions equally, to evaluate the effectiveness of overlap scores. As shown in Table 3, on 3DMatch, the learned overlap scores improve the performance by nearly 2.0% (92.9% vs. 90.9%) RR, 0.7% (98.5% vs. 97.8%) FMR, and 7.8% (86.1% vs. 68.3%) IR, respectively. Structure matching can boost RR by 1.1% (92.9% vs. 91.8%), FMR by 0.5% (98.5% vs. 98.0%) and IR by 10.2% (86.1% vs. 75.9%), respectively. It also indicates that CluReg benefits from the overlap scores and structure matching. Table 3 also shows that the positional encoding can improve the performance in terms of RR, FMR and IR. On 3DLoMatch, the same results can be concluded.

Table 3. Ablation study of individual modules, tested with #Samples=1000. **self** and **cross** indicate self- and cross-attention.

		3DMatch			3DLoMatch		
self	cross	RR	FMR	IR	RR	FMR	IR
✓		**92.9**	**98.5**	**86.1**	**79.7**	**89.7**	**55.1**
✓		91.8	98.0	75.9	74.6	88.9	46.4
✓	✓	90.9	97.8	68.3	67.2		
coarse	fine	RR	FMR	IR	RR	FMR	IR
✓		**92.9**	**98.5**	**86.1**	**79.7**	**89.7**	**55.1**
✓		91.8	98.0	75.9	74.6	88.9	46.4
✓	✓	90.9	97.8	68.3	67.2		

References

[1] Xuyang Bai and et al. D3feat: Joint learning of dense detection and description of 3d local features. In *CVPR*, pages 6359–6367, 2020.

[2] Xuyang Bai and et al. Pointdsc: Robust point cloud registration using deep spatial consistency. In *CVPR*, pages 15859–15869, 2021.

[3] Nathan Brightman, Lei Fan, and Yang Zhao. Point cloud registration: a mini-review of current state, challenging issues and future directions. *AIMS Geosciences*, 9(1):68–85, 2023.

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

[5] Christopher Choy, Jaesik Park, and Vladlen Koltun. Fully convolutional geometric features. In *ICCV*, pages 8958–8966, 2019.

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

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

[8] Martin A Fischler and Robert C Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *COMMUN ACM*, 24(6):381–395, 1981.

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

[10] 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.

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

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

[13] 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.

[14] 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.

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

[16] Guofeng Mei, Fabio Poiesi, Cristiano Saltori, Jian Zhang, Elisa Ricci, and Nicu Sebe. Overlap-guided gaussian mixture models for point cloud registration. In *WACV*, pages 4511–4520, 2023.

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

[18] G Dias Pais, Srikumar Ramalingam, Venu Madhav Govindu, Jacinto C Nascimento, Rama Chellappa, and Pedro Miraldo. 3dregnet: A deep neural network for 3d point registration. In *CVPR*, pages 7193–7203, 2020.

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

[20] Yue Wang and Justin M Solomon. Prnet: Self-supervised learning for partial-to-partial registration. In *NeurIPS*, 2019.

[21] 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.

[22] Yuxi Xie, Boyuan Li, Chao Wang, Kun Zhou, CT Wu, and Shaofan Li. A bayesian regularization network approach to thermal distortion control in 3d printing. *Computational Mechanics*, pages 1–18, 2023.

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

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

[25] Hao Yu and et al. Cofinet: Reliable coarse-to-fine correspondences for robust pointcloud registration. *NeurIPS*, 34, 2021.

[26] 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.

[27] Andy Zeng, Shuran Song, Matthias Nießner, Matthew Fisher, Jianxiong Xiao, and Thomas Funkhouser. 3dmatch: Learning local geometric descriptors from rgb-d reconstructions. In *CVPR*, pages 1802–1811, 2017.

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