Fully-Geometric Cross-Attention for Point Cloud Registration

Weijie Wang
University of Trento
weijie.wang@unitn.it

Guofeng Mei*
Fondazione Bruno Kessler
gmei@fbk.eu

Jian Zhang
University of Technology Sydney
Jian.Zhang@uts.edu.au

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

Bruno Lepri
Fondazione Bruno Kessler
lepri@fbk.eu

Fabio Poiesi
Fondazione Bruno Kessler
poiesi@fbk.eu

Point cloud registration approaches often fail when the overlap between point clouds is low due to noisy point correspondences. This work introduces a novel cross-attention mechanism tailored for Transformer-based architectures that tackles this problem, by fusing information from coordinates and features at the super-point level between point clouds. This formulation has remained unexplored primarily because it must guarantee rotation and translation invariance since point clouds reside in different and independent reference frames. We integrate the Gromov–Wasserstein distance into the cross-attention formulation to jointly compute distances between points across different point clouds and account for their geometric structure. By doing so, points from two distinct point clouds can attend to each other under arbitrary rigid transformations. At the point level, we also devise a self-attention mechanism that aggregates the local geometric structure information into point features for fine matching. Our formulation boosts the number of inlier correspondences, thereby yielding more precise registration results compared to state-of-the-art approaches. We have conducted an extensive evaluation on 3DMatch, 3DLoMatch, KITTI, and 3DCSR datasets. Project page: https://github.com/twowwj/FLAT.

1. Introduction

Point cloud registration aims to determine a relatively rigid transformation that aligns two partially observed point clouds. This is important in various applications, including 3D printing [23, 35], robotics [17], and autonomous driving [5]. In recent registration advances, deep learning-based methods have outperformed hand-crafted ones in both efficiency and accuracy [7, 26].

Learning-based point cloud registration can be categorized into *correspondence-free* [2, 14, 21, 36] or *correspondence-based* [4, 6, 8, 13, 26, 32, 39]. The former aims to minimize the difference between the global features of two point clouds. However, global features may hinder these approaches in handling partially overlapping scenes [6, 44]. The latter aims to detect keypoints, compute local descriptors, find correspondences, and estimate the rigid transformation [13, 19]. Correspondences can be defined at the point level or at the distribution level [26, 31]. Point-level correspondences may be noisy in the case of point clouds with different densities of points and/or with geometrically repetitive and uninformative local patterns (e.g. flat surfaces) [22, 37]. Distribution-level correspondences are designed to align point clouds with varying densities without establishing point-level correspondences, however, they fall short in handling point clouds with low overlaps [16, 20]. There also exist methods that replace keypoint detection by downsampling input point clouds into super-points in order to make computation efficient [40]. Super-points are matched between point clouds to find correspondences, which are in turn propagated at point-level to build dense point correspondences. Furthermore, self-attention and cross-attention in Transformers [29] can be used to incorporate global information (context) into features [13, 27, 40], thus producing distinctive features to register point clouds more accurately. Self-attention is performed in the coordinate space to encode the transformation-invariant geometric structure from each point cloud [27]. Cross-attention is performed in the feature space to model the geometric consistency across point clouds, by allowing information from one point cloud to be attended by another [27].

We argue that previously proposed cross-attention formulations only employ point feature information, overlooking coordinate information. Although cross-attention in the feature space is effective in improving correspondence quality between super-points, point-level correspondences remain rather noisy (we will show this experimentally). Intuitively, if we only consider feature information, disre-

*denotes the corresponding author.

garding their location, there could be situations where similar objects in different locations have similar geometric structures, hence similar features. This can produce incorrect correspondences. By enriching the cross-attention with coordinate information, we can encourage the network to explicitly learn corresponding geometric structures across point clouds, thus promoting feature distinctiveness.

To this end, in this paper we present a **ful**ly-geometric cross-**at**tention formulation, FLAT for short, followed by overlap-constrained clustering to learn accurate correspondences for point cloud registration. Unlike GeoTransformer [27], which only considers geometric relationships within a single point cloud. Our method, FLAT, uses geometric cross-attention to incorporate both source and target relationships. This provides comprehensive scene information, overcoming the limitations of partial scene representation in overlapped point clouds. In addition, we develop cross-spatial invariant geometric features and use the Gromov-Wasserstein distance to measure discrepancies across different metric spaces, such as pose differences. Our cross-attention is significantly more challenging to achieve, as measures of distance and angle used in GeoTransformer are no longer applicable because they are not invariant to rigid transformations with respect to different reference frames. To fix this gap, we introduce two new metrics, one for measuring the pair-wise distance and the other for determining the triplet-wise angle that can be computed between two different point clouds. As these two metrics are invariant to rigid transformation, our geometry-enhanced attention can efficiently exchange geometric structural information between point clouds, leading to more reliable correspondences, even in scenarios with low overlaps. Our registration approach follows a coarse-to-fine correspondence prediction strategy, identifying approximate matches using super-points and refining them by expanding them to patches (i.e. sets of points defined in the neighborhood of a super-point). Moreover, unlike previous techniques, we employ a distance-weighted self-attention to inject the local location information to further improve the distinctive of the local features. We evaluate our method on four popular benchmarks: 3DMatch [42] and 3DLoMatch [13] (indoors), KITTI [12] (outdoors), and cross-source 3DCSR [15] (indoors). The results show that FLAT outperforms previous approaches [13, 27, 40]. In summary, our contributions are:

- We introduce a geometry-enhanced cross-attention mechanism alongside a distance-weighted self-attention approach, aiming to refine the learning of accurate correspondences for point cloud registration.
- Our proposed geometry-enhanced cross-attention effectively integrates a transformation invariant geometric structure, enabling the model to learn distinctive features and emphasize the overlapping regions between point clouds.
- Leveraging local self-attention, which is grounded in coordinate relative distances, we generate distinctive features that enhance fine matching capabilities.

2. Related Work

Correspondence-based Registration. Correspondence-based methods typically involve two main steps: feature extraction and correspondence estimation through feature matching. They perform outlier rejection and robust estimation of the rigid transformation. FCGF [7], RGM [11], and GeDi [26] are examples of methods used to extract discriminative features. For correspondence prediction, RPMNet [38] performs feature matching by integrating the Sinkhorn algorithm into a network that generates soft feature correspondences. IDAM [18] incorporates both geometric and distance features in the iterative matching process. To reject outliers, DGR [6] and 3DRegNet [24] use networks for inlier prediction. Correspondence-based techniques can be further classified into two groups based on the strategy they employ to extract correspondences [27]. The methods in the first group aim to identify repeatable keypoints [3, 13] and develop discriminative descriptors for those keypoints [1, 30, 31]. The effectiveness of keypoint detection methods may be limited in scenarios with uneven point density or similar local structures. Repetitive local structures are typically present in indoor settings, where featureless flat surfaces can occupy a significant portion of the visual field. The methods in the second group aim to retrieve correspondences without detecting keypoints by examining all possible matches [27, 40]. They first downsample the point clouds into super-points and then match them by examining whether their neighborhoods (patches) overlap [19, 40, 43]. The accuracy of dense point correspondences relies on the accuracy of super-point matches [27]. When dealing with low-overlapping point clouds, the super-point matching mechanism merely exchanges information from the feature spaces, which contain only partial structural information. This limitation emerges when compared to the original 3D scenes contained within the point cloud pair, leading to false matches. The points of a patch tend to have similar features, which can challenge dense correspondence prediction. To overcome these limitations, FLAT merges the advantages of both cross-geometric structures and feature relationships, enhancing the accuracy of super-point matching. We introduce a distance-enhanced local self-attention mechanism that generates point-level features, improving dense matching capabilities.

Transformer on Registration. Transformer attention [29] has recently been successful in point cloud tasks due to its ability to learn long-range dependencies and invariance to input token permutations. Using the Transformer has been shown to enhance point cloud registration performance effectively. DCP [33] applies standard cross-attention to high-

light similarities of matched points across two point clouds for soft correspondence generation. The Geometric Transformer [27] aims to improve feature matching accuracy by enhancing the effectiveness of self-attention in acquiring geometric information. Their method encodes both pairwise distances and triplet-wise angles, allowing it to handle low-overlap scenarios while remaining invariant to rigid transformations. REGTR [39] integrates the Transformer [29] into a network that generates soft correspondences from local features, allowing feature matching for point clouds with partial overlaps. Predator [13] and PRNet [34] apply Transformer to detect points in the overlap region and use the features of the detected points to generate matches. Several of these methods fail to consider the fact that multiple regions within a point cloud may display similar structures, which can limit the effectiveness of standard cross-attention when dealing with comparable local structures. Motivated by this, we propose an improved cross-attention that incorporates transformation-invariant geometric structure into learned features to better highlight overlap regions of both point clouds.

3. Our approach

3.1. Overview

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 are 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 $\boldsymbol{\mathcal{Q}}$. Fig. 1 shows FLAT's pipeline, which adopts a hierarchical paradigm to predict correspondences in a coarse-to-fine manner. Given the pair of point clouds $\boldsymbol{\mathcal{P}}$ and $\boldsymbol{\mathcal{Q}}$, the encoder aggregates the raw points into superpoints $\bar{\boldsymbol{\mathcal{P}}}$ and $\bar{\boldsymbol{\mathcal{Q}}}$, while jointly learning their associated characteristics $\boldsymbol{\mathcal{F}}_{\bar{p}}$ and $\boldsymbol{\mathcal{F}}_{\bar{q}}$. The full geometric Transformer block updates the features as $\hat{\boldsymbol{\mathcal{F}}}_{\bar{p}}$ and $\hat{\boldsymbol{\mathcal{F}}}_{\bar{q}}$. A *Coarse Matching* module is applied to extract super-point correspondences whose neighboring local patches overlap with each other. The decoder transforms the features into per-point features $\boldsymbol{\mathcal{F}}_p$, $\boldsymbol{\mathcal{F}}_q$. Then, we apply a local geometric self-attention on each patch to refine the features. Lastly, a *Fine Matching* module extracts cluster-level correspondences, which are then used to estimate the transformation.

3.2. Feature Extraction

Encoder. We use KPConv-FPN [28] to downsample $\boldsymbol{\mathcal{P}}$ 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 $\bar{\boldsymbol{\mathcal{Q}}} = \{\bar{\boldsymbol{q}}_j \in \mathbb{R}^3 | j = 1, 2, ..., \bar{M}\}$, and to extract associated point-wise features $\boldsymbol{\mathcal{F}}_{\bar{p}} = \{\boldsymbol{f}_{\bar{p}} \in \mathbb{R}^b | i = 1, 2, ..., \bar{N}\}$ and $\boldsymbol{\mathcal{F}}_{\bar{q}} = \{\boldsymbol{f}_{\bar{q}_j} \in \mathbb{R}^b | j = 1, 2, ..., \bar{M}\}$, respectively, with $b = 256$. KPConv-FPN consists of a series of ResNet-like blocks and stridden convolutions.

3.3. Full Geometric Transformer

Global Geometric Self-attention. We use the Geometric self-attention as it is implemented in GeoTransformer [27].

Global Geometric Cross-attention. We develop a new fully-geometric cross-attention strategy that can identify global correlations between super-points in two point clouds, based on both feature and coordinate information. Given two super-points $\bar{\boldsymbol{p}}_i \in \bar{\boldsymbol{\mathcal{P}}}$ and $\bar{\boldsymbol{q}}_j \in \bar{\boldsymbol{\mathcal{Q}}}$, the cross-attention output $\hat{\boldsymbol{f}}_{\bar{p}_i} \in \hat{\boldsymbol{\mathcal{F}}}_{\bar{p}}$ for $\bar{\boldsymbol{p}}_i$ can be obtained by computing the weighted sum of all projected input features:

$$\hat{\boldsymbol{f}}_{\bar{p}_i} = \sum_{j=1}^{\bar{M}} \alpha_{ij} \boldsymbol{f}_{\bar{q}_j} \boldsymbol{W}^V, \tag{1}$$

where the weight coefficient α_{ij} is calculated using row-wise softmax on the attention scores as:

$$\alpha_{ij} = \text{softmax}\left(\boldsymbol{f}_{\bar{p}_i} \boldsymbol{W}^Q \left(\boldsymbol{f}_{\bar{q}_j} \boldsymbol{W}^K + \boldsymbol{r}_{ij} \boldsymbol{W}^R\right)^\top / \sqrt{b}\right), \tag{2}$$

where $\boldsymbol{W}^Q, \boldsymbol{W}^K, \boldsymbol{W}^R, \boldsymbol{W}^V$ correspond to the projection of queries, keys, values, and geometric structure embeddings, respectively. $\boldsymbol{r}_{ij}$ is the embedding of the cross-geometric structure of distance and angle that we compute as follows.

The distance and angle calculated in GeoTransformer are invariant only for rigid transformations within a single point cloud [27]. However, these measures are not invariant to rigid transformations when calculated between different point clouds that live in different reference frames. To address this issue, we have developed a new way of computing distance and angle measures to compare points between different point clouds. As shown in Fig. 2, we first uniformly sample $k = 10$ neighbors $\Omega_{\bar{p}_i}$ of $\bar{\boldsymbol{p}}_i$ in a ball of radius $r > 0$, and compute the covariance matrix $\Sigma_{\bar{p}_i}$ of these neighbouring points as:

$$\begin{aligned} \Sigma_{\bar{\boldsymbol{p}}_i} &= \sum_{\boldsymbol{x}_t \in \Omega_{\bar{\boldsymbol{p}}_i}} \omega_{x_t} (\boldsymbol{x}_t - \bar{\boldsymbol{p}}_i)(\boldsymbol{x}_t - \bar{\boldsymbol{p}}_i)^\top, \\ \omega_{x_t} &= \frac{\phi - \|\boldsymbol{x}_t - \bar{\boldsymbol{p}}_i\|_2}{\sum_{\boldsymbol{x}_t \in \Omega_{\bar{\boldsymbol{p}}_i}} (\phi - \|\boldsymbol{x}_t - \bar{\boldsymbol{p}}_i\|_2)}, \end{aligned} \tag{3}$$

where $\phi = \max_{\boldsymbol{x}_t \in \Omega_{\bar{\boldsymbol{p}}_i}} \|\boldsymbol{x}_t - \bar{\boldsymbol{p}}_i\|_2$. For $\bar{\boldsymbol{p}}_i$, we calculate an eigenvalue tuple of $\Sigma_{\bar{\boldsymbol{p}}_i}$ denoted by $\boldsymbol{\lambda}_{\bar{p}_i} = (\lambda^1_{\bar{p}_i}, \lambda^2_{\bar{p}_i}, \lambda^3_{\bar{p}_i})$. with $\lambda^1_{\bar{p}_i} \leq \lambda^2_{\bar{p}_i} \leq \lambda^3_{\bar{p}_i}$. The $\Sigma_{\bar{\boldsymbol{p}}_i}$ is transformation invariant. We sign the eigenvector associated with $\lambda^1_{\bar{p}i}$ as $\boldsymbol{v}_{\bar{p}_i}$, ensuring that its orientation is positive from the center of the point cloud towards the current point. Using the same operation, we get $\boldsymbol{\lambda}_{\bar{q}_j}$ for $\bar{\boldsymbol{q}}_j \in \bar{\boldsymbol{\mathcal{Q}}}$ and $\boldsymbol{v}_{\bar{q}_j}$.

For distance embedding, we choose the Gromov-Wasserstein distance [25], which is transformation invariant, and it can calculate the distance between metrics defined within each of the source and target spaces. Therefore, we use the GWD map to measure the transformation

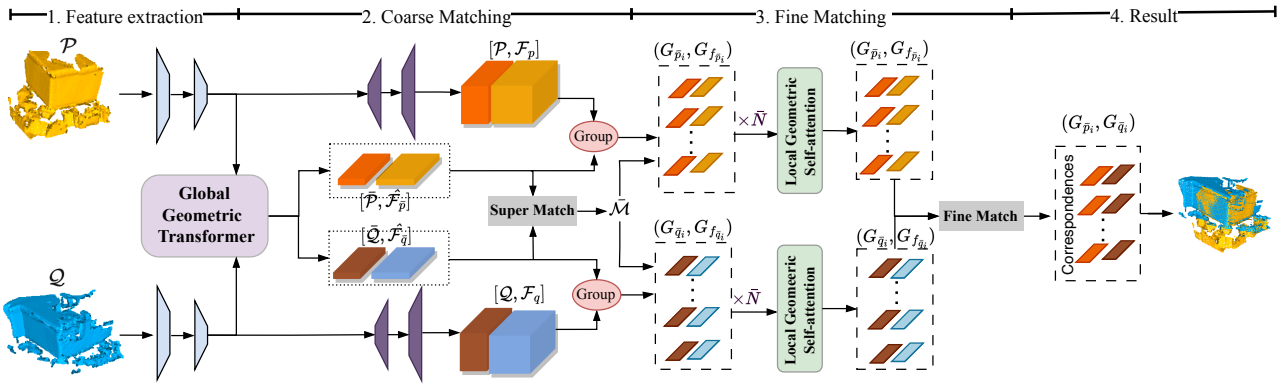

Figure 1. The encoder downsizes the input point clouds and generates super-points with associated features. The global geometric Transformer injects geometric information into learned features. Coarse matching of the super-points is carried out between the two downsampled inputs. The local geometric Transformer is utilized to generate distinctive local features, which enables the prediction of fine-level correspondences between the inputs. Finally, the rigid transformation is estimated from the fine-level correspondences.

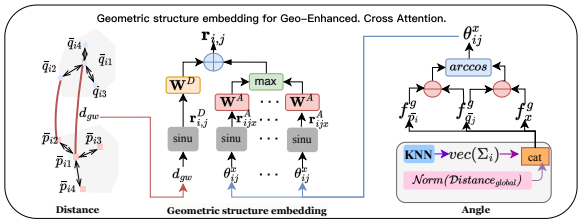

Figure 2. Angle and distance provide geometry information for cross-attention.

relationship $\boldsymbol{S} = \{s_{ij}\}$ of points from $\bar{\boldsymbol{\mathcal{P}}}$ and $\bar{\boldsymbol{\mathcal{Q}}}$, which formulates as:

$$
\begin{aligned}
& d_{gw}\left(\bar{\boldsymbol{\mathcal{P}}}, \bar{\boldsymbol{\mathcal{Q}}}\right) = \min_{S \geq 0} \sum_{stkl} \left(C_{sk}^{\bar{p}} - C_{tl}^{\bar{q}}\right)^2 \boldsymbol{S}_{st} \boldsymbol{S}_{kl}, \\
& \text{s.t.,} \begin{cases} C_{sk}^{\bar{p}} = \|\bar{\boldsymbol{p}}_s - \bar{\boldsymbol{p}}_k\|_2^2 + \alpha \|\boldsymbol{v}_{\bar{p}_s} - \boldsymbol{v}_{\bar{p}_k}\|_2^2, \\ C_{sk}^{\bar{q}} = \|\bar{\boldsymbol{q}}_t - \bar{\boldsymbol{q}}_l\|_2^2 + \alpha \|\boldsymbol{v}_{\bar{q}_t} - \boldsymbol{v}_{\bar{q}_l}\|_2^2, \end{cases}
\end{aligned} \tag{4}
$$

where $\alpha > 0$ is a learning parameter. The $\boldsymbol{S}_{ij}$ describes the similarity between the points $\bar{\boldsymbol{p}}_i$ and $\bar{\boldsymbol{q}}_j$. Eq. (4) is solved using the Sinkhorn algorithm [25].

For angle computation, we first define $\boldsymbol{\lambda}_c = \frac{1}{\bar{N}+\bar{M}} \left(\sum_{i=1}^{\bar{N}} \boldsymbol{\lambda}_{\bar{p}_i} + \sum_{j=1}^{\bar{M}} \boldsymbol{\lambda}_{\bar{q}_j}\right)$ as the center of union of $\bar{\boldsymbol{\mathcal{P}}}$ and $\bar{\boldsymbol{\mathcal{Q}}}$ in the eigenvalue space. For each $\bar{\boldsymbol{p}}_i$, we compute a feature vector $\boldsymbol{f}_{\bar{p}_i}^g = \text{cat}\left[\frac{\|\boldsymbol{\lambda}_{\bar{p}_i} - \boldsymbol{\lambda}_c\|_2}{\max\limits_j \|\bar{\boldsymbol{p}}_j - \bar{\boldsymbol{p}}_c\|_2}, \boldsymbol{\lambda}_{\bar{p}_i}\right]$ that encodes the geometric and spatial properties of the local patch of a single point within a single point cloud and the global structure information from two point clouds. cat$[\cdot,\cdot]$ concatenates two vectors into a single vector. Applying the same operator to $\bar{\boldsymbol{q}}_j \in \bar{\boldsymbol{\mathcal{Q}}}$, we can obtain $\boldsymbol{f}_{\bar{q}_j}^g$ for $\bar{\boldsymbol{q}}_j$. After that, for each $\boldsymbol{x} \in \Omega_{\bar{\boldsymbol{p}}_i}$, we compute the angle θ_{ij}^x between the vectors $\boldsymbol{f}_{\bar{p}_i}^g - \boldsymbol{f}_{\bar{q}_j}^g$ and $\boldsymbol{f}_x^g - \boldsymbol{f}_{\bar{q}_j}^g$ as:

$$
\theta_{ij}^x = \arccos \frac{\left\langle \boldsymbol{f}_{\bar{p}_i}^g - \boldsymbol{f}_{\bar{q}_j}^g, \boldsymbol{f}_x^g - \boldsymbol{f}_{\bar{q}_j}^g \right\rangle}{\|\boldsymbol{f}_{\bar{p}_i}^g - \boldsymbol{f}_{\bar{q}_j}^g\| \|\boldsymbol{f}_x^g - \boldsymbol{f}_{\bar{q}_j}^g\|}. \tag{5}
$$

The geometric structure embedding of $\bar{\boldsymbol{p}}_i \in \bar{\boldsymbol{\mathcal{P}}}$ and $\bar{\boldsymbol{q}}_j \in \bar{\boldsymbol{\mathcal{Q}}}$ comprises a pair-wise cross distance embedding and a triplet-wise cross angular embedding defined as follows:

- `Pair-wise Cross Distance Embedding`. The distance embedding $\boldsymbol{r}_{ij}^D$ between $\bar{\boldsymbol{p}}_i$ and $\bar{\boldsymbol{q}}_j$ is calculated as $\boldsymbol{r}_{ij}^D$=sinu$\left(\left(1 - s_{ij}\right)/\sigma_\rho\right)$, where sinu$(\cdot)$ is a sinusoidal function as in [29], and σ_ρ>0 is a learning parameter.
- `Triplet-wise Cross Angle Embedding`. Angular embedding is computed by using triplets of super-points. Specifically, for each $\boldsymbol{x} \in \Omega_{\bar{\boldsymbol{p}}_i}$, the triplet-wise angular embedding $\boldsymbol{r}_{ijx}^A$ is computed as $\boldsymbol{r}_{ijx}^A$=sinu$(\theta_{ij}^x/\sigma_\theta)$ with a learning parameter σ_θ>0.

Lastly, the geometric structure embedding $\boldsymbol{r}_{ij}$ is obtained by combining the pair-wise distance embedding and the triplet-wise angular embedding through aggregation, expressed as:

$$
\boldsymbol{r}_{ij} = \boldsymbol{r}_{ij}^D \boldsymbol{W}^D + \max_x \{\boldsymbol{r}_{ijx}^A \boldsymbol{W}^A\}, \tag{6}
$$

where $\boldsymbol{W}^D, \boldsymbol{W}^A$ are projection matrices corresponding to the two kinds of embeddings, respectively. The latent features $\hat{\boldsymbol{\mathcal{F}}}_{\bar{p}}$ then carry the knowledge of $\hat{\boldsymbol{\mathcal{F}}}_{\bar{q}}$, and vice versa.

Decoder. The decoder, based on KPConv layers [28], 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 the 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 $\boldsymbol{\mathcal{F}}_q \in \mathbb{R}^{M \times 32}$ are obtained in the same way.

Local Geometric Self-attention. For each super-point, we first construct a local patch of points around it using the point-to-node grouping strategy [40]. For a super-point $\bar{\boldsymbol{p}}_i \in \bar{\boldsymbol{\mathcal{P}}}$, its associated point set $G_{\bar{p}_i}$ and feature set $G_{\boldsymbol{f}_{\bar{p}_i}}$ are denoted as:

$$
\begin{cases} G_{\bar{p}_i} = \{\boldsymbol{p} \in \boldsymbol{\mathcal{P}} \,|\, \|\boldsymbol{p} - \bar{\boldsymbol{p}}_i\|_2 \leq \|\boldsymbol{p} - \bar{\boldsymbol{p}}_j\|_2, i \neq j\}, \\ G_{\boldsymbol{f}_{\bar{p}_i}} = \{\boldsymbol{f}_{\boldsymbol{x}_j} \in \boldsymbol{\mathcal{F}}_p \,|\, \boldsymbol{x}_j \in G_{\bar{p}_i}\}. \end{cases} \tag{7}
$$

In a similar way, we can get $G_{\bar{q}_i}$ and $G_{\boldsymbol{f}_{\bar{q}_i}}$.

Given the local patch $G_{\bar{p}_i} = \{\bar{\boldsymbol{p}}_{i1}, \bar{\boldsymbol{p}}_{i2}, \cdots, \bar{\boldsymbol{p}}_{iK}\}$, to perform local geometric attention, we first calculate the distance matrix $D^i = \{d^i_{kl}\}^K_{k,l=1}$, where $d^i_{kl} = \{\bar{\boldsymbol{p}}_{i1}, \bar{\boldsymbol{p}}_{i2}, \cdots, \bar{\boldsymbol{p}}_{iK}\}$ with $d^i_{kl} = \|\bar{\boldsymbol{p}}_{ik} - \bar{\boldsymbol{p}}_{il}\|^2_2$. Distance-based weights are calculated by $R^i = \texttt{DS}(D^i) = \{r^i_{kl}\}$. $\texttt{DS}(\dot{})$ is a Dual Softmax operator. Then, the self-attention output $\boldsymbol{f}^i_{\bar{p}_{ik}} \in G_{\boldsymbol{f}_{\bar{p}_i}}$ for $\bar{\boldsymbol{p}}_{ik}$ can be updated by computing the weighted sum of all weighted input features:

$$\boldsymbol{f}^i_{\bar{p}_{ik}} = \sum_{l=1}^{K} \frac{\beta_{kl}}{\sum_j \beta_{kj}} \boldsymbol{f}^i_{\bar{p}_{il}}, \tag{8}$$

where the weight coefficient β_{kl} is calculated using row-wise softmax on the attention scores as:

$$\beta_{kl} = r^i_{kl} \cdot \text{softmax}\left(\frac{\boldsymbol{f}^i_{\bar{p}_{ik}} \boldsymbol{W}^Q_i (\boldsymbol{f}^i_{\bar{p}_{il}} \boldsymbol{W}^K_i)^\top}{\sqrt{b}}\right), \tag{9}$$

We also applied the same operator to update $G_{\boldsymbol{f}_{\bar{q}_i}}$.

3.3.1 Correspondence Prediction

Coarse Matching. Coarse Matching estimates super-point correspondences between $\bar{\boldsymbol{\mathcal{P}}}$ and $\bar{\boldsymbol{\mathcal{Q}}}$, which can be formulated as an assignment problem and solved by calculating an assignment matrix $\bar{\Gamma} \in \mathbb{R}^{\bar{N} \times \bar{M}}$ as:

$$\min_{\bar{\Gamma}} \left\langle \bar{\boldsymbol{C}}, \bar{\Gamma} \right\rangle, \tag{10}$$

where $\langle \cdot, \cdot \rangle$ is the Frobenius dot product. $\bar{\boldsymbol{C}}$ is a distance matrix with elements that satisfy

$$\bar{\boldsymbol{C}}_{ij} = (1-\eta)\| \frac{\hat{\boldsymbol{f}}_{\bar{p}_i}}{\|\hat{\boldsymbol{f}}_{\bar{p}_i}\|} - \frac{\hat{\boldsymbol{f}}_{\bar{p}_j}}{\|\hat{\boldsymbol{f}}_{\bar{p}_j}\|} \| + \eta \| \frac{\boldsymbol{f}^g_{\bar{p}_i}}{\|\boldsymbol{f}^g_{\bar{p}_i}\|} - \frac{\boldsymbol{f}^g_{\bar{q}_j}}{\|\boldsymbol{f}^g_{\bar{q}_j}\|} \|,$$

with $\hat{\boldsymbol{f}}_{\bar{p}_i} \in \hat{\boldsymbol{\mathcal{F}}}, \hat{\boldsymbol{f}}_{\bar{q}_j} \in \hat{\boldsymbol{\mathcal{F}}}_{\bar{q}}$, and each $\bar{\Gamma}_{ij} \in \bar{\Gamma} \in [0,1]^{\bar{N} \times \bar{M}}$ represents the matching score between $\bar{\boldsymbol{p}}_i$ and $\bar{\boldsymbol{q}}_j$. $\eta = 0.1$ is a parameter that controls the weight of the feature distance and the structure distance. Eq. (10) is an example of the optimal transport problem [9] and can be solved efficiently using the Sinkhorn-Knopp algorithm [9]. After computing $\bar{\Gamma}$, we select correspondences with confidence higher than a threshold τ_c, and enforce the mutual nearest neighbor (MNN) constraint to have fewer but reliable correspondences. The super-point correspondence set $\bar{\mathcal{M}}$ is then defined as:

$$\bar{\mathcal{M}} = \{(\bar{\boldsymbol{p}}_{\hat{i}}, \bar{\boldsymbol{q}}_{\hat{j}}) \big| \forall (\hat{i}, \hat{j}) \in \text{MNN}(\bar{\Gamma}), \bar{\Gamma}_{\hat{i},\hat{j}} \geq \tau_c\}. \tag{11}$$

Fine Matching. Extracting point correspondences is analogous to matching two smaller-scale point clouds by solving an optimal transport problem to calculate a matrix $\Gamma^{G_{\bar{p}_i}}$ in a manner of coarse-level done. For correspondences, we choose the maximum confidence score of $\Gamma^{G_{\bar{p}_i}}$ in every row and column to guarantee higher precision. The final point correspondence set $\mathcal{M}$ is represented as the union of all the correspondence sets obtained. After obtaining the correspondences $\mathcal{M}$, following [27, 40], a variant of RANSAC [10] that is specialized in 3D correspondence-based registration [45] is utilized to estimate the transformation.

3.4. Loss Function and Training

We train FLAT using ground-truth correspondences as supervision. The loss function is: $\mathcal{L} = \mathcal{L}_C + \mathcal{L}_F$, where $\mathcal{L}_C$ and $\mathcal{L}_F$ denote a coarse and a fine matching loss.

Coarse Matching Loss. Following [11, 40], we formulate super-point matching as a multilabel classification problem and adopt a cross-entropy loss with optimal transport. As there is no direct supervision for super-point matching, we leverage the overlap ratio r_{ij} of points in $G_{\bar{p}_i}$ that have correspondences in $G_{\bar{q}_j}$ to depict the matching probability between super-points $\bar{p}_i$ and $\bar{q}_j$. r_{ij} is defined as:

$$r_{ij} = \frac{1}{|G_{\bar{p}_i}|} |\{\boldsymbol{p} \in G_{\bar{p}_i} | \min_{\boldsymbol{q} \in G_{\bar{q}_j}} \|\hat{\boldsymbol{T}}(\boldsymbol{p}) - \boldsymbol{q}\|_2 < r_p\}|, \tag{12}$$

where $\hat{\boldsymbol{T}}$ is the ground-truth transformation and r_p is a set threshold. All other pairs are omitted. We select the super-points in $\bar{\boldsymbol{\mathcal{P}}}$ which have at least one positive super-point in $\bar{\boldsymbol{\mathcal{Q}}}$ to form a set of anchor super-points, $\tilde{\boldsymbol{\mathcal{P}}}$. Based on Eq. (12), we define the weight matrix $\bar{\mathbf{W}} \in \mathbb{R}^{\bar{N} \times \bar{M}}$ as:

$$\bar{\mathbf{W}}(i,j) = \begin{cases} r(i,j), & i \leq \bar{N} \wedge j \leq \bar{M}, \\ 0, & \text{otherwise.} \end{cases}$$

Finally, the coarse matching loss can be written as:

$$\mathcal{L}_C = -\frac{\sum_{i,j} \bar{\mathbf{W}}(i,j) \log\left(\bar{\mathbf{\Gamma}}(i,j)\right)}{\sum_{i,j} \bar{\mathbf{W}}(i,j)}. \tag{13}$$

Fine Matching Loss. We apply the circle loss again to supervise the point matching. Consider a pair of matched super-points $\bar{\boldsymbol{p}}_i$ 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 $\boldsymbol{g}^k_{\bar{p}_i} \in \tilde{G}_{\bar{p}_i}$ has at least one (possibly multiple) correspondence in $G_{\bar{q}_j}$, i.e.,

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

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

$$\mathcal{L}^{\mathcal{P}}_F = \frac{1}{|\tilde{\boldsymbol{\mathcal{P}}}|} \sum_{\tilde{\boldsymbol{p}}_i \in \tilde{\boldsymbol{\mathcal{P}}}} \frac{1}{|\tilde{G}_{\bar{p}_i}|} \sum_{\boldsymbol{g}^s_{\bar{p}_i} \in \tilde{G}_{\bar{p}_i}} \log\left[1 + \xi_s\right],$$

$$\xi_s = \sum_{\boldsymbol{g}^k_{\bar{q}_j} \in \mathcal{N}_p^{\boldsymbol{g}^s_{\bar{p}_i}}} e^{r^k_s \beta^{sk}_p (d^k_s - \Delta p)} \cdot \sum_{\boldsymbol{g}^l_{\bar{q}_j} \in \mathcal{N}_n^{\boldsymbol{g}^s_{\bar{p}_i}}} e^{\beta^{sl}_n (\Delta n - d^l_s)},$$

where $d_s^k = \mathcal{D}_f(\boldsymbol{f}_{\boldsymbol{g}_{\tilde{p}_i}^s}, \boldsymbol{f}_{\boldsymbol{g}_{\tilde{q}_j}^s})$ 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 $\mathcal{L}_F^{\mathcal{Q}}$ on $\mathcal{Q}$. The overall super-point matching loss is written as $\mathcal{L}_F = \frac{1}{2}(\mathcal{L}_F^{\mathcal{P}} + \mathcal{L}_F^{\mathcal{Q}})$.

4. Experiments

We evaluate FLAT on typical point cloud registration benchmarks, i.e. indoor 3DMatch [42] and 3DLoMatch [13], outdoor KITTI [12], and cross-source 3DCSR [15]. Please refer to Appendix Section 1 for detailed implementation, including running details, network pipeline, and correspondence sampling. For cluster numbers and the time computational analysis, and additional qualitative results please refer to the Sec. 2 in the Supplementary Material.

4.1. Evaluation on 3DMatch and 3DLoMatch

Datasets. 3DMatch [42] and 3DLoMatch [13] are two widely used indoor benchmarks that contain more than 30% and 10% to 30% partially overlapping scene pairs, respectively. 3DMatch contains 62 scenes: we use 46 scenes for training, 8 scenes for validation, and 8 scenes for testing. The test set contains 1,623 partially overlapped point cloud fragments and their corresponding transformation matrices. We use training data preprocessed by [13] and evaluate on both the 3DMatch and 3DLoMatch [13] protocols. We first voxelize the point clouds with a $2.5cm$ voxel size and then extract different feature descriptors. We set $\tau_c = 0.15$, $r = r_o = r_p = 3.75cm$, and $r_n = 10.0cm$ [13].

Metrics. Following Predator [13] and CoFiNet [40], we evaluate performance with three metrics: (i) *Inlier Ratio* (IR), the fraction of putative correspondences whose residuals are below a certain threshold (i.e. 0.1m) under the ground-truth transformation, (ii) *Feature Matching Recall* (FMR), the fraction of point cloud pairs whose inlier ratio is above a certain threshold (i.e. 5%), and (iii) *Registration Recall* (RR), the fraction of point cloud pairs whose transformation error is smaller than a certain threshold (i.e. $RMSE < 0.2m$). We compare FLAT with FCGF [7], D3Feat [3], SpinNet [1], Predator [13], YOHO [30], CoFiNet [40], GeoTransformer [27] short as GeoTrans, GLORN [37], and RoITr [41].

Inlier Ratio and Feature Matching Recall. The primary contribution of our FLAT is its use of geometric cross-attention to emphasize matched point pair similarities and estimate more accurate correspondences. Therefore, we begin by examining the inlier ratio of the correspondences generated by FLAT, which directly reflects the quality of the extracted correspondences. Following [13], we present the results on varying sampled numbers of correspondences. Table 1 (top) shows that FLAT outperforms all previous methods in terms of Inlier Ratio on both benchmarks. In particular, FLAT consistently outperforms RoITr, the second best baseline, by $0.5\% \sim 1.2\%$ on 3DMatch and $1.8\% \sim 2.1\%$ on 3DLoMatch, with sample numbers ranging from 250 to 5000. This notable increase in the Inlier Ratio suggests that incorporating the cross-geometric structure effectively enhances the reliability of correspondence production. Additionally, FLAT outshines all competitors in Feature Matching Recall, as detailed in Table 1 (middle section). Particularly in the challenging low-overlap scenarios of 3DLoMatch, our method demonstrates its robustness with improvements exceeding 0.4%, underscoring its efficacy in complex situations.

Table 1. Results on both 3DMatch and 3DLoMatch datasets under different numbers of samples. Best performance in bold.

	3DMatch					3DLoMatch				
# Samples	5000	2500	1000	500	250	5000	2500	1000	500	250
Method	Inlier Ratio (%) ↑									
FCGF[7]	56.8	54.1	48.7	42.5	34.1	21.4	20.0	17.2	14.8	11.6
D3Feat[3]	39.0	38.8	40.4	41.5	41.8	13.2	13.1	14.0	14.6	15.0
SpinNet [1]	47.5	44.7	39.4	33.9	27.6	20.5	19.0	16.3	13.8	11.1
Predator [13]	58.0	58.4	57.1	54.1	49.3	26.7	28.1	28.3	27.5	25.8
CoFiNet[40]	49.8	51.2	51.9	52.2	52.2	24.4	25.9	26.7	26.8	26.9
YOHO [30]	64.4	60.7	55.7	46.4	41.2	25.9	23.3	22.6	18.2	15.0
GeoTrans[27]	71.9	75.2	76.0	82.2	85.1	43.5	45.3	46.2	52.9	57.7
GLORN [37]	72.6	73.5	76.4	82.3	83.2	42.3	45.6	46.5	53.1	**57.9**
RoITr [41]	82.6	82.8	83.0	83.0	83.0	54.3	54.6	55.1	55.2	55.3
FLAT (Ours)	**83.1**	**83.6**	**84.2**	**84.2**	**84.1**	**56.1**	**56.4**	**57.3**	**57.3**	57.4
	Feature Matching Recall (%) ↑									
FCGF[7]	97.4	97.3	97.0	96.7	96.6	76.6	75.4	74.2	71.7	67.3
D3Feat [3]	95.6	95.4	94.5	94.1	93.1	67.3	66.7	67.0	66.7	66.5
SpinNet [1]	97.6	97.2	96.8	95.5	94.3	75.3	74.9	72.5	70.0	63.6
Predator[13]	96.6	96.6	96.5	96.3	96.5	78.6	77.4	76.3	75.7	75.3
CoFiNet[40]	98.1	98.3	98.1	**98.2**	98.3	83.1	83.5	83.3	83.1	82.6
YOHO[30]	98.2	97.6	97.5	97.7	96.0	79.4	78.1	76.3	73.8	69.1
GeoTrans[27]	97.9	97.9	97.9	97.9	97.6	88.3	88.6	88.8	88.6	88.3
GLORN [37]	97.8	97.9	98.0	98.1	97.2	87.8	88.8	88.9	88.7	88.5
RoITr [41]	98.0	98.0	97.9	98.0	97.9	**89.6**	**89.6**	89.5	89.4	89.3
FLAT (Ours)	**98.2**	**98.2**	**98.1**	**98.2**	**98.3**	89.5	**89.6**	**89.6**	**89.5**	**89.4**
	Registration Recall (%) ↑									
FCGF[7]	85.1	84.7	83.3	81.6	71.4	40.1	41.7	38.2	35.4	26.8
D3Feat[3]	81.6	84.5	83.4	82.4	77.9	37.2	42.7	46.9	43.8	39.1
SpinNet[1]	88.6	86.6	85.5	83.5	70.2	59.8	54.9	48.3	39.8	26.8
Predator[13]	89.0	89.9	90.6	88.5	86.6	59.8	61.2	62.4	60.8	58.1
CoFiNet[40]	89.3	88.9	88.4	87.4	87.0	67.5	66.2	64.2	63.1	61.0
YOHO [30]	90.8	90.3	89.1	88.6	84.5	65.2	65.5	63.2	56.5	48.0
GeoTrans[27]	92.0	91.8	91.8	91.4	91.2	75.0	74.8	74.2	74.1	73.5
GLORN [37]	92.2	91.6	91.9	91.5	91.0	74.9	75.2	74.4	74.3	73.6
RoITr [41]	91.9	91.7	91.8	91.4	91.0	74.7	74.8	74.8	74.2	73.6
FLAT (Ours)	**92.4**	**92.1**	**92.2**	**91.8**	**91.5**	**78.6**	**78.7**	**78.1**	**76.4**	**75.2**

Registration Recall. The Registration Recall (RR) reflects the final performance on point cloud registration. Table 1 (bottom) shows that our method outperforms all other models on both datasets in terms of RR. FLAT achieves RR of 92.4% and 78.7% on both 3DMatch and 3DLoMatch, surpassing the previous best performance achieved by GeoTransformer (which has a RR of 92.0% on 3DMatch and 75.0% on 3DLoMatch) by 0.4% and 3.7%, respectively. This shows that incorporating geometrical information into

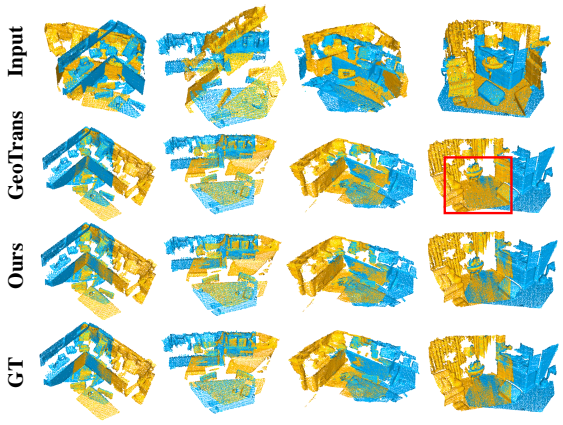

Figure 3. Examples of qualitative registration results on the 3DMatch dataset. Inaccurate regions are enclosed in red boxes.

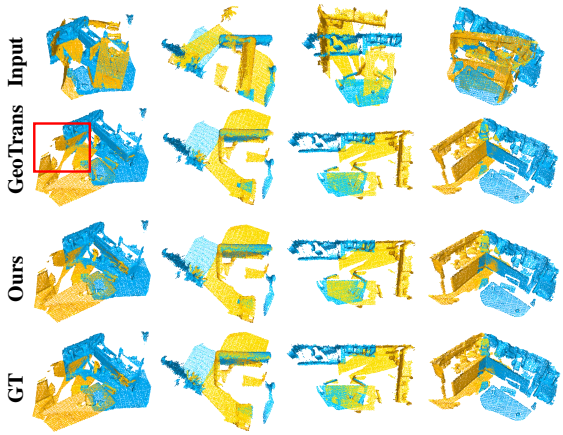

Figure 4. Examples of qualitative registration results on the 3DLoMatch dataset. Inaccurate regions are enclosed in red boxes.

the correspondence prediction process can alleviate ambiguity and lead to superior performance compared to methods that only consider feature similarity in cross-attention. Figs. 3 and 4 show comparison examples on 3DMatch and 3DLoMatch, respectively. FLAT achieves better results in challenging indoor scenes with a low overlap ratio.

4.2. Evaluation on KITTI

Dataset. KITTI consists of 11 sequences of LiDAR-scanned outdoor driving scenarios. To ensure fairness in comparisons, we adopt the same data-splitting approach as [6, 7], with sequences 0-5 utilized for training, 6-7 for validation, and 8-10 for testing purposes. We refine the provided ground-truth poses by employing ICP and limit the evaluation to point cloud pairs within a distance of $10m$ from each other, as in [6]. Furthermore, following in [13], we downsample the point clouds using a voxel size $30cm$ and set $\tau_c = 0.15$, $r = r_o$=$45cm$, r_p=$21cm$, and r_n=$75cm$.
Metrics. Following Predator [13] and CoFiNet [40], our FLAT is evaluated using three metrics: Registration Recall (RR), Relative Rotation Error (RRE), and Relative Translation Error (RTE). RR calculates the percentage of successful alignments where the rotation and translation errors are below specified thresholds (i.e., RRE $< 5°$ and RTE $< 2m$). The definitions of RRE and RTE are RRE= $\arccos \frac{\mathbf{Tr}(\boldsymbol{R}^\top \boldsymbol{R}^\star)-1}{2}$ and RTE= $|\boldsymbol{t} - \boldsymbol{t}^\star|_2$, respectively. The ground-truth rotation matrix and the translation vector are denoted by $\boldsymbol{R}^\star$ and $\boldsymbol{t}^\star$.

Table 2. Results on KITTI dataset. Best performance in bold.

Method	RTE (cm) ↓	RRE (°) ↓	RR(%) ↑
FCGF [7]	9.5	0.30	96.6
D3Feat [3]	7.2	0.30	**99.8**
SpinNet [1]	9.9	0.47	99.1
Predator [13]	6.8	0.27	**99.8**
CoFiNet [40]	8.5	0.41	**99.8**
GeoTrans [27]	7.4	0.27	**99.8**
FLAT (ours)	**6.9**	**0.24**	**99.8**

Table 3. Registration results on the 3DCSR dataset. Best performance in bold.

Method	RTE (cm) ↓	RRE (°) ↓	RR(%) ↑
FCGF [7]	**0.21**	7.47	49.6
D3Feat [3]	0.26	6.41	52.0
SpinNet [1]	0.24	6.56	53.5
Predator [13]	0.27	6.26	54.6
CoFiNet [40]	0.26	5.76	57.3
GeoTrans [27]	0.24	5.60	60.2
FLAT (ours)	0.22	**5.44**	**62.9**

Registration Results. We compare FLAT against FCGF [7], D3Feat [3], SpinNet [1], Predator [13], CoFiNet [40], and GeoTransformer [27]. Table 2 shows that our method achieves the best performance in terms of RR and the lowest average RTE and RRE. These results confirm the effectiveness of FLAT also in an outdoor scenario. It also suggests that incorporating cross-attention with geometry enhancement could be beneficial in acquiring more distinct features, resulting in better performance in registration.

4.3. Generalization on 3D Cross-Source Dataset

Dataset. 3DCSR [15] contains two sets: Kinect Lidar and Kinect SFM. Kinect Lidar comprises 19 scenes captured from both Kinect and Lidar sensors. Each scene is divided into different parts. Kinect SFM comprises 2 scenes captured from both Kinect and RGB-D sensors. RGB-D images are transformed into a point cloud by employing the VSFM software. The model trained on 3DMatch is used since the cross-source dataset is captured in an indoor setting. The metric used for successful alignment is RR, which represents the percentage of aligned scenes with RRE less than $15°$ and RTE less than $6m$. 3DCSR is a challenging dataset for registration due to a mixture of noise, outliers, density differences, and partial overlaps.
Registration Results. We use FCGF [7], D3Feat [3], SpinNet [1], Predator [13], CoFiNet [40], and GeoTransformer [27] as comparison methods. Table 3 shows that our method also achieves the highest accuracy in this experimental configuration, i.e. generalization ability. Notably, it surpasses GeoTransformer, the second-best, by more than 2.7% in

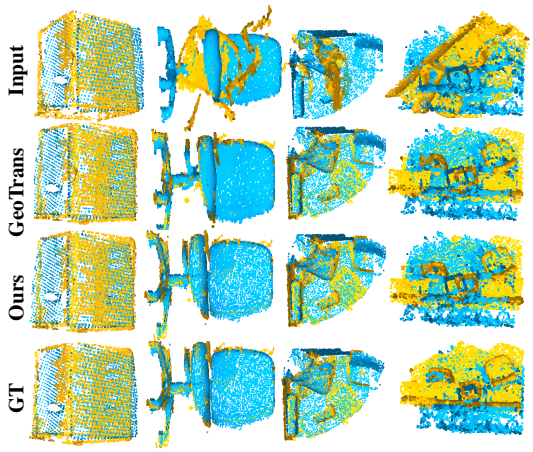

Figure 5. Examples of qualitative registration results on the 3DCSR dataset.

Table 4. Ablation study of model design.

Model	3DMatch			3DLoMatch		
	RR	FMR	IR	RR	FMR	IR
Cross-attention						
vanilla cross-attention	92.0	98.0	73.4	74.3	88.6	43.8
cross-attention w/PDE	92.1	98.1	81.4	75.6	88.7	54.4
cross-attention w/TAE	92.1	98.0	81.3	74.9	88.6	54.1
cross-attention w/DAE	**92.2**	**98.1**	**84.2**	**78.1**	**89.6**	**57.3**
Self-attention						
vanilla self-attention	92.1	98.0	83.9	75.1	88.8	56.4
self-attention w/PDE	**92.2**	**98.1**	**84.2**	**78.1**	**89.6**	**57.3**

terms of registration recall (62.9% vs 60.2%). Our ability to achieve better results is attributed to the cross-attention module enhanced by geometry. However, the recall rate falls short, indicating that the registration challenges on 3DCSR persist. Fig. 5 shows examples of qualitative results on 3DCSR.

4.4. Ablation Study

We conducted an ablation analysis on 3DMatch and 3DLoMatch with #Samples=1000 to examine the specific roles of each element in our approach. To assess the efficacy of geometry information in cross-attention, we compared the registration outcomes of four types of cross-attention - vanilla, pair-wise distance embedding-based (PDE), triplet-wise angle embedding-based (TAE), and the whole geometric cross-attention (DAE) in Table 4. Both pair-wise distance and triplet-wise angle embedding can improve the registration performance. To be more detailed, Geometric information improves the performance by nearly 0.2% (92.0% vs. 92.2%) RR, 0.1% (98.0% vs. 98.1%) FMR, and 2.8% (81.4% vs. 84.2%) IR on 3DMatch indicating that FLAT benefits from pair-wise distance and triplet-wise angle embedding. Additionally, the study presents results for employing self-attention mechanisms combined with distance mapping, a technique that further enhanced performance. Fig. 6 presents two examples of attention maps. It includes only two source point clouds and omits target point clouds for clarity. Given a point from the source, the red boxes highlight the correct correspondence regions. Our network focuses on corresponding superpoints in the source point cloud for each point in the target cloud. We emphasize that attention is selectively focused on areas within the red rectangles, ensuring clarity on this key aspect. Additionally, the figure contrasts attention maps from two example configurations, showcasing our method's superior ability to identify overlapping regions compared to GeoTr. In these examples, a selected point in the target point cloud is matched with corresponding superpoints in the source point cloud. Blue points indicate low attention weights, while points in other colors signify higher attention weights. Notably, our approach focuses attention on the red rectangles.

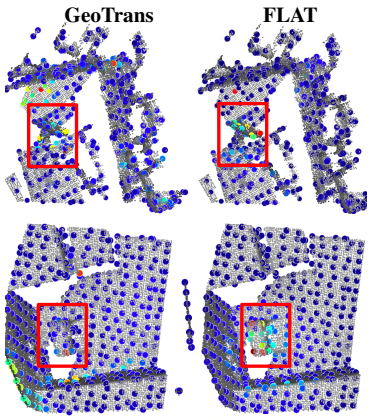

Figure 6. Visualization of attention weights for the point indicated with a red dot. Brighter colors indicate higher attention.

5. Conclusions

We present a fully-geometric attention algorithm to achieve accurate point cloud registration through coarse-to-fine matching. We fuse information from coordinates and features at the super-point level between point clouds, an unaddressed problem in the literature primarily because it must guarantee rotation and translation invariance as point clouds reside in different and independent reference frames. Cross-attention can identify overlap areas and accurately match coarse features. Instead, self-attention produces more distinctive local features for fine matching. The results showed that our approach achieves state-of-the-art performance on a large benchmark, including 3DMatch, 3DLoMatch, KITTI, and 3DCSR datasets.

Limitations. The time complexity of computing the pair-wise distance mapping is relatively high. However, this operation acts on superpoints, making it more manageable. Addressing the issue of dealing with source and target point clouds of different densities, especially since different sensors capture them, remains a challenge.

Acknowledgment. This work was sponsored by the FAIR - Future AI Research (PE00000013), funded by NextGeneration EU. Bruno Lepri and Nicu Sebe acknowledge funding by the European Union's Horizon Europe research and innovation program under grant agreement No. 101120237 (ELIAS).

References

[1] Sheng Ao, Qingyong Hu, Bo Yang, Andrew Markham, and Yulan Guo. Spinnet: Learning a general surface descriptor for 3d point cloud registration. In *CVPR*, pages 11753–11762, 2021. 2, 6, 7

[2] Yasuhiro Aoki, , and et al. Pointnetlk: Robust & efficient point cloud registration using pointnet. In *CVPR*, pages 7163–7172, 2019. 1

[3] Xuyang Bai and et al. D3feat: Joint learning of dense detection and description of 3d local features. In *CVPR*, pages 6359–6367, 2020. 2, 6, 7

[4] Xuyang Bai and et al. Pointdsc: Robust point cloud registration using deep spatial consistency. In *CVPR*, pages 15859–15869, 2021. 1

[5] 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. 1

[6] Christopher Choy and et al. Deep global registration. In *CVPR*, pages 2514–2523, 2020. 1, 2, 7

[7] Christopher Choy, Jaesik Park, and Vladlen Koltun. Fully convolutional geometric features. In *ICCV*, pages 8958–8966, 2019. 1, 2, 6, 7

[8] J. Corsetti, D. Boscaini, and F. Poiesi. Revisiting fully convolutional geometric features for object 6d pose estimation. In *ICCVW*, 2023. 1

[9] Marco Cuturi. Sinkhorn distances: Lightspeed computation of optimal transport. *NeurIPS*, 26:2292–2300, 2013. 5

[10] 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. 5

[11] Kexue Fu and et al. Robust point cloud registration framework based on deep graph matching. In *CVPR*, pages 8893–8902, 2021. 2, 5

[12] Andreas Geiger and et al. Are we ready for autonomous driving? the kitti vision benchmark suite. In *CVPR*, pages 3354–3361. IEEE, 2012. 2, 6

[13] Shengyu Huang and et al. Predator: Registration of 3d point clouds with low overlap. In *CVPR*, pages 4267–4276, 2021. 1, 2, 3, 6, 7

[14] 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. 1

[15] Xiaoshui Huang, Guofeng Mei, Jian Zhang, and Rana Abbas. A comprehensive survey on point cloud registration. *arXiv preprint arXiv:2103.02690*, 2021. 2, 6, 7

[16] 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*, 7 (3):7028–7035, 2022. 1

[17] Pileun Kim, Jingdao Chen, and Yong K Cho. Slam-driven robotic mapping and registration of 3d point clouds. *Automation in Construction*, 89:38–48, 2018. 1

[18] 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. 2

[19] Guofeng Mei. Point cloud registration with self-supervised feature learning and beam search. In *DICTA*, pages 01–08, 2021. 1, 2

[20] Guofeng Mei, Xiaoshui Huang, Jian Zhang, and Qiang Wu. Overlap-guided coarse-to-fine correspondence prediction for point cloud registration. In *ICME*, pages 1–6. IEEE, 2022. 1

[21] G. Mei, C. Saltori, F. Poiesi, J. Zhang, E. Ricci, N. Sebe, and Q. Wu. Data augmentation-free unsupervised learning for 3d point cloud understanding. In *BMVC*, 2022. 1

[22] 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. 1

[23] Weizhi Nie, Ruidong Chen, Weijie Wang, Bruno Lepri, and Nicu Sebe. T2td: Text-3d generation model based on prior knowledge guidance. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2024. 1

[24] 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. 2

[25] 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. 3, 4

[26] F. Poiesi and D. Boscaini. Generalisable and distinctive 3D local deep descriptors for point cloud registration. *arXiv:2105.10382*, 2021. 1, 2

[27] 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. 1, 2, 3, 5, 6, 7

[28] Hugues Thomas, Charles R Qi, Jean-Emmanuel Deschaud, Beatriz Marcotegui, François Goulette, and Leonidas J Guibas. Kpconv: Flexible and deformable convolution for point clouds. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 6411–6420, 2019. 3, 4

[29] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *NeurIPS*, 30, 2017. 1, 2, 3, 4

[30] Haiping Wang, Yuan Liu, Zhen Dong, and Wenping Wang. You only hypothesize once: Point cloud registration with rotation-equivariant descriptors. In *ACM MM*, pages 1630–1641, 2022. 2, 6

[31] Haiping Wang, Yuan Liu, Qingyong Hu, Bing Wang, Jianguo Chen, Zhen Dong, Yulan Guo, Wenping Wang, and Bisheng Yang. Roreg: Pairwise point cloud registration with oriented descriptors and local rotations. *TPAMI*, 2023. 1, 2

[32] Weijie Wang, Wenqi Ren, Guofeng Mei, Bin Ren, Xiaoshui Huang, Fabio Poiesi, Nicu Sebe, and Bruno Lepri. Zeroreg: Zero-shot point cloud registration with foundation models, 2024. 1

[33] Yue Wang and Justin M Solomon. Deep closest point: Learning representations for point cloud registration. In *ICCV*, pages 3523–3532, 2019. 2

[34] Yue Wang and Justin M Solomon. Prnet: Self-supervised learning for partial-to-partial registration. In *NeurIPS*, 2019. 3

[35] 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. 1

[36] Hao Xu, Shuaicheng Liu, Guangfu Wang, Guanghui Liu, and Bing Zeng. Omnet: Learning overlapping mask for partial-to-partial point cloud registration. In *ICCV*, pages 3132–3141, 2021. 1

[37] Jiabo Xu, Yukun Huang, Zeyun Wan, and Jingbo Wei. Glorn: Strong generalization fully convolutional network for low-overlap point cloud registration. *T-GE*, 60:1–14, 2022. 1, 6

[38] Zi Jian Yew and Gim Hee Lee. Rpm-net: Robust point matching using learned features. In *CVPR*, pages 11824–11833, 2020. 2

[39] Zi Jian Yew and Gim Hee Lee. Regtr: End-to-end point cloud correspondences with transformers. In *CVPR*, pages 6677–6686, 2022. 1, 3

[40] Hao Yu and et al. Cofinet: Reliable coarse-to-fine correspondences for robust pointcloud registration. *NeurIPS*, 34, 2021. 1, 2, 4, 5, 6, 7

[41] Hao Yu, Zheng Qin, Ji Hou, Mahdi Saleh, Dongsheng Li, Benjamin Busam, and Slobodan Ilic. Rotation-invariant transformer for point cloud matching. In *CVPR*, pages 5384–5393, 2023. 6

[42] 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. 2, 6

[43] Cheng Zhang, Haocheng Wan, Xinyi Shen, and Zizhao Wu. Patchformer: An efficient point transformer with patch attention. In *CVPR*, pages 11799–11808, 2022. 2

[44] Zhiyuan Zhang, Yuchao Dai, and Jiadai Sun. Deep learning based point cloud registration: an overview. *VRIH*, 2(3):222–246, 2020. 1

[45] Qian-Yi Zhou, Jaesik Park, and Vladlen Koltun. Open3d: A modern library for 3d data processing. *arXiv preprint arXiv:1801.09847*, 2018. 5