

GeoTransformer: Fast and Robust Point Cloud Registration with Geometric Transformer

Zheng Qin, Hao Yu, Changjian Wang, Yulan Guo, Yuxing Peng, Slobodan Ilic, Dewen Hu, Kai Xu*

Abstract—We study the problem of extracting accurate correspondences for point cloud registration. Recent keypoint-free methods have shown great potential through bypassing the detection of repeatable keypoints which is difficult to do especially in low-overlap scenarios. They seek correspondences over downsampled superpoints, which are then propagated to dense points. Superpoints are matched based on whether their neighboring patches overlap. Such sparse and loose matching requires contextual features capturing the geometric structure of the point clouds. We propose Geometric Transformer, or GeoTransformer for short, to learn geometric feature for robust superpoint matching. It encodes pair-wise distances and triplet-wise angles, making it invariant to rigid transformation and robust in low-overlap cases. The simplistic design attains surprisingly high matching accuracy such that no RANSAC is required in the estimation of alignment transformation, leading to 100 times acceleration. Extensive experiments on rich benchmarks encompassing indoor, outdoor, synthetic, multiway and non-rigid demonstrate the efficacy of GeoTransformer. Notably, our method improves the inlier ratio by 18~31 percentage points and the registration recall by over 7 points on the challenging 3DLoMatch benchmark. Our code and models are available at <https://github.com/qinzheng93/GeoTransformer>.

Index Terms—Point cloud registration, transformer, geometric consistency, coarse-to-fine correspondence, point cloud matching

1 INTRODUCTION

POINT cloud registration is a fundamental task in graphics, vision and robotics. Given two partially overlapping 3D point clouds, the goal is to estimate a rigid transformation that aligns them. The problem has gained renewed interest recently thanks to the fast growing of 3D point representation learning and differentiable optimization.

The recent advances have been dominated by learning-based, correspondence-based methods [1], [2], [3], [4], [5], [6]. A neural network is trained to extract point correspondences between two input point clouds, based on which an alignment transformation is calculated with a robust estimator, *e.g.*, RANSAC. Most correspondence-based methods rely on keypoint detection [3], [4], [5], [7]. However, it is challenging to detect repeatable keypoints across two point clouds, especially when they have small overlapping area. This usually results in low inlier ratio in the putative correspondences.

Inspired by the recent advances in image matching [8], [9], [10], keypoint-free methods [6] downsample the input point clouds into superpoints and then match them through examining whether their local neighborhood (patch) overlaps. Such superpoint (patch) matching is then propagated to individual points, yielding dense point correspondences. Consequently, the accuracy of dense point correspondences highly depends on that of superpoint matches.

Superpoint matching is sparse and loose. The upside is that it reduces strict point matching into loose patch overlapping, thus relaxing the repeatability requirement. Mean-

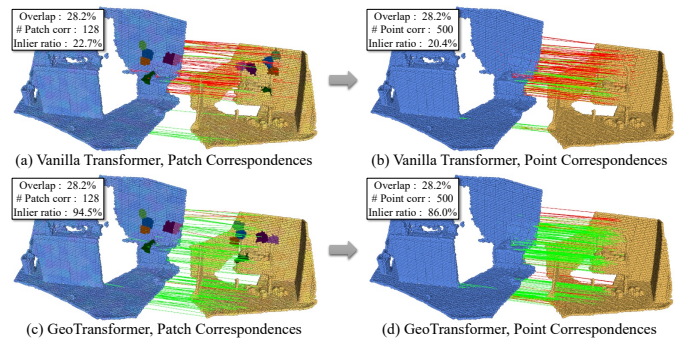


Fig. 1. Given two low-overlap point clouds, GeoTransformer improves inlier ratio over vanilla transformer significantly, both for superpoint (patch) level (left) and for dense point level (right). A few representative patch correspondences are visualized with distinct colors. Notice how GeoTransformer preserves the spatial consistency of the matching patches across two point clouds. It corrects the wrongly matched patches around the symmetric corners of the chair back (see the yellow point cloud).

while, patch overlapping is a more reliable and informative constraint than distance-based point matching for learning correspondence; consider that two spatially close points could be geodesically distant. On the other hand, superpoint matching calls for features capturing more global context.

To this end, Transformer [11] has been adopted [6], [12] to encode contextual information in point cloud registration. However, vanilla transformer overlooks the geometric structure of the point clouds, which makes the learned features geometrically less discriminative and induces numerous outlier matches (Fig. 1(top)). Although one can inject positional embeddings [13], [14], the coordinate-based encoding is transformation-variant, which is problematic when registering point clouds given in arbitrary poses. We advocate that a point transformer for registration task should be

- Z. Qin, C. Wang, Y. Guo, Y. Peng, D. Hu and K. Xu are with National University of Defense Technology, China. Y. Guo is also with Sun Yat-sen University, China. H. Yu and S. Ilic are with Technical University of Munich, Germany. S. Ilic is also with Siemens AG, Germany.
- Corresponding authors: Dewen Hu (dwhu@nudt.edu.cn), Kai Xu (kevin.kai.xu@gmail.com).

Manuscript received April 19, 2005; revised August 26, 2015.

learned with the *geometric structure* of the point clouds so as to extract transformation-invariant geometric features. We propose *Geometric Transformer*, or *GeoTransformer* for short, for 3D point clouds which encodes only distances of point pairs and angles in point triplets.

Given a superpoint, we learn a non-local representation through geometrically “pinpointing” it w.r.t. all other superpoints based on pair-wise distances and triplet-wise angles. Self-attention mechanism is utilized to weigh the importance of those anchoring superpoints. Since distances and angles are invariant to rigid transformation, GeoTransformer learns geometric structure of point clouds efficiently, leading to highly robust superpoint matching even in low-overlap scenarios. Fig. 1(left) demonstrates that GeoTransformer significantly improves the inlier ratio of superpoint (patch) correspondences. For better convergence, we devise an overlap-aware circle loss to make GeoTransformer focus on superpoint pairs with higher patch overlap.

Benefitting from the high-quality superpoint matches, our method attains high-inlier-ratio dense point correspondences (Fig. 1(right)) using an optimal transport layer [15], as well as highly robust and accurate registration without relying on RANSAC. Therefore, the registration part of our method runs extremely fast, *e.g.*, 0.01s for two point clouds with 5K correspondences, 100 times faster than RANSAC. Extensive experiments on indoor, outdoor, synthetic, multi-way and non-rigid benchmarks [5], [16], [17], [18], [19] have demonstrated the efficacy of GeoTransformer. Specifically, our method attains significant improvements on challenging scenarios with low overlap and large rotations. For example, our method improves the inlier ratio by 18~31 percentage points and the registration recall by over 7 points on the 3DLoMatch benchmark [5]. Our main contributions are:

- A fast and accurate point cloud registration method which is both keypoint-free and RANSAC-free.
- A geometric transformer architecture which learns transformation-invariant geometric representation of point clouds for robust superpoint matching.
- An overlap-aware circle loss which reweights the loss of each superpoint match according to the patch overlap ratio for better convergence.

A previous version of this work was published at CVPR 2022 [20]. This paper extends the conference version with the following new contributions: First, to reduce the memory footprint and the computational cost of GeoTransformer, we propose *shared geometric self-attention* which makes the attention weights for the geometric structure embeddings shared across all self-attention modules. Second, we extend our method to deal with non-rigid registration through relaxing the selection of superpoint correspondences, demonstrating the strong generality of GeoTransformer. Third, we further conduct more extensive experiments and detailed ablation analysis to provide a thorough understanding of the effectiveness of GeoTransformer.

2 RELATED WORK

Correspondence-based methods. Our work follows the line of the correspondence-based methods [1], [2], [3], [21]. They first extract correspondences between two point

clouds and then recover the transformation with robust pose estimators, *e.g.*, RANSAC. Thanks to the robust estimators, they achieve state-of-the-art performance in indoor and outdoor scene registration. These methods can be further categorized into two classes according to how they extract correspondences. The first class aims to detect more repeatable keypoints [4], [5] and learn more powerful descriptors for the keypoints [3], [7], [22]. While the second class [6] retrieves correspondences without keypoint detection by considering all possible matches. Our method follows the detection-free methods and improves the accuracy of correspondences by leveraging the geometric information.

Direct registration methods. Recently, direct registration methods have emerged. They estimate the transformation with a neural network in an end-to-end manner and eliminate the use of a robust estimator. According to how the alignment transformation is estimated, these methods can be further classified into two classes. The first class [12], [23], [24], [25], [26], [27], [28] follows the idea of ICP [29], which iteratively establishes soft correspondences and computes the transformation with differentiable weighted SVD. And the second class [30], [31], [32], [33] first extracts a global feature vector for each point cloud and regresses the transformation with the global feature vectors. Due to the lack of a robust estimator, direct registration methods opt to adopt an iterative registration scheme to progressively refine the estimated transformation. Albeit achieving promising results on single synthetic shapes, direct registration methods could still fail in large-scale scenes as stated in [5].

Deep robust estimators. As traditional robust estimators such as RANSAC suffer from slow convergence and instability in case of high outlier ratio, deep robust estimators [34], [35], [36] have been proposed as the alternatives for them. They usually contain a classification network to reject outliers and an estimation network to compute the transformation. Compared with traditional robust estimators, they achieve improvements in both accuracy and speed. However, they require training a specific network. In comparison, our method achieves fast and accurate registration with a parameter-free local-to-global registration scheme.

Geometric consistency in point cloud registration. Geometric consistency has been an important and long-standing research topic in point cloud registration. Given two point clouds in arbitrary poses, certain geometric properties such as distances and angles are preserved between them, which provides a strong geometric guidance for registration. To this end, previous hand-crafted methods [37], [38], [39] directly encode lengths and angles around an anchor point to obtain transformation-invariant descriptors. However, these descriptors are not aware of the global structure, which restricts their distinctiveness. Besides, geometric consistency has also been adopted to reject outlier correspondences such that more accurate transformation could be recovered [36], [40], [41]. These methods need a preceding correspondence extractor and are orthogonal to this work.

3 METHOD

Given two point clouds $\mathcal{P} = \{\mathbf{p}_i \in \mathbb{R}^3 \mid i = 1, \dots, N\}$ and $\mathcal{Q} = \{\mathbf{q}_i \in \mathbb{R}^3 \mid i = 1, \dots, M\}$, our goal is to estimate a rigid transformation $\mathbf{T} = \{\mathbf{R}, \mathbf{t}\}$ which aligns the two point

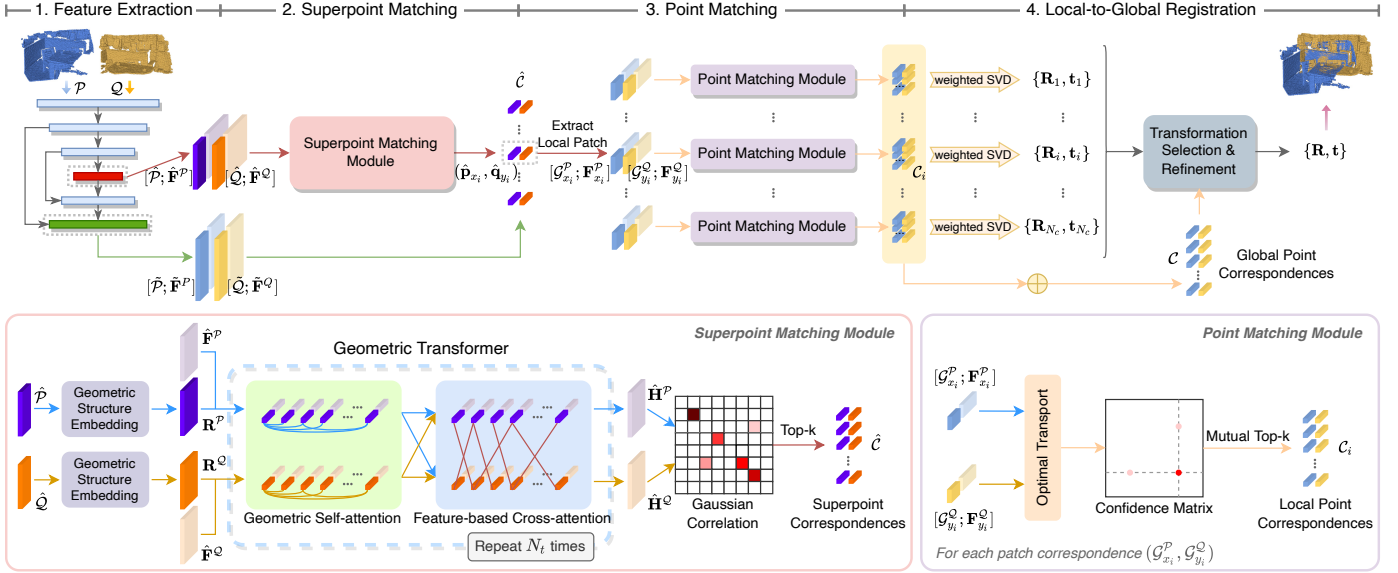


Fig. 2. **The overall pipeline of our method.** The backbone downsamples the input point clouds and learns features in multiple resolution levels. The Superpoint Matching Module extracts high-quality superpoint correspondences between \hat{P} and \hat{Q} using the Geometric Transformer which iteratively encodes intra-point-cloud geometric structures and inter-point-cloud geometric consistency. The superpoint correspondences are then propagated to dense points \tilde{P} and \tilde{Q} by the Point Matching Module. Finally, the transformation is computed with a local-to-global registration method.

clouds, with a 3D rotation $\mathbf{R} \in SO(3)$ and a 3D translation $\mathbf{t} \in \mathbb{R}^3$. The transformation can be solved by:

$$\min_{\mathbf{R}, \mathbf{t}} \sum_{(\mathbf{p}_{x_i}^*, \mathbf{q}_{y_i}^*) \in \mathcal{C}^*} \|\mathbf{R} \cdot \mathbf{p}_{x_i}^* + \mathbf{t} - \mathbf{q}_{y_i}^*\|_2^2. \quad (1)$$

Here \mathcal{C}^* is the set of ground-truth correspondences between P and Q . Since \mathcal{C}^* is unknown in reality, we need to first establish point correspondences between two point clouds and then estimate the alignment transformation.

Our method adopts the hierarchical correspondence paradigm which finds correspondences in a coarse-to-fine manner. We adopt KPConv-FPN to simultaneously down-sample the input point clouds and extract point-wise features (Sec. 3.1). The first and the last (coarsest) level down-sampled points correspond to the dense points and the superpoints to be matched. A *Superpoint Matching Module* is used to extract superpoint correspondences whose neighboring local patches overlap with each other (Sec. 3.2). Based on that, a *Point Matching Module* then refines the superpoint correspondences to dense points (Sec. 3.3). At last, the alignment transformation is recovered from the dense correspondences without relying on RANSAC (Sec. 3.4). The pipeline is illustrated in Fig. 2.

3.1 Superpoint Sampling and Feature Extraction

We utilize the KPConv-FPN backbone [42], [43] to extract multi-level features for the point clouds. A byproduct of the point feature learning is point downsampling. We work on downsampled points since point cloud registration can actually be pinned down by the correspondences of a much coarser subset of points. The original point clouds are usually too dense so that point-wise correspondences are redundant and sometimes too clustered to be useful.

The points correspond to the coarsest resolution, denoted by \hat{P} and \hat{Q} , are treated as *superpoints* to be matched. The associated learned features are denoted as $\hat{\mathbf{F}}^P \in \mathbb{R}^{|\hat{P}| \times \hat{d}}$

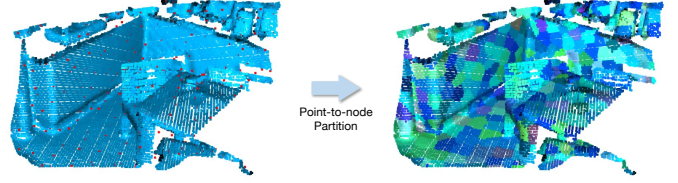


Fig. 3. **Point-to-node grouping strategy.** Each point is assigned to its nearest superpoint. Left: the point cloud (in blue) and the sampled superpoints (in red). Right: the points are color-coded according to the superpoints that they are assigned to.

and $\hat{\mathbf{F}}^Q \in \mathbb{R}^{|\hat{Q}| \times \hat{d}}$. The dense point correspondences are computed at 1/2 of the original resolution, *i.e.*, the first level downsampled points denoted by \tilde{P} and \tilde{Q} . Their learned features are represented by $\tilde{\mathbf{F}}^P \in \mathbb{R}^{|\tilde{P}| \times \tilde{d}}$ and $\tilde{\mathbf{F}}^Q \in \mathbb{R}^{|\tilde{Q}| \times \tilde{d}}$.

For each superpoint, we construct a local *patch* of points around it using the point-to-node grouping strategy [6], [44]. As shown in Fig. 3, each point in \tilde{P} and its features from $\tilde{\mathbf{F}}^P$ are assigned to its nearest superpoint in the geometric space:

$$\mathcal{G}_i^P = \{\tilde{\mathbf{p}} \in \tilde{P} \mid i = \operatorname{argmin}_j (\|\tilde{\mathbf{p}} - \hat{\mathbf{p}}_j\|_2), \hat{\mathbf{p}}_j \in \hat{P}\}. \quad (2)$$

This essentially leads to a Voronoi decomposition of the input point cloud seeded by superpoints. The feature matrix associated with the points in \mathcal{G}_i^P is denoted as $\mathbf{F}_i^P \subset \tilde{\mathbf{F}}^P$. The superpoints with an empty patch are removed. The patches $\{\mathcal{G}_i^Q\}$ and the feature matrices $\{\mathbf{F}_i^Q\}$ for Q are computed and denoted in a similar way.

3.2 Superpoint Matching Module

Geometric Transformer. Global context has proven critical in many computer vision tasks [6], [10], [45]. For this reason, transformer has been adopted to leverage global contextual information for point cloud registration. However, existing methods [5], [6], [12] usually feed transformer with

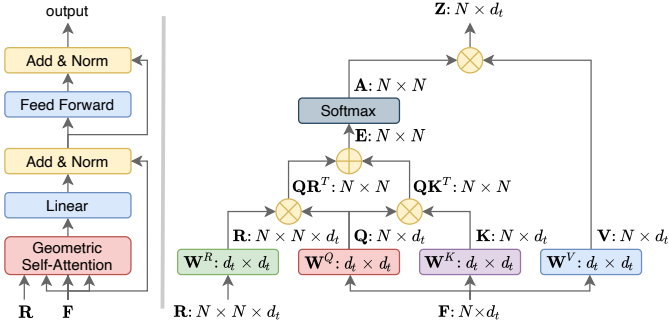


Fig. 4. **Geometric self-attention module.** Left: The structure of geometric self-attention module. Right: The computation graph of geometric self-attention mechanism.

only high-level point cloud features and does not explicitly encode the geometric structure. This makes the learned features geometrically less discriminative, which causes severe matching ambiguity and numerous outlier matches, especially in low-overlap cases. A straightforward recipe is to explicitly inject positional embeddings [13], [14] of 3D point coordinates. However, the resultant coordinate-based transformers are naturally *transformation-variant*, while registration requires *transformation invariance* since the input point clouds can be in arbitrary poses.

To this end, we propose *Geometric Transformer* which not only encodes high-level point features but also explicitly captures intra-point-cloud geometric structures and inter-point-cloud geometric consistency. GeoTransformer is composed of a *geometric self-attention* module for learning intra-point-cloud features and a *feature-based cross-attention* module for modeling inter-point-cloud consistency. The two modules are interleaved for N_t times to extract hybrid features $\hat{\mathbf{H}}^P$ and $\hat{\mathbf{H}}^Q$ for reliable superpoint matching (see Fig. 2 (bottom left)).

Geometric self-attention. We design a *geometric self-attention* to learn the global correlations in both feature and geometric spaces among the superpoints for each point cloud. In the following, we describe the computation for \hat{P} and the same goes for \hat{Q} . Given the input feature matrix $\mathbf{X} \in \mathbb{R}^{|\hat{P}| \times d_t}$, the output feature matrix $\mathbf{Z} \in \mathbb{R}^{|\hat{P}| \times d_t}$ is the weighted sum of all projected input features:

$$\mathbf{z}_i = \sum_{j=1}^{|\hat{P}|} a_{i,j} (\mathbf{x}_j \mathbf{W}^V), \quad (3)$$

where the weight coefficient $a_{i,j}$ is computed by a row-wise softmax on the attention score $e_{i,j}$, and $e_{i,j}$ is computed as:

$$e_{i,j} = \frac{(\mathbf{x}_i \mathbf{W}^Q)(\mathbf{x}_j \mathbf{W}^K + \mathbf{r}_{i,j} \mathbf{W}^R)^T}{\sqrt{d_t}}. \quad (4)$$

Here, $\mathbf{r}_{i,j} \in \mathbb{R}^{d_t}$ is a *geometric structure embedding* to be described in the next. $\mathbf{W}^Q, \mathbf{W}^K, \mathbf{W}^V, \mathbf{W}^R \in \mathbb{R}^{d_t \times d_t}$ are the respective projection matrices for queries, keys, values and geometric structure embeddings. Fig. 4 shows the structure and the computation of geometric self-attention.

We design a novel *geometric structure embedding* to encode the transformation-invariant geometric structure of the superpoints. The core idea is to leverage the distances and

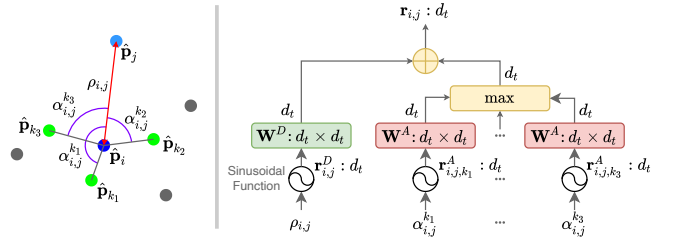


Fig. 5. **Geometric structure embedding.** Left: An illustration of the pair-wise distance and the triplet-wise angles encoded. Right: The computation graph of the geometric structure embedding.

angles computed with the superpoints which are consistent across different point clouds of the same scene. Given two superpoints $\hat{\mathbf{p}}_i, \hat{\mathbf{p}}_j \in \hat{P}$, their geometric structure embedding consists of a *pair-wise distance embedding* and a *triplet-wise angular embedding*, which will be described below.

(1) *Pair-wise Distance Embedding.* Given the distance $\rho_{i,j} = \|\hat{\mathbf{p}}_i - \hat{\mathbf{p}}_j\|_2$ between $\hat{\mathbf{p}}_i$ and $\hat{\mathbf{p}}_j$, the distance embedding $\mathbf{r}_{i,j}^D \in \mathbb{R}^{d_t}$ between them is computed by applying a sinusoidal function [11] on $\rho_{i,j}$:

$$\begin{cases} r_{i,j,2k}^D = \sin\left(\frac{d_{i,j}/\sigma_d}{10000^{2k/d_t}}\right) \\ r_{i,j,2k+1}^D = \cos\left(\frac{d_{i,j}/\sigma_d}{10000^{2k/d_t}}\right) \end{cases}, \quad (5)$$

where σ_d is a temperature hyper-parameter used to tune the sensitivity on distance variations.

(2) *Triplet-wise Angular Embedding.* We compute angular embedding with triplets of superpoints. We first select the k nearest neighbors \mathcal{K}_i of $\hat{\mathbf{p}}_i$. For each $\hat{\mathbf{p}}_x \in \mathcal{K}_i$, we compute the angle $\alpha_{i,j}^x = \angle(\Delta_{x,i}, \Delta_{j,i})$, where $\Delta_{i,j} := \hat{\mathbf{p}}_i - \hat{\mathbf{p}}_j$. The triplet-wise angular embedding $\mathbf{r}_{i,j,x}^A \in \mathbb{R}^{d_t}$ is then computed with a sinusoidal function on $\alpha_{i,j}^x$:

$$\begin{cases} r_{i,j,x,2l}^A = \sin\left(\frac{\alpha_{i,j}^x/\sigma_a}{10000^{2l/d_t}}\right) \\ r_{i,j,x,2l+1}^A = \cos\left(\frac{\alpha_{i,j}^x/\sigma_a}{10000^{2l/d_t}}\right) \end{cases}, \quad (6)$$

where σ_a controls the sensitivity on angular variations.

Finally, the geometric structure embedding $\mathbf{r}_{i,j}$ is computed by aggregating the pair-wise distance embedding and the triplet-wise angular embedding:

$$\mathbf{r}_{i,j} = \mathbf{r}_{i,j}^D \mathbf{W}^D + \max_x \left\{ \mathbf{r}_{i,j,x}^A \mathbf{W}^A \right\}, \quad (7)$$

where $\mathbf{W}^D, \mathbf{W}^A \in \mathbb{R}^{d_t \times d_t}$ are the respective projection matrices for the two types of embeddings. We use max pooling here to improve the robustness to the varying nearest neighbors of a superpoint due to self-occlusion. Fig. 5 illustrates the computation of geometric structure embedding.

Shared geometric self-attention. Albeit enjoying a strong representation capability, the geometric self-attention suffers from the heavy computation of the embedding projection $\mathbf{r}_{i,j} \mathbf{W}^R$ in Eq. (4). The computational complexity of the standard geometric self-attention is $O(|\hat{P}|d_t^2 + |\hat{P}|^2d_t^2)$, which limits its scalability and efficiency especially when the number of superpoints is large. To reduce the computation, we design a *shared geometric self-attention* which makes the

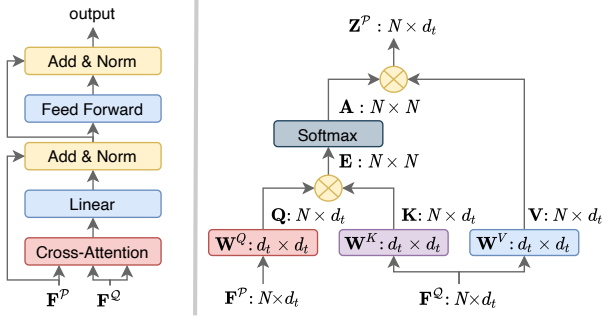


Fig. 6. **Feature-based cross-attention module.** Left: The structure of feature-based cross-attention module. Right: The computation graph of cross-attention mechanism.

projection weights \mathbf{W}^R shared across all geometric self-attention modules and apply \mathbf{W}^R in Eq. (7) instead:

$$e_{i,j} = \frac{(\mathbf{x}_i \mathbf{W}^Q)(\mathbf{x}_j \mathbf{W}^K + \mathbf{r}_{i,j})}{\sqrt{d_t}}. \quad (8)$$

The geometric structure embedding is then computed as

$$\mathbf{r}_{i,j} = \sigma(\mathbf{r}_{i,j}^D \mathbf{W}^D + \max_x \{ \mathbf{r}_{i,j,x}^A \mathbf{W}^A \}) \mathbf{W}^R, \quad (9)$$

where σ is the LeakyReLU function. With this modification, the computation complexity of geometric self-attention is reduced to $O(|\hat{\mathcal{P}}|d_t^2)$. As shown in Sec. 4.2, shared geometric self-attention attains comparable accuracy with the standard version with a significant reduction of computation time.

Feature-based cross-attention. Cross-attention is a typical module for point cloud registration task [5], [6], [12], used to perform feature exchange between two input point clouds. Given the self-attention feature matrices \mathbf{X}^P , \mathbf{X}^Q for $\hat{\mathcal{P}}$, $\hat{\mathcal{Q}}$ respectively, the cross-attention feature matrix \mathbf{Z}^P of $\hat{\mathcal{P}}$ is computed with the features of $\hat{\mathcal{Q}}$:

$$\mathbf{z}_i^P = \sum_{j=1}^{|\hat{\mathcal{Q}}|} a_{i,j} (\mathbf{x}_j^Q \mathbf{W}^V). \quad (10)$$

Similarly, $a_{i,j}$ is computed by a row-wise softmax on the cross-attention score $e_{i,j}$, and $e_{i,j}$ is computed as the feature correlation between the \mathbf{X}^P and \mathbf{X}^Q :

$$e_{i,j} = \frac{(\mathbf{x}_i^P \mathbf{W}^Q)(\mathbf{x}_j^Q \mathbf{W}^K)^T}{\sqrt{d_t}}. \quad (11)$$

Fig. 6 shows the structure and the computation of the cross-attention. The cross-attention features for \mathcal{Q} are computed in the same way. While the geometric self-attention module encodes the transformation-invariant geometric structure for each individual point cloud, the feature-based cross-attention module can model the geometric consistency across the two point clouds. The resultant hybrid features are both invariant to transformation and robust for reasoning correspondence.

Superpoint matching. To find the superpoint correspondences, we propose a matching scheme based on global feature correlation. We first normalize $\hat{\mathbf{H}}^P$ and $\hat{\mathbf{H}}^Q$ onto a unit hypersphere and compute a Gaussian correlation matrix $\mathbf{S} \in \mathbb{R}^{|\hat{\mathcal{P}}| \times |\hat{\mathcal{Q}}|}$ with $s_{i,j} = \exp(-\|\hat{\mathbf{h}}_i^P - \hat{\mathbf{h}}_j^Q\|_2^2)$. In practice, some patches of a point cloud are less geometrically

discriminative and have numerous similar patches in the other point cloud. Besides our powerful hybrid features, we also perform a dual-normalization operation [8], [10] on \mathbf{S} to further suppress ambiguous matches, leading to $\bar{\mathbf{S}}$ with

$$\bar{s}_{i,j} = \frac{s_{i,j}}{\sum_{k=1}^{|\hat{\mathcal{Q}}|} s_{i,k}} \cdot \frac{s_{i,j}}{\sum_{k=1}^{|\hat{\mathcal{P}}|} s_{k,j}}. \quad (12)$$

We found that this suppression can effectively eliminate wrong matches. Finally, we select the largest N_c entries in $\bar{\mathbf{S}}$ as the *superpoint correspondences*:

$$\hat{\mathcal{C}} = \{(\hat{\mathbf{p}}_{x_i}, \hat{\mathbf{q}}_{y_i}) \mid (x_i, y_i) \in \text{topk}_{x,y}(\bar{s}_{x,y})\}. \quad (13)$$

Due to the powerful geometric structure encoding of GeoTransformer, our method is able to achieve accurate registration in low-overlap cases and with less point correspondences, and most notably, in a RANSAC-free manner.

3.3 Point Matching Module

Having obtained the superpoint correspondences, we extract point correspondences using a simple yet effective *Point Matching Module*. At point level, we use only local point features learned by the backbone. The rationale is that point level matching is mainly determined by the vicinities of the two points being matched, once the global ambiguity has been resolved by superpoint matching. This design choice improves the robustness.

For each superpoint correspondence $\hat{\mathcal{C}}_i = (\hat{\mathbf{p}}_{x_i}, \hat{\mathbf{q}}_{y_i})$, an optimal transport layer [15] is used to extract the *local* dense point correspondences between $\mathcal{G}_{x_i}^P$ and $\mathcal{G}_{y_i}^Q$. Specifically, we first compute a cost matrix $\mathbf{C}_i \in \mathbb{R}^{n_i \times m_i}$:

$$\mathbf{C}_i = \mathbf{F}_{x_i}^P (\mathbf{F}_{y_i}^Q)^T / \sqrt{d}, \quad (14)$$

where $n_i = |\mathcal{G}_{x_i}^P|$, $m_i = |\mathcal{G}_{y_i}^Q|$. The cost matrix \mathbf{C}_i is then augmented into $\bar{\mathbf{C}}_i$ by appending a new row and a new column as in [15], filled with a learnable dustbin parameter α . We then utilize the Sinkhorn algorithm [46] on $\bar{\mathbf{C}}_i$ to compute a soft assignment matrix $\bar{\mathbf{Z}}_i$ which is then recovered to \mathbf{Z}_i by dropping the last row and the last column. We use \mathbf{Z}_i as the confidence matrix of the candidate matches and extract point correspondences via mutual top- k selection, where a point match is selected if it is among the k largest entries of both the row and the column that it resides in:

$$\mathcal{C}_i = \{(\mathcal{G}_{x_i}^P(x_j), \mathcal{G}_{y_i}^Q(y_j)) \mid (x_j, y_j) \in \text{mutual_topk}_{x,y}(z_{x,y}^i)\}. \quad (15)$$

The point correspondences computed from each superpoint match are then collected together to form the final *global* dense point correspondences: $\mathcal{C} = \bigcup_{i=1}^{N_c} \mathcal{C}_i$.

3.4 RANSAC-free Local-to-Global Registration

Previous methods generally rely on robust pose estimators to estimate the transformation since the putative correspondences are often predominated by outliers. Most robust estimators such as RANSAC suffer from slow convergence. Given the high inlier ratio of GeoTransformer, we are able to achieve robust registration without relying on robust estimators, which also greatly reduces computation cost.

We design a *local-to-global registration* (LGR) scheme. As a hypothesize-and-verify approach, LGR is comprised of

Algorithm 1: Local-to-Global Registration

Input: C_i : local point correspondences of superpoint correspondences
Output: \mathbf{R}, \mathbf{t} : alignment transformation
1. *local step*
for $i \leftarrow 1, \dots, N_c$ **do**
| Compute $\mathbf{R}_i, \mathbf{t}_i$ by solving Eq. (16) on C_i .
end
2. *global step*
Select best transformation candidate \mathbf{R}, \mathbf{t} by Eq. (17).
 $\mathcal{C} \leftarrow C_1 \cup \dots \cup C_{N_c}$
for $t \leftarrow 1, \dots, N_r$ **do**
| ${}^{(t)}\mathcal{C} \leftarrow$ inliers in \mathcal{C} under \mathbf{R} and \mathbf{t} .
| Update \mathbf{R}, \mathbf{t} by solving Eq. (16) on ${}^{(t)}\mathcal{C}$.
end

a local phase of transformation candidates generation and a global phase for transformation selection. In the local phase, we solve for a transformation $\mathbf{T}_i = \{\mathbf{R}_i, \mathbf{t}_i\}$ for each superpoint match using its *local point correspondences*:

$$\mathbf{R}_i, \mathbf{t}_i = \min_{\mathbf{R}, \mathbf{t}} \sum_{(\tilde{\mathbf{p}}_{x_j}, \tilde{\mathbf{q}}_{y_j}) \in C_i} w_j^i \|\mathbf{R} \cdot \tilde{\mathbf{p}}_{x_j} + \mathbf{t} - \tilde{\mathbf{q}}_{y_j}\|_2^2. \quad (16)$$

This can be solved in closed form using weighted SVD [29]. The corresponding confidence score for each correspondence in \mathbf{Z}_i is used as the weight w_j^i . Benefitting from the high-quality correspondences, the transformations obtained in this phase are already very accurate. In the global phase, we select the transformation which admits the most inlier matches over the entire *global point correspondences*:

$$\mathbf{R}, \mathbf{t} = \max_{\mathbf{R}, \mathbf{t}} \sum_{(\tilde{\mathbf{p}}_{x_j}, \tilde{\mathbf{q}}_{y_j}) \in \mathcal{C}} \mathbb{I}[\|\mathbf{R}_i \cdot \tilde{\mathbf{p}}_{x_j} + \mathbf{t}_i - \tilde{\mathbf{q}}_{y_j}\|_2 < \tau_a], \quad (17)$$

where $\mathbb{I}[\cdot]$ is the Iverson bracket. τ_a is the acceptance radius. We then iteratively re-estimate the transformation with the surviving inlier matches for N_r times by solving Eq. (16). Alg. 1 shows the computation of the local-to-global registration. As shown in Sec. 4.2, our approach achieves comparable registration accuracy with RANSAC but reduces the computation time by more than 100 times. Moreover, unlike deep robust estimators [34], [35], [36], our method is parameter-free and no network training is needed.

3.5 Loss Functions

The loss function $\mathcal{L} = \mathcal{L}_{oc} + \mathcal{L}_p$ is composed of an *overlap-aware circle loss* \mathcal{L}_{oc} for superpoint matching and a *point matching loss* \mathcal{L}_p for point matching.

Overlap-aware circle loss. Existing methods [6], [10] usually formulate superpoint matching as a multi-label classification problem and adopt a cross-entropy loss with dual-softmax [10] or optimal transport [6], [15]. Each superpoint is assigned (classified) to one or many of the other superpoints, where the ground truth is computed based on patch overlap and it is very likely that one patch could overlap with multiple patches. By analyzing the gradients from the cross-entropy loss, we find that the positive classes with high confidence scores are suppressed by positive gradients in the multi-label classification. This hinders the model from extracting reliable superpoint correspondences.

To address this issue, we opt to extract superpoint descriptors in a metric learning fashion. A straightforward

solution is to adopt a circle loss [47] similar to [4], [5]. However, the circle loss overlooks the differences between the positive samples and weights them equally. As a result, it struggles in matching patches with relatively low overlap. For this reason, we design an *overlap-aware circle loss* to focus the model on those matches with high overlap. We select the patches in \mathcal{P} which have at least one positive patch in \mathcal{Q} to form a set of anchor patches, \mathcal{A} . A pair of patches are positive if they share at least 10% overlap, and negative if they do not overlap. All other pairs are omitted. For each anchor patch $\mathcal{G}_i^{\mathcal{P}} \in \mathcal{A}$, we denote the set of its positive patches in \mathcal{Q} as ε_p^i , and the set of its negative patches as ε_n^i . The overlap-aware circle loss on \mathcal{P} is then defined as:

$$\mathcal{L}_{oc}^{\mathcal{P}} = \frac{1}{|\mathcal{A}|} \sum_{\mathcal{G}_i^{\mathcal{P}} \in \mathcal{A}} \log[1 + \sum_{\mathcal{G}_j^{\mathcal{Q}} \in \varepsilon_p^i} e^{\lambda_j^i \beta_p^{i,j} (d_j^i - \Delta_p)} \cdot \sum_{\mathcal{G}_k^{\mathcal{Q}} \in \varepsilon_n^i} e^{\beta_n^{i,k} (\Delta_n - d_k^i)}], \quad (18)$$

where $d_j^i = \|\hat{\mathbf{h}}_i^{\mathcal{P}} - \hat{\mathbf{h}}_j^{\mathcal{Q}}\|_2$ is the distance in the feature space, $\lambda_j^i = (\sigma_j^i)^{\frac{1}{2}}$ and σ_j^i represents the overlap ratio between $\mathcal{G}_i^{\mathcal{P}}$ and $\mathcal{G}_j^{\mathcal{Q}}$. The positive and negative weights are computed for each sample individually with $\beta_p^{i,j} = \gamma(d_j^i - \Delta_p)$ and $\beta_n^{i,k} = \gamma(\Delta_n - d_k^i)$. The margin hyper-parameters are set to $\Delta_p = 0.1$ and $\Delta_n = 1.4$. The overlap-aware circle loss reweights the loss values on ε_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 $\mathcal{L}_{oc}^{\mathcal{Q}}$ on \mathcal{Q} . And the overall loss is $\mathcal{L}_{oc} = (\mathcal{L}_{oc}^{\mathcal{P}} + \mathcal{L}_{oc}^{\mathcal{Q}})/2$.

Point matching loss. The ground-truth point correspondences are relatively sparse because they are available only for downsampled point clouds. We simply use a negative log-likelihood loss [15] on the assignment matrix $\hat{\mathbf{Z}}_i$ of each superpoint correspondence. During training, we randomly sample N_g ground-truth superpoint correspondences $\{\hat{C}_i^*\}$ instead of using the predicted ones. For each \hat{C}_i^* , a set of ground-truth point correspondences \mathcal{M}_i is extracted with a matching radius τ . The sets of unmatched points in the two patches are denoted as \mathcal{I}_i and \mathcal{J}_i . The individual point matching loss for \hat{C}_i^* is computed as:

$$\mathcal{L}_{p,i} = - \sum_{(x,y) \in \mathcal{M}_i} \log \bar{z}_{x,y}^i - \sum_{x \in \mathcal{I}_i} \log \bar{z}_{x,m_i+1}^i - \sum_{y \in \mathcal{J}_i} \log \bar{z}_{n_i+1,y}^i, \quad (19)$$

The final loss is computed by averaging the individual loss over all sampled superpoint matches: $\mathcal{L}_p = \frac{1}{N_g} \sum_{i=1}^{N_g} \mathcal{L}_{p,i}$.

4 EXPERIMENTS

In this section, we conduct extensive experiments to evaluate the effectiveness of our GeoTransformer. We first introduce the implementation details in the experiments in Sec. 4.1. Then, we evaluate our method and compare with previous state-of-the-art methods on indoor 3DMatch and 3DLoMatch benchmarks [5], [16] (Sec. 4.2), outdoor KITTI odometry benchmark [17] (Sec. 4.3), synthetic ModelNet40 benchmark [48] (Sec. 4.4), and multiway Augmented ICL-NUIM benchmark [18] (Sec. 4.5). We further investigate the generality of GeoTransformer to non-rigid registration [19] (Sec. 4.6). Next, the ablation study is shown in Sec. 4.7 to provide a comprehensive understanding of our design. At last, we compare our method with recent deep robust estimators in Sec. 4.8.

4.1 Implementation Details

Network architecture. As the point clouds from different benchmarks differ in density and size, we use slightly different backbones in the experiments. To be specific, we use a 4-stage backbone for 3DMatch, ModelNet40 and 4DMatch, while a 5-stage backbone is used for KITTI due to the much larger point clouds. Please refer to our code for more details.

In the superpoint matching module, we interleave the geometric self-attention module and the feature-based cross-attention module for $N_t=3$ times on all benchmarks. All attention modules have 4 attention heads. To compute the geometric structure embedding, we simply set σ_d to the voxel size in the superpoint level for the pair-wise distance embedding, and use $\sigma_a=15^\circ$ and $k=3$ for the triplet-wise angular embedding. We study the influence of these hyper-parameters in Sec. 4.7.

In the local-to-global registration, only the superpoint matches with at least 3 local point correspondences are used to compute the transformation candidates. At last, we iteratively recompute the transformation with the surviving inlier matches for $N_r=5$ times.

Training and testing. We implement and evaluate GeoTransformer with PyTorch [49] on a RTX 3090 GPU. The models are trained with Adam optimizer [50] for 40 epochs on 3DMatch/4DMatch, 200 epochs on ModelNet40 and 80 epochs on KITTI. The batch size is 1 and the weight decay is 10^{-6} . The learning rate starts from 10^{-4} and decays exponentially by 0.05 every epoch on 3DMatch/4DMatch, every 5 epochs on ModelNet40, and every 4 epochs on KITTI. The same data augmentation as in [5] is adopted. Unless otherwise noted, we randomly sample $N_g=128$ ground-truth superpoint correspondences during training, and use $N_c=256$ putative superpoint matches during testing.

4.2 Indoor Benchmark: 3DMatch & 3DLoMatch

Dataset. 3DMatch [16] contains 62 scenes among which 46 are used for training, 8 for validation and 8 for testing. We use the training data preprocessed by [5] and evaluate on both 3DMatch and 3DLoMatch [5] protocols. The point cloud pairs in 3DMatch have $> 30\%$ overlap, while those in 3DLoMatch have low overlap of $10\% \sim 30\%$.

Metrics. Following [4], [5], we evaluate the performance with three metrics: (1) *Inlier Ratio* (IR), the fraction of putative correspondences whose residuals are below a certain threshold (*i.e.*, 0.1m) under the ground-truth transformation, (2) *Feature Matching Recall* (FMR), the fraction of point cloud pairs whose inlier ratio is above a certain threshold (*i.e.*, 5%), and (3) *Registration Recall* (RR), the fraction of point cloud pairs whose transformation error is smaller than a certain threshold (*i.e.*, RMSE $< 0.2m$).

Correspondence results. We first compare the correspondence results of our method with the recent state of the arts: PerfectMatch [2], FCGF [3], D3Feat [4], SpinNet [7], Predator [5], YOHO [22] and CoFiNet [6] in Tab. 1(top and middle)¹. Following [4], [5], we report the results with different numbers of correspondences. To control the number of the correspondences, we vary the hyper-parameter k of

1. We refine our code and retrain the models, so the results are slightly better than our conference version [20].

TABLE 1

Evaluation results on 3DMatch and 3DLoMatch. RANSAC is used for registration with 50K iterations. † indicates the lite model with shared geometric self-attention. **Boldfaced** numbers highlight the best and the second best are underlined.

# Samples	3DMatch					3DLoMatch				
	5000	2500	1000	500	250	5000	2500	1000	500	250
<i>Feature Matching Recall (%)</i> †										
PerfectMatch [2]	95.0	94.3	92.9	90.1	82.9	63.6	61.7	53.6	45.2	34.2
FCGF [3]	97.4	97.3	97.0	96.7	96.6	76.6	75.4	74.2	71.7	67.3
D3Feat [4]	95.6	95.4	94.5	94.1	93.1	67.3	66.7	67.0	66.7	66.5
SpinNet [7]	97.6	97.2	96.8	95.5	94.3	75.3	74.9	72.5	70.0	63.6
Predator [5]	96.6	96.6	96.5	96.3	96.5	78.6	77.4	76.3	75.7	75.3
YOHO [22]	98.2	97.6	<u>97.5</u>	97.7	96.0	79.4	78.1	76.3	73.8	69.1
CoFiNet [6]	<u>98.1</u>	98.3	98.1	98.2	98.3	83.1	83.5	83.3	83.1	82.6
GeoTransformer (ours)	<u>98.1</u>	<u>98.1</u>	98.1	98.2	<u>98.1</u>	<u>87.7</u>	<u>87.7</u>	<u>87.8</u>	<u>88.0</u>	<u>88.2</u>
GeoTransformer† (ours)	<u>98.1</u>	<u>98.1</u>	98.1	<u>98.1</u>	97.8	88.7	88.8	88.7	89.1	88.7
<i>Inlier Ratio (%)</i> †										
PerfectMatch [2]	36.0	32.5	26.4	21.5	16.4	11.4	10.1	8.0	6.4	4.8
FCGF [3]	56.8	54.1	48.7	42.5	34.1	21.4	20.0	17.2	14.8	11.6
D3Feat [4]	39.0	38.8	40.4	41.5	41.8	13.2	13.1	14.0	14.6	15.0
SpinNet [7]	47.5	44.7	39.4	33.9	27.6	20.5	19.0	16.3	13.8	11.1
Predator [5]	58.0	58.4	57.1	54.1	49.3	26.7	28.1	28.3	27.5	25.8
YOHO [22]	64.4	60.7	55.7	46.4	41.2	25.9	23.3	22.6	18.2	15.0
CoFiNet [6]	49.8	51.2	51.9	52.2	52.2	24.4	25.9	26.7	26.8	26.9
GeoTransformer (ours)	<u>72.5</u>	<u>75.9</u>	<u>76.8</u>	<u>82.8</u>	<u>85.6</u>	44.7	45.8	46.7	53.3	58.0
GeoTransformer† (ours)	<u>72.7</u>	76.1	76.9	82.9	85.7	<u>43.9</u>	45.9	46.7	53.6	58.3
<i>Registration Recall (%)</i> †										
PerfectMatch [2]	78.4	76.2	71.4	67.6	50.8	33.0	29.0	23.3	17.0	11.0
FCGF [3]	85.1	84.7	83.3	81.6	71.4	40.1	41.7	38.2	35.4	26.8
D3Feat [4]	81.6	84.5	83.4	82.4	77.9	37.2	42.7	46.9	43.8	39.1
SpinNet [7]	88.6	86.6	85.5	83.5	70.2	59.8	54.9	48.3	39.8	26.8
Predator [5]	89.0	89.9	90.6	88.5	86.6	59.8	61.2	62.4	60.8	58.1
YOHO [22]	90.8	90.3	89.1	88.6	84.5	65.2	65.5	63.2	56.5	48.0
CoFiNet [6]	89.3	88.9	88.4	87.4	87.0	67.5	66.2	64.2	63.1	61.0
GeoTransformer (ours)	92.3	92.1	92.0	91.7	91.2	75.4	75.0	74.6	74.0	73.9
GeoTransformer† (ours)	<u>92.2</u>	<u>92.0</u>	<u>91.6</u>	<u>91.5</u>	<u>91.1</u>	<u>74.9</u>	<u>74.5</u>	<u>73.9</u>	<u>73.6</u>	<u>73.0</u>

the mutual top- k selection in the point matching module and select the correspondences with the highest confidence scores. For *Feature Matching Recall*, our method achieves improvements of at least 5 percentage points (pp) on 3DLoMatch, demonstrating its effectiveness in low-overlap cases. For *Inlier Ratio*, the improvements are even more prominent. It surpasses the baselines consistently by 8~33 pp on 3DMatch and 18~31 pp on 3DLoMatch. The gain is larger with less correspondences. It implies that our method extracts more reliable correspondences.

Registration results. To evaluate the registration performance, we first compare the *Registration Recall* obtained by RANSAC in Tab. 1(bottom). Following [4], [5], we run 50K RANSAC iterations to estimate the transformation. GeoTransformer attains new state-of-the-art results on both 3DMatch and 3DLoMatch. It outperforms the previous best by 1.5 pp on 3DMatch and 7.9 pp on 3DLoMatch, showing its efficacy in both high- and low-overlap scenarios. And the shared geometric self-attention based model (*i.e.*, the *lite model*) attains very close performance to the standard one. More importantly, our method is quite stable under different numbers of samples, so it does not require sampling a large number of correspondences to boost the performance as previous methods [3], [6], [7], [22].

We then compare the registration results *without* using RANSAC in Tab. 2. We start with weighted SVD over correspondences in solving for alignment transformation. For the baselines, we first sample 5000 keypoints and generate

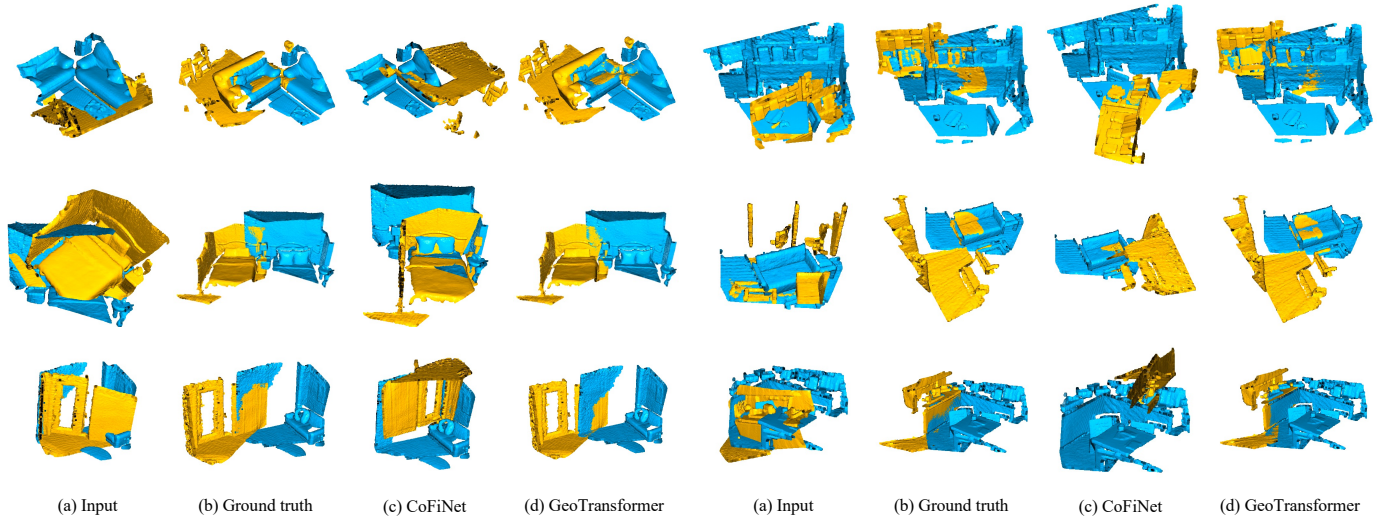


Fig. 7. **Comparison of the registration results on 3DLoMatch.** GeoTransformer can effectively recognize small overlapping area in complex scenes (see the first and third rows on the right) and distinguish similar objects at different positions (see the second and third rows on the left) thanks to the structure information from geometric self-attention.

TABLE 2

Registration results w/o RANSAC on 3DMatch (3DM) and 3DLoMatch (3DLM). The *model time* is the time for feature extraction, while the *pose time* is the time for transformation estimation. The time is averaged over all point cloud pairs in 3DMatch and 3DLoMatch. † indicates the lite model with shared geometric self-attention. **Boldfaced** numbers highlight the best and the second best are underlined.

Model	Estimator	#Samples	RR(%)		Time(s)		
			3DM	3DLM	Model	Pose	Total
FCGF [3]	RANSAC-50k	5000	85.1	40.1	0.052	3.326	3.378
D3Feat [4]	RANSAC-50k	5000	81.6	37.2	0.024	3.088	3.112
SpinNet [7]	RANSAC-50k	5000	88.6	59.8	60.248	0.388	60.636
Predator [5]	RANSAC-50k	5000	89.0	59.8	0.032	5.120	5.152
CoFiNet [6]	RANSAC-50k	5000	89.3	67.5	0.115	1.807	1.922
GeoTransformer (ours)	RANSAC-50k	5000	92.3	75.4	0.075	1.558	1.633
GeoTransformer† (ours)	RANSAC-50k	5000	<u>92.2</u>	<u>74.9</u>	0.060	1.546	1.606
FCGF [3]	weighted SVD	250	42.1	3.9	0.052	0.008	0.056
D3Feat [4]	weighted SVD	250	37.4	2.8	0.024	0.008	0.032
SpinNet [7]	weighted SVD	250	34.0	2.5	60.248	0.006	60.254
Predator [5]	weighted SVD	250	50.0	6.4	0.032	0.009	0.041
CoFiNet [6]	weighted SVD	250	64.6	21.6	0.115	0.003	0.118
GeoTransformer (ours)	weighted SVD	250	86.7	60.5	0.075	0.003	0.078
GeoTransformer† (ours)	weighted SVD	250	87.5	61.4	0.060	0.003	0.063
CoFiNet [6]	LGR	all	87.6	64.8	0.115	0.028	0.143
GeoTransformer (ours)	LGR	all	91.8	74.5	0.075	0.013	0.088
GeoTransformer† (ours)	LGR	all	91.8	<u>74.2</u>	0.060	0.013	0.073

the correspondences with mutual nearest neighbor selection on their descriptors, and then the top 250 correspondences are used to compute the transformation. The baselines either fail to achieve reasonable results or suffer from severe performance degradation. In contrast, GeoTransformer (with weighted SVD) achieves the registration recall of 86.7% on 3DMatch and 60.5% on 3DLoMatch, close to Predator with RANSAC. Note that the lite model performs even better than the standard model thanks to the higher inlier ratio. Without outlier filtering by RANSAC, high inlier ratio is necessary for successful registration. However, high inlier ratio does not necessarily lead to high registration recall since the correspondences could cluster together as noted in [5]. Nevertheless, our method without RANSAC

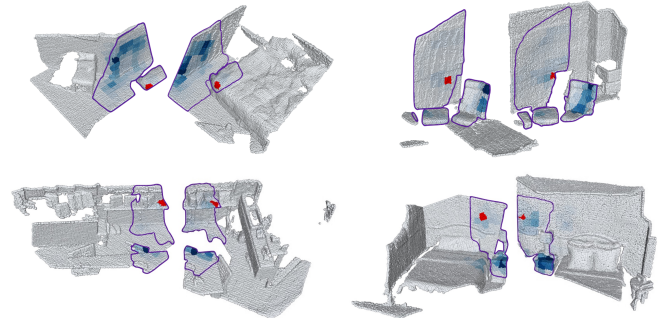


Fig. 8. **Visualizing geometric self-attention scores** on four pairs of point clouds. The overlap areas are delineated with purple lines. The anchor patches (in correspondence) are highlighted in red and the attention scores to other patches are color-coded (*deeper is larger*). Note how the attention patterns of the two matching anchors are consistent even across disjoint overlap areas.

performs well by extracting reliable and well-distributed superpoint correspondences.

When using our local-to-global registration (LGR) for computing transformation, our method brings the registration recall to 91.8% on 3DMatch and 74.5% on 3DLoMatch, surpassing all RANSAC-based baselines by a large margin. The results are also very close to those of ours with RANSAC, but LGR gains over 100 times acceleration over RANSAC in the pose time. These results demonstrate the superiority of our method in both accuracy and speed. Moreover, our lite model achieves very similar results but reduces the overall time by 17%, running at 13 fps. We believe it has a good potential in real-time applications such as online 3D reconstruction and camera relocalization.

Qualitative results. We provide some qualitative comparison of CoFiNet [6] and GeoTransformer on 3DLoMatch in Fig. 7. Our method performs quite well in these low-overlap cases. It is noteworthy that our method can distinguish similar objects at different positions (see the comparison of CoFiNet and GeoTransformer in the 2nd and 3rd rows on the

TABLE 3
Registration results on KITTI odometry. Boldfaced numbers highlight the best and the second best are underlined.

Model	RTE(cm)	RRE(°)	RR(%)
3DFeat-Net [51]	25.9	0.25	96.0
FCGF [3]	9.5	0.30	<u>96.6</u>
D3Feat [4]	<u>7.2</u>	0.30	99.8
SpinNet [7]	9.9	0.47	99.1
Predator [5]	6.8	<u>0.27</u>	99.8
CoFiNet [6]	8.2	0.41	99.8
GeoTransformer (<i>ours</i> , RANSAC-50k)	7.4	<u>0.27</u>	99.8
FMR [31]	~66	1.49	90.6
DGR [35]	~32	0.37	98.7
HRegNet [52]	~12	0.29	99.7
GeoTransformer (<i>ours</i> , LGR)	6.8	0.24	99.8

left) and recognize small overlapping regions in complex environment thanks to the geometric structure information obtained from the geometric self-attention.

Fig. 8 visualizes the attention scores learned by our geometric self-attention, which exhibits significant consistency between the anchor patch matches. It shows that our method is able to learn inter-point-cloud geometric consistency which is important to accurate correspondences.

4.3 Outdoor Benchmark: KITTI odometry

Dataset. KITTI odometry [17] consists of 11 sequences of outdoor driving scenarios scanned by LiDAR. We follow [3], [4], [5] and use sequences 0-5 for training, 6-7 for validation and 8-10 for testing. As in [3], [4], [5], the ground-truth poses are refined with ICP and we only use point cloud pairs that are at least 10m away for evaluation.

Metrics. We follow [5] to evaluate our GeoTransformer with three metrics: (1) *Relative Rotation Error* (RRE), the geodesic distance between estimated and ground-truth rotation matrices, (2) *Relative Translation Error* (RTE), the Euclidean distance between estimated and ground-truth translation vectors, and (3) *Registration Recall* (RR), the fraction of point cloud pairs whose RRE and RTE are both below certain thresholds (*i.e.*, $RRE < 5^\circ$ and $RTE < 2m$).

Registration results. In Tab. 3(top), we compare to the state-of-the-art RANSAC-based methods: 3DFeat-Net [51], FCGF [3], D3Feat [4], SpinNet [7], Predator [5] and CoFiNet [6]. Our method performs on par with these methods, showing good generality on outdoor scenes. Note that the backbone in Predator is 2 times wider than that in GeoTransformer, demonstrating the efficacy and the parameter efficiency of our method.

We further compare to three RANSAC-free methods in Tab. 3(bottom): FMR [31], DGR [35] and HRegNet [52]. Our method outperforms all the baselines by large margin. In addition, our method with LGR beats all the RANSAC-based methods. To the best of our knowledge, GeoTransformer is the *first* RANSAC-free method that surpasses RANSAC-based methods on this benchmark.

4.4 Synthetic Benchmark: ModelNet40

Dataset. ModelNet40 [48] contains man-made CAD models from 40 categories. Following [5], [26], we use the processed data from [53], which uniformly samples 2048 points on the surface of each CAD model. We first normalize the

TABLE 4
Registration results on ModelNet40. Boldfaced numbers highlight the best and the second best are underlined.

Model	ModelNet			ModelLoNet		
	RRE(°)	RTE	CD	RRE(°)	RTE	CD
Small Rotation						
RPM-Net [26]	2.357	0.028	0.00130	8.123	0.086	0.00611
RGM [27]	4.548	0.049	0.00268	14.806	0.139	0.01482
Predator [5]	2.064	0.023	0.00145	<u>5.022</u>	<u>0.084</u>	0.00734
CoFiNet [6]	3.584	0.044	0.00205	6.992	0.091	0.00599
GeoTransformer (<i>ours</i>)	<u>2.160</u>	<u>0.024</u>	<u>0.00143</u>	3.638	0.064	0.00448
Large Rotation						
RPM-Net [26]	31.509	0.206	0.01074	51.478	0.346	0.01985
RGM [27]	45.560	0.289	0.01697	68.724	0.442	0.03634
Predator [5]	24.839	0.171	0.01940	46.990	0.378	0.05052
CoFiNet [6]	<u>10.496</u>	<u>0.084</u>	<u>0.00319</u>	<u>32.578</u>	<u>0.226</u>	<u>0.02273</u>
GeoTransformer (<i>ours</i>)	6.436	0.047	0.00154	23.478	0.152	0.01296

CAD model into a unit sphere and adopt the same strategy as in [26] to generate the source and the target point clouds: a half-space with a random direction is sampled and shifted to retain a proportion p of the points. The source point cloud is then randomly transformed with a rotation within $[0, r]$ and a translation within $[-0.5, 0.5]$. Both point clouds are then jittered with a noise sampled from $\mathcal{N}(0, 0.01)$ and clipped to $[-0.05, 0.05]$. At last, 717 points are randomly sampled from each point cloud independently as the final point cloud pair. We evaluate our method on two overlap settings (*ModelNet* with $p=0.7$ and *ModelLoNet* with $p=0.5$) and two rotation settings (*Small* with $r=45^\circ$ and *Large* with $r=180^\circ$). We follow [5] to use the first 20 categories in the official training/testing split for training/validation and the other 20 categories in the official testing split for testing. We further remove 8 symmetric categories (*i.e.*, bottle, bowl, cone, cup, flower pot, lamp, tent, and vase) as their poses are ambiguous. As a result, we have 4194 CAD models for training, 1002 for validation, and 1146 for testing.

Metrics. We follow [26] to evaluate GeoTransformer with two metrics: (1) *Relative Rotation Error* (RRE), (2) *Relative Translation Error* (RTE), and (3) *Chamfer Distance* (CD) between two aligned point clouds. And we use the modified Chamfer distance from [26] which compares with the clean and complete versions of the other point cloud.

Registration results. We compare GeoTransformer with four baseline methods in Tab. 4: RPM-Net [26], RGM [27], Predator [5] and CoFiNet [6]. RPM-Net and RGM are end-to-end registration methods, while Predator, CoFiNet and GeoTransformer are correspondence-based methods. All the models are train for 200 epochs. For fair comparison, we adopt the same KPConv-based backbone in Predator, CoFiNet and GeoTransformer. To estimate the transformation, Predator and CoFiNet use RANSAC-50k while LGR is used in GeoTransformer. As the point clouds are relatively small, we use $N_c = 128$ superpoint correspondences during testing.

When the rotation is small, RPM-Net, Predator and GeoTransformer achieve comparable results on the high-overlap setting. As this setting is relatively easy, the performance tends to be saturated. For the low-overlap setting, GeoTransformer surpasses other methods by a large margin, demonstrating the effectiveness of our method. When the rotation is large, as all the methods are not completely

TABLE 5
Registration results on Augmented ICL-NUIM. ATE (cm) are reported. **Boldfaced** numbers highlight the best and the second best are underlined.

Model	Living1	Living2	Office1	Office2	Mean
FGR [54]	78.97	24.91	14.96	21.05	34.98
RANSAC [55]	110.9	19.33	14.42	17.31	40.49
DGR [35]	21.06	21.88	15.76	11.56	17.57
PointDSC [36]	<u>20.25</u>	<u>15.58</u>	13.56	11.30	15.18
GeoTransformer (ours)	17.54	15.31	<u>13.85</u>	9.78	14.12

invariant to transformation (*i.e.*, the backbone part), the performance inevitably drops compared with the small-rotation setting. Nevertheless, as the geometric self-attention provides more structure information about the point clouds, our GeoTransformer attains significantly better results than the baseline methods on both high- and low-overlap settings, showing strong robustness to low overlap and large rotations. Moreover, it is noteworthy that GeoTransformer requires neither iterative registration as in RPM-Net and RGM, nor RANSAC as in Predator and CoFiNet, thus achieves very fast registration speed.

Besides, we have more interesting observations. First, the correspondence-based methods perform much better than the end-to-end methods in complicated scenarios with large perturbations or heavy occlusion. In this case, the end-to-end methods have difficulty learning accurate soft correspondences, while the correspondence-based methods are more robust because they establish hard correspondences directly from the existing points. And robust estimators such as RANSAC further improve the stability of them. Second, the coarse-to-fine methods are more robust to large rotations than the detection-based methods. Compared with directly matching dense points with their descriptors, the superpoint matching is more sparse and discriminative. The two-stage pipeline can effectively alleviate the risk of mismatching and contributes to better registration performance.

4.5 Multiway Benchmark: Augmented ICL-NUIM

Dataset. Augmented ICL-NUIM [18] augments the synthetic scenes in ICL-NUIM [56] with a realistic noise model. It consists of four camera trajectories from two scenes for testing. Following [35], [36], we fuse 50 consecutive RGB-D frames to generate the point cloud fragments. To solve multiway registration, we follow [35], [36] to first conduct pair-wise registration with GeoTransformer and then optimize the poses with the global pose graph optimization [57] implemented in [58].

Metrics. We follow [35], [36] to evaluate GeoTransformer with the metric of *Absolute Trajectory Error* (ATE). It first aligns the ground-truth and the estimated trajectories with SVD, and then computes the root mean square error (RMSE) of the differences between the points at the same timestamp in the two trajectories.

Registration results. We compare GeoTransformer with four baseline methods in Tab. 5: FGR [54], RANSAC [55], DGR [35], and PointDSC [36]. Following [36], we directly use the models trained on 3DMatch without fine-tuning. However, the point clouds in Augmented ICL-NUIM are larger than those in 3DMatch due to faster camera motion,

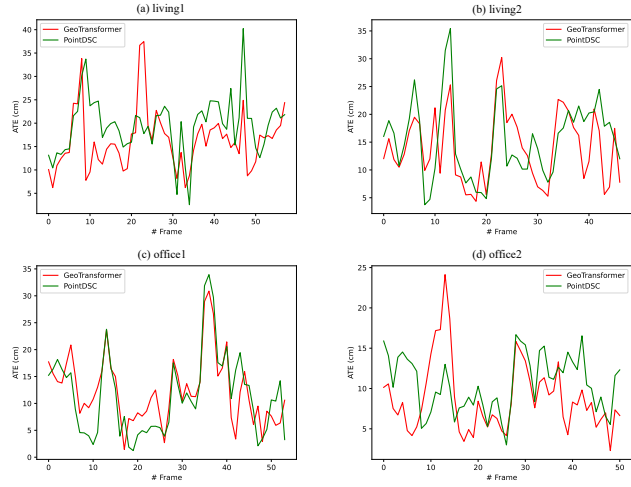


Fig. 9. **Comparison of per-frame ATE on Augmented ICL-NUIM.** Our GeoTransformer attains better results on most of the frames.

so we further downsample the superpoints to reduce memory footprint. And the superpoint features $\hat{\mathbf{F}}^P$ and $\hat{\mathbf{F}}^Q$ are downsampled by k NN interpolation, where the geometric transformer module is then applied. At last, the resultant features are interpolated and upsampled to generate $\hat{\mathbf{H}}^P$ and $\hat{\mathbf{H}}^Q$. This modification effectively improves the memory efficiency without sacrificing the performance. GeoTransformer attains the best performance on all testing trajectories except *Office1*, showing strong generality to unknown scenes and more complex applications. Fig. 9 visualizes the ATE of each frame in the four trajectories and GeoTransformer achieves better results on most of the frames.

4.6 Non-rigid Benchmark: 4DMatch & 4DLoMatch

Dataset. 4DMatch [19] is a challenging benchmark for non-rigid point cloud registration. It is constructed using the animation sequences from DeformingThings4D [59], where 1232 sequences are used for training, 176 for validation and 353 for testing. The point cloud pairs in the testing sequences are divided into 4DMatch and 4DLoMatch based on an overlapping ratio threshold of 45%.

Metrics. Following [19], we evaluate our GeoTransformer with two metrics: (1) *Non-rigid Inlier Ratio* (NIR), the fraction of putative correspondences whose residuals are below a certain threshold (*i.e.*, 0.04m) under the ground-truth warping function, and (2) *Non-rigid Feature Matching Recall* (NFMR), the fraction of the ground-truth matches that can be successfully recovered by the putative correspondences.

Implementation details. Unlike rigid registration which can be pinned down by a set of sparse correspondences, non-rigid registration is more challenging due to the complex and irregular deformation and requires denser correspondences to cover the overlapping region as much as possible. To this end, we modify our superpoint matching strategy to increase the overall coverage of the correspondences. Specifically, we first select the superpoint matches whose feature distances are below a certain threshold (*i.e.*, 0.75), and augment them with the top 128 ones if there are too few superpoint matches. We use all point correspondences for evaluation and LGR is not performed.

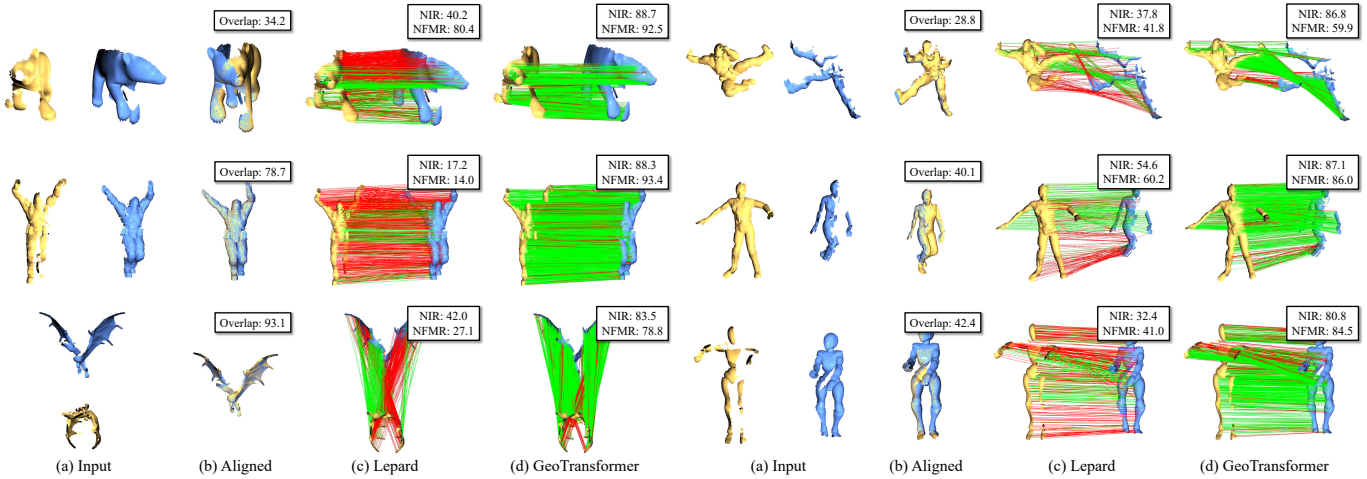


Fig. 10. Comparison of the correspondences on 4DMatch and 4DLoMatch. GeoTransformer shows two advantages. First, it extracts much denser correspondences, which contributes to more precise description of the deformations. Second, it achieves higher inlier ratio despite significant deformations, which is important for non-rigid registration.

TABLE 6

Evaluation results on 4DMatch and 4DLoMatch. NIR and NFM are measured in %. Our method uses shared geometric self-attention. **Boldfaced** numbers are the best and the second best are underlined.

Model	4DMatch			4DLoMatch		
	# Corr	NIR	NFM	# Corr	NIR	NFM
D3Feat [4]	697	55.3	56.1	204	21.3	28.1
Predator [5]	698	59.3	56.8	480	25.0	32.1
Lepard [19]	596	82.7	83.7	407	<u>55.7</u>	66.9
Lepard (w/o repos) [19]	624	80.5	80.8	448	53.7	63.6
GeoTransformer (ours)	2331	<u>82.2</u>	<u>83.2</u>	1212	63.6	<u>65.4</u>

Evaluation results. We compare GeoTransformer with three recent methods in Tab. 6: D3Feat [4], Predator [5] and Lepard [19]. Our method surpasses D3Feat and Predator by a large margin on both high- and low-overlap scenarios. Compared with Lepard, GeoTransformer achieves very close performance on 4DMatch and significantly better inlier ratio on 4DLoMatch. Note that Lepard benefits from a repositioning mechanism with a coarse rigid registration, which effectively boosts the performance. Without repositioning, our GeoTransformer consistently outperforms Lepard on both benchmarks. Albeit not carefully designed and optimized to handle deformation, GeoTransformer shows strong generality to non-rigid registration. In most cases, the non-rigid deformation could be approximated by a set of local rigid transformations. We suppose that the novel geometric self-attention endows GeoTransformer with the capability to capture the local rigidity consistency between two point clouds, which helps extracting high-quality correspondences in non-rigid scenarios.

Qualitative results. Fig. 10 compares the correspondences from Lepard [19] and GeoTransformer on some cases with relatively large deformations. GeoTransformer has two advantages as shown in these cases. First, our method can extract much denser correspondences, which is important for precisely describing the deformation. Due to the irregularity of the deformations, it is difficult to capture the deformation details if the correspondences are too sparse. Second, our method attains much higher inlier ratio especially in

low-overlap cases, which benefits the following registration algorithms such as Non-rigid ICP [60], [61]. As there are few effective outlier rejection methods for non-rigid registration, high inlier ratio is crucial for estimating a proper deformation field. Thanks to the coarse-to-fine framework and the geometric self-attention, our method can establish high-quality correspondences for non-rigid registration.

4.7 Ablation Studies

We conduct extensive ablation studies on 3DMatch and 3DLoMatch for a better understanding of the various modules in our method. To evaluate superpoint (patch) matching, we introduce another metric *Patch Inlier Ratio* (PIR) which is the fraction of patch matches with actual overlap. The FMR and IR are reported with *all* global dense point correspondences, with LGR being used for registration.

Geometric self-attention. To study the effectiveness of the geometric self-attention, we compare seven methods for intra-point-cloud feature learning in Tab. 7(a.1-7): (1) graph neural network [5], (2) self-attention with no positional embedding [6], (3) absolute coordinate embedding [15], (4) relative coordinate embedding [13], (5) point pair features embedding [38], [39], (6) pair-wise distance embedding, and (7) geometric structure embedding. Generally, injecting geometric information boosts the performance. But the gains of coordinate-based embeddings are limited due to their transformation variance. Surprisingly, GNN performs well on RR thanks to the transformation invariance of k NN graphs. However, it suffers from limited receptive fields which harms the IR performance. Although PPF embedding is theoretically invariant to transformation, it is hard to estimate accurate normals for the downsampled superpoints in practice, which leads to inferior performance. Our method outperforms the alternatives by a large margin on all the metrics, especially in the low-overlap scenarios, even with only the pair-wise distance embedding, demonstrating the strong robustness of our method. Fig. 11 provides a gallery of the registration results of the models with vanilla self-attention and our geometric self-attention. Geometric self-attention helps infer patch matches in structure-less regions

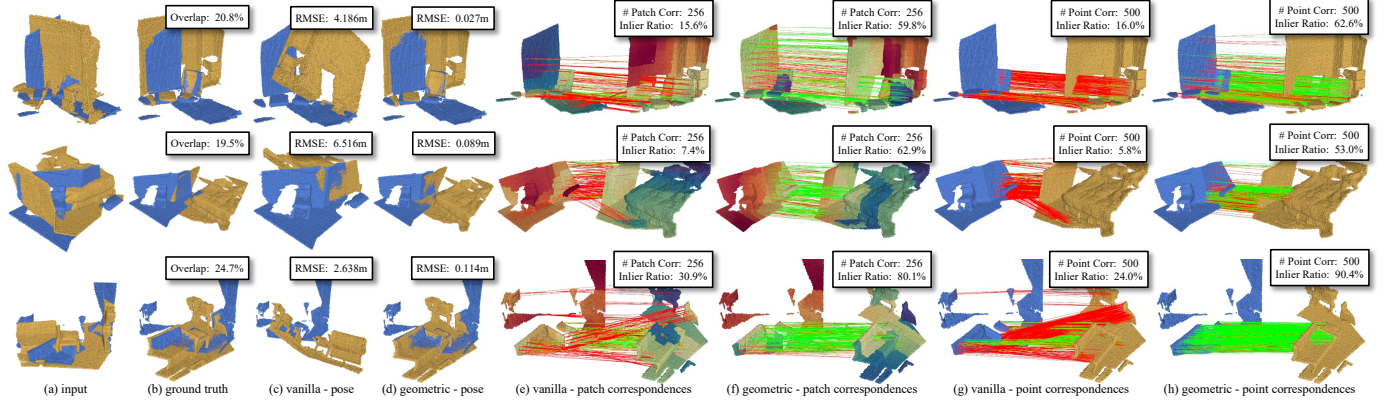


Fig. 11. Registration results of the models with vanilla self-attention and geometric self-attention. In the columns (e) and (f), we visualize the features of the patches with t-SNE. In the first row, the geometric self-attention helps find the inlier matches on the structure-less wall based on their geometric relationships to the more salient regions (e.g., the chairs). In the following rows, the geometric self-attention helps reject the outlier matches between the similar flat or corner patches based on their geometric relationships to the bed or the sofa.

TABLE 7

Ablation experiments on 3DMatch and 3DLoMatch. The results are measured in %. * indicates the default settings of GeoTransformer. **Boldfaced** numbers are the best and the second best are underlined.

Model	3DMatch				3DLoMatch			
	PIR	FMR	IR	RR	PIR	FMR	IR	RR
(a.1) graph neural network	73.3	97.9	56.5	89.5	39.4	84.9	29.2	69.8
(a.2) vanilla self-attention	79.6	97.9	60.1	89.0	45.2	85.6	32.6	68.4
(a.3) self-attention w/ ACE	83.2	98.1	68.5	89.3	48.2	84.3	38.9	69.3
(a.4) self-attention w/ RCE	80.0	97.9	66.1	88.5	46.1	84.6	37.9	68.7
(a.5) self-attention w/ PPF	83.5	97.5	68.5	88.6	49.8	83.8	39.9	69.5
(a.6) self-attention w/ RDE	<u>84.9</u>	<u>98.0</u>	<u>69.1</u>	<u>90.7</u>	<u>50.6</u>	<u>85.8</u>	<u>40.3</u>	<u>72.1</u>
(a.7) geometric self-attention*	86.1	98.1	71.0	91.8	54.6	87.8	43.8	74.5
(b.1) cross-entropy loss	80.0	97.7	65.7	90.0	45.9	85.1	37.4	68.4
(b.2) weighted cross-entropy loss	83.2	98.0	67.4	90.0	49.0	86.2	38.6	70.7
(b.3) circle loss	85.1	97.8	<u>69.5</u>	<u>90.4</u>	<u>51.5</u>	<u>86.1</u>	<u>41.3</u>	<u>71.5</u>
(b.4) overlap-aware circle loss*	86.1	98.1	71.0	91.8	54.6	87.8	43.8	74.5
(c.1) distance only	84.9	98.0	69.1	90.7	50.6	85.8	40.3	72.1
(c.2) $k = 1$ angles	86.5	97.9	70.6	91.0	54.6	87.1	42.7	73.1
(c.3) $k = 2$ angles	86.1	97.9	70.4	91.3	55.0	88.2	43.5	73.5
(c.4) $k = 3$ angles*	86.1	98.1	71.0	91.8	54.6	87.8	43.8	74.5
(c.5) $k = 4$ angles	<u>86.4</u>	98.2	71.1	92.1	<u>54.8</u>	<u>87.8</u>	43.9	75.1
(d.1) $\sigma_d = 0.1m$	<u>86.6</u>	97.6	71.4	90.7	<u>54.6</u>	<u>87.7</u>	43.8	73.2
(d.2) $\sigma_d = 0.2m^*$	86.1	98.1	71.0	91.8	<u>54.6</u>	87.8	43.8	74.5
(d.3) $\sigma_d = 0.3m$	86.7	98.4	70.3	92.0	55.3	87.0	43.0	74.1
(d.4) $\sigma_d = 0.4m$	86.7	<u>98.1</u>	71.4	<u>91.8</u>	54.3	87.8	<u>43.5</u>	74.0
(d.5) $\sigma_d = 0.5m$	86.0	97.8	70.3	91.0	54.2	86.3	43.3	73.7
(e.1) $\sigma_a = 5^\circ$	86.1	97.9	70.4	91.3	53.7	86.9	42.4	72.6
(e.2) $\sigma_a = 10^\circ$	87.0	98.0	71.4	91.4	54.5	87.3	43.6	74.2
(e.3) $\sigma_a = 15^\circ^*$	86.1	98.1	71.0	91.8	<u>54.6</u>	87.8	43.8	74.5
(e.4) $\sigma_a = 20^\circ$	<u>86.7</u>	97.9	70.7	92.1	54.7	86.7	43.0	73.6
(e.5) $\sigma_a = 25^\circ$	86.5	97.8	70.6	91.2	54.0	86.6	42.7	73.6
(f.1) w/ max pooling*	86.1	98.1	71.0	91.8	54.6	87.8	43.8	74.5
(f.2) w/ average pooling	86.3	98.0	70.2	91.3	54.6	87.3	42.8	74.0
(g.1) w/ dual-normalization*	86.1	98.1	71.0	91.8	54.6	87.8	43.8	74.5
(g.2) w/o dual-normalization	86.2	98.1	70.9	91.8	53.5	87.9	43.4	74.4

from their geometric relationships to more salient regions (1st row) and reject outlier matches which are similar in the feature space but different in positions (2nd and 3rd rows).

Overlap-aware circle loss. To investigate the efficacy of the overlap-aware circle loss, we compare four loss functions for supervising the superpoint matching in Tab. 7(b.1-4): (1) cross-entropy loss [15], (2) weighted cross-entropy loss [6], (3) circle loss [47], and (4) overlap-aware circle loss. For the first two models, an optimal transport layer is used to com-

pute the matching matrix as in [6]. Circle loss works much better than the two variants of cross-entropy loss, verifying the effectiveness of supervising superpoint matching in a metric learning fashion. Our overlap-aware circle loss beats the vanilla circle loss by a large margin on all the metrics.

Geometric structure embedding. Next, we study the design of geometric structure embedding. We first vary the number of nearest neighbors for computing the triplet-wise angular embedding. As shown in Tab. 7(c.1-5), the model with both the distance and angular embeddings outperforms the one with only the distance embedding by a large margin, which is consistent with our motivation. Moreover, increasing the number of neighbors slightly improves the performance as it provides more precise structure information, but also requires more computation. To better balance accuracy and speed, we select $k=3$ in our experiments unless otherwise noted.

We further investigate the influence of the temperature hyper-parameters σ_d in Eq. (5) and σ_a in Eq. (6). From Tab. 7(d.1-5), the best results are achieved around the voxel size of the superpoint level (i.e., 0.2m). A too small (where the embedding is too sensitive to distance changes) or too large (where the embedding neglects small distance variations) σ_d could harm the performance, but the differences are not significant. And similar observations can be obtained from Tab. 7(e.1-5) for the angular temperature σ_a . Nevertheless, all of these models outperforms previous methods by a large margin, indicating that GeoTransformer is still robust to the temperature hyper-parameters.

At last, we replace max pooling with average pooling when aggregating the triplet-wise angular embedding in Eq. (7). As shown in Tab. 7(f.1-2), max pooling performs better than average pooling. Due to self-occlusion from viewpoint changes, the nearest neighbors of a given superpoint in one point cloud could be missing in the other. Compared with average pooling, max pooling provides better robustness to the varying neighbors. For this reason, we use max pooling as the default setting.

Dual-normalization. We then investigate the effectiveness of the dual-normalization operation in the superpoint matching module. As observed in Tab. 7(g.1-2), it slightly improves the accuracy of the superpoint correspondences in

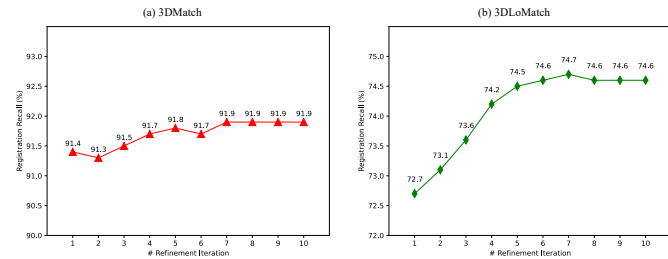


Fig. 12. Ablation study of the pose refinement. Pose refinement consistently improves the results and gets saturated after 5 iterations.

low-overlap scenarios. As there is less overlapping context when the overlapping area is small, it is much easier to extract outlier matches between the less geometrically discriminative patches. The dual-normalization operation can mitigate this issue and slightly improves the performance.

Pose refinement. At last, we evaluate the impact of the pose refinement in LGR. We vary the number of refinement steps N_r from 1 to 10. As shown in Fig. 12, the registration recall consistently improves with more refinement iterations and quickly gets saturated. To better balance accuracy and speed, we choose 5 iterations in the experiments.

4.8 Comparison with Deep Robust Estimators

At last, we compare GeoTransformer with recent deep robust estimators: 3DRegNet [34], DGR [35], PointDSC [36], DHVR [62] and PCAM [63] on 3DMatch and KITTI odometry benchmarks. For fair comparison with these methods, we follow common practice to report RTE, RRE and RR on both benchmarks. Here RR is defined as in Sec. 4.3 but with different thresholds. The RTE threshold is 30cm on 3DMatch and 60cm on KITTI, while the RRE threshold is 15° on 3DMatch and 5° on KITTI.

As shown in Tab. 8, our method outperforms all the baselines by a large margin on both benchmarks. The results demonstrate the superiority of GeoTransformer over the alternative methods, although different correspondence extractors are used by those methods. It is noteworthy that our LGR is parameter-free and does not require training a specific network, which contributes to faster registration speed (0.08s of PointDSC [36] vs. 0.013s of LGR according to our experiments).

5 CONCLUSION

We have presented Geometric Transformer to learn robust coarse-to-fine correspondences for point cloud registration. Through encoding pair-wise distances and triplet-wise angles among superpoints, our method captures the geometric consistency across point clouds with transformation invariance. Thanks to the reliable correspondences, it attains fast and accurate registration in a RANSAC-free manner. Extensive experiments on five challenging benchmarks have demonstrated the efficacy of GeoTransformer.

Limitations. In spite of the state-of-the-art performance, there are still some limitations in GeoTransformer. (1) GeoTransformer relies on uniformly downsampled superpoints to hierarchically extract correspondences. However, there

TABLE 8 Comparison with deep robust estimators on 3DMatch and KITTI. Boldfaced numbers are the best and the second best are underlined.

Model	RTE(cm)	RRE(°)	RR(%)
3DMatch			
FCGF+3DRegNet [34]	8.13	2.74	77.8
FCGF+DGR [35]	7.36	2.33	86.5
FCGF+PointDSC [36]	6.55	2.06	<u>93.3</u>
FCGF+DHVR [62]	6.61	2.08	91.4
PCAM [63]	~7	2.16	92.4
GeoTransformer (ours, LGR)	5.69	1.92	95.7
3DLoMatch			
FCGF+PointDSC [36]	10.50	3.82	56.2
FCGF+DHVR [62]	11.76	3.88	55.6
GeoTransformer (ours, LGR)	8.55	2.95	78.0
KITTI			
FCGF+DGR [35]	21.7	0.34	96.9
FCGF+PointDSC [36]	20.9	0.33	98.2
FCGF+DHVR [62]	19.8	<u>0.29</u>	<u>99.1</u>
PCAM [63]	~8	0.33	97.2
GeoTransformer (ours, LGR)	6.5	0.24	99.5

could be numerous superpoints if the input point clouds cover a large area, which could cause huge memory footprint and computational cost. For this reason, we might need to carefully select the downsampling rate to balance performance and efficiency. For example, we add an additional downsampling stage on KITTI and Augmented ICL-NUIM, which effectively improves the memory and computational efficiency without sacrificing the accuracy. (2) The inflexibility of uniformly sampling superpoints (patches) is another concern. In practice, it is common that a single object is decomposed into multiple patches, but it could be easily registered as a whole. So we believe that it is a very promising topic to integrate point cloud registration with semantic scene understanding tasks (e.g., object detection and instance segmentation), which converts scene registration into semantic object registration.

Future work. Besides the aforementioned limitations, there are also many directions where GeoTransformer could be extended, including cross-modality (e.g., 2D-3D) registration and end-to-end non-rigid registration.

ACKNOWLEDGMENTS

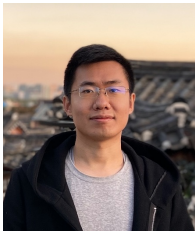
This work was supported in part by the NSFC (62132021, 62102435) and the National Key Research and Development Program of China (2018AAA0102200).

REFERENCES

- [1] H. Deng, T. Birdal, and S. Ilic, "Ppfnet: Global context aware local features for robust 3d point matching," in CVPR, 2018, pp. 195–205.
- [2] Z. Gojcic, C. Zhou, J. D. Wegner, and A. Wieser, "The perfect match: 3d point cloud matching with smoothed densities," in CVPR, 2019, pp. 5545–5554.
- [3] C. Choy, J. Park, and V. Koltun, "Fully convolutional geometric features," in CVPR, 2019, pp. 8958–8966.
- [4] X. Bai, Z. Luo, L. Zhou, H. Fu, L. Quan, and C.-L. Tai, "D3feat: Joint learning of dense detection and description of 3d local features," in CVPR, 2020, pp. 6359–6367.
- [5] S. Huang, Z. Gojcic, M. Usvyatsov, A. Wieser, and K. Schindler, "Predator: Registration of 3d point clouds with low overlap," in CVPR, 2021, pp. 4267–4276.

- [6] H. Yu, F. Li, M. Saleh, B. Busam, and S. Ilic, "Cofinet: Reliable coarse-to-fine correspondences for robust pointcloud registration," *NeurIPS*, vol. 34, 2021.
- [7] S. Ao, Q. Hu, B. Yang, A. Markham, and Y. Guo, "Spinnet: Learning a general surface descriptor for 3d point cloud registration," in *CVPR*, 2021, pp. 11 753–11 762.
- [8] I. Rocco, M. Cimpoi, R. Arandjelović, A. Torii, T. Pajdla, and J. Sivic, "Neighbourhood consensus networks," *NeurIPS*, vol. 31, pp. 1651–1662, 2018.
- [9] Q. Zhou, T. Sattler, and L. Leal-Taixe, "Patch2pix: Epipolar-guided pixel-level correspondences," in *CVPR*, 2021, pp. 4669–4678.
- [10] J. Sun, Z. Shen, Y. Wang, H. Bao, and X. Zhou, "Loftr: Detector-free local feature matching with transformers," in *CVPR*, 2021, pp. 8922–8931.
- [11] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," in *NeurIPS*, 2017, pp. 5998–6008.
- [12] Y. Wang and J. M. Solomon, "Deep closest point: Learning representations for point cloud registration," in *ICCV*, 2019, pp. 3523–3532.
- [13] H. Zhao, L. Jiang, J. Jia, P. H. Torr, and V. Koltun, "Point transformer," in *ICCV*, 2021, pp. 16 259–16 268.
- [14] J. Yang, Q. Zhang, B. Ni, L. Li, J. Liu, M. Zhou, and Q. Tian, "Modeling point clouds with self-attention and gumbel subset sampling," in *CVPR*, 2019, pp. 3323–3332.
- [15] P.-E. Sarlin, D. DeTone, T. Malisiewicz, and A. Rabinovich, "Superglue: Learning feature matching with graph neural networks," in *CVPR*, 2020, pp. 4938–4947.
- [16] A. Zeng, S. Song, M. Nießner, M. Fisher, J. Xiao, and T. Funkhouser, "3dmatch: Learning local geometric descriptors from rgb-d reconstructions," in *CVPR*, 2017, pp. 1802–1811.
- [17] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for autonomous driving? the kitti vision benchmark suite," in *CVPR*. IEEE, 2012, pp. 3354–3361.
- [18] S. Choi, Q.-Y. Zhou, and V. Koltun, "Robust reconstruction of indoor scenes," in *CVPR*, 2015, pp. 5556–5565.
- [19] Y. Li and T. Harada, "Lepard: Learning partial point cloud matching in rigid and deformable scenes," in *CVPR*, 2022, pp. 5554–5564.
- [20] Z. Qin, H. Yu, C. Wang, Y. Guo, Y. Peng, and K. Xu, "Geometric transformer for fast and robust point cloud registration," in *CVPR (oral)*, 2022, pp. 11 143–11 152.
- [21] H. Deng, T. Birdal, and S. Ilic, "Ppf-foldnet: Unsupervised learning of rotation invariant 3d local descriptors," in *ECCV*, 2018, pp. 602–618.
- [22] H. Wang, Y. Liu, Z. Dong, W. Wang, and B. Yang, "You only hypothesize once: Point cloud registration with rotation-equivariant descriptors," in *ACM MM*, 2022, pp. 1630–1641.
- [23] Y. Wang and J. Solomon, "Prnet: self-supervised learning for partial-to-partial registration," in *NeurIPS*, 2019, pp. 8814–8826.
- [24] W. Yuan, B. Eckart, K. Kim, V. Jampani, D. Fox, and J. Kautz, "Deepgmr: Learning latent gaussian mixture models for registration," in *ECCV*. Springer, 2020, pp. 733–750.
- [25] J. Li, C. Zhang, Z. Xu, H. Zhou, and C. Zhang, "Iterative distance-aware similarity matrix convolution with mutual-supervised point elimination for efficient point cloud registration," in *ECCV*. Springer, 2020, pp. 378–394.
- [26] Z. J. Yew and G. H. Lee, "Rpm-net: Robust point matching using learned features," in *CVPR*, 2020, pp. 11 824–11 833.
- [27] K. Fu, S. Liu, X. Luo, and M. Wang, "Robust point cloud registration framework based on deep graph matching," in *CVPR*, 2021, pp. 8893–8902.
- [28] Z. Zhang, J. Sun, Y. Dai, D. Zhou, X. Song, and M. He, "End-to-end learning the partial permutation matrix for robust 3d point cloud registration," in *AAAI*, vol. 36, no. 3, 2022, pp. 3399–3407.
- [29] P. J. Besl and N. D. McKay, "Method for registration of 3-d shapes," in *Sensor fusion IV: control paradigms and data structures*, vol. 1611. International Society for Optics and Photonics, 1992, pp. 586–606.
- [30] Y. Aoki, H. Goforth, R. A. Srivatsan, and S. Lucey, "Pointnetlk: Robust & efficient point cloud registration using pointnet," in *CVPR*, 2019, pp. 7163–7172.
- [31] X. Huang, G. Mei, and J. Zhang, "Feature-metric registration: A fast semi-supervised approach for robust point cloud registration without correspondences," in *CVPR*, 2020, pp. 11 366–11 374.
- [32] H. Xu, S. Liu, G. Wang, G. Liu, and B. Zeng, "Omnet: Learning overlapping mask for partial-to-partial point cloud registration," in *ICCV*, 2021, pp. 3132–3141.
- [33] H. Xu, N. Ye, G. Liu, B. Zeng, and S. Liu, "Finet: Dual branches feature interaction for partial-to-partial point cloud registration," in *AAAI*, vol. 36, no. 3, 2022, pp. 2848–2856.
- [34] G. D. Pais, S. Ramalingam, V. M. Govindu, J. C. Nascimento, R. Chellappa, and P. Miraldo, "3dregnet: A deep neural network for 3d point registration," in *CVPR*, 2020, pp. 7193–7203.
- [35] C. Choy, W. Dong, and V. Koltun, "Deep global registration," in *CVPR*, 2020, pp. 2514–2523.
- [36] X. Bai, Z. Luo, L. Zhou, H. Chen, L. Li, Z. Hu, H. Fu, and C.-L. Tai, "Pointdsc: Robust point cloud registration using deep spatial consistency," in *CVPR*, 2021, pp. 15 859–15 869.
- [37] R. B. Rusu, N. Blodow, and M. Beetz, "Fast point feature histograms (fpfh) for 3d registration," in *ICRA*. IEEE, 2009, pp. 3212–3217.
- [38] B. Drost, M. Ulrich, N. Navab, and S. Ilic, "Model globally, match locally: Efficient and robust 3d object recognition," in *CVPR*. IEEE, 2010, pp. 998–1005.
- [39] C. Raposo and J. P. Barreto, "Using 2 point+ normal sets for fast registration of point clouds with small overlap," in *ICRA*. IEEE, 2017, pp. 5652–5658.
- [40] M. Leordeanu and M. Hebert, "A spectral technique for correspondence problems using pairwise constraints," in *ICCV*, vol. 2. IEEE, 2005, pp. 1482–1489.
- [41] H. Yang, J. Shi, and L. Carlone, "Teaser: Fast and certifiable point cloud registration," *IEEE Transactions on Robotics*, vol. 37, no. 2, pp. 314–333, 2020.
- [42] H. Thomas, C. R. Qi, J.-E. Deschaud, B. Marcotegui, F. Goulette, and L. J. Guibas, "Kpconv: Flexible and deformable convolution for point clouds," in *ICCV*, 2019, pp. 6411–6420.
- [43] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature pyramid networks for object detection," in *CVPR*, 2017, pp. 2117–2125.
- [44] J. Li, B. M. Chen, and G. H. Lee, "So-net: Self-organizing network for point cloud analysis," in *CVPR*, 2018, pp. 9397–9406.
- [45] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly *et al.*, "An image is worth 16x16 words: Transformers for image recognition at scale," 2021.
- [46] R. Sinkhorn and P. Knopp, "Concerning nonnegative matrices and doubly stochastic matrices," *Pacific Journal of Mathematics*, vol. 21, no. 2, pp. 343–348, 1967.
- [47] Y. Sun, C. Cheng, Y. Zhang, C. Zhang, L. Zheng, Z. Wang, and Y. Wei, "Circle loss: A unified perspective of pair similarity optimization," in *CVPR*, 2020, pp. 6398–6407.
- [48] Z. Wu, S. Song, A. Khosla, F. Yu, L. Zhang, X. Tang, and J. Xiao, "3d shapenets: A deep representation for volumetric shapes," in *CVPR*, 2015, pp. 1912–1920.
- [49] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga *et al.*, "Pytorch: An imperative style, high-performance deep learning library," *NeurIPS*, vol. 32, pp. 8026–8037, 2019.
- [50] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2015.
- [51] Z. J. Yew and G. H. Lee, "3dfeat-net: Weakly supervised local 3d features for point cloud registration," in *ECCV*, 2018, pp. 607–623.
- [52] F. Lu, G. Chen, Y. Liu, L. Zhang, S. Qu, S. Liu, and R. Gu, "Hregnet: A hierarchical network for large-scale outdoor lidar point cloud registration," in *ICCV*, 2021, pp. 16 014–16 023.
- [53] C. R. Qi, H. Su, K. Mo, and L. J. Guibas, "Pointnet: Deep learning on point sets for 3d classification and segmentation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 652–660.
- [54] Q.-Y. Zhou, J. Park, and V. Koltun, "Fast global registration," in *ECCV*. Springer, 2016, pp. 766–782.
- [55] M. A. Fischler and R. C. Bolles, "Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography," *Communications of the ACM*, vol. 24, no. 6, pp. 381–395, 1981.
- [56] A. Handa, T. Whelan, J. McDonald, and A. J. Davison, "A benchmark for rgb-d visual odometry, 3d reconstruction and slam," in *ICRA*. IEEE, 2014, pp. 1524–1531.
- [57] R. Kümmerle, G. Grisetti, H. Strasdat, K. Konolige, and W. Burgard, "g²o: A general framework for graph optimization," in *ICRA*. IEEE, 2011, pp. 3607–3613.
- [58] Q.-Y. Zhou, J. Park, and V. Koltun, "Open3d: A modern library for 3d data processing," *arXiv preprint arXiv:1801.09847*, 2018.

- [59] Y. Li, H. Takehara, T. Taketomi, B. Zheng, and M. Nießner, "4dcomplete: Non-rigid motion estimation beyond the observable surface," in *ICCV*, 2021, pp. 12706–12716.
- [60] H. Li, R. W. Sumner, and M. Pauly, "Global correspondence optimization for non-rigid registration of depth scans," in *Computer graphics forum*, vol. 27, no. 5. Wiley Online Library, 2008, pp. 1421–1430.
- [61] Y. Yao, B. Deng, W. Xu, and J. Zhang, "Quasi-newton solver for robust non-rigid registration," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 7600–7609.
- [62] J. Lee, S. Kim, M. Cho, and J. Park, "Deep hough voting for robust global registration," in *ICCV*, 2021, pp. 15994–16003.
- [63] A.-Q. Cao, G. Puy, A. Boulch, and R. Marlet, "Pcam: Product of cross-attention matrices for rigid registration of point clouds," in *ICCV*, 2021, pp. 13229–13238.



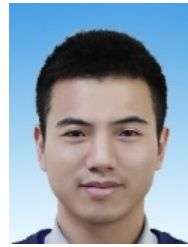
Zheng Qin received the B.E. and M.E. degree in computer science and technology from National University of Defense Technology (NUDT), China, in 2016 and 2018, respectively, where he is currently pursuing the Ph.D. degree. His research interests focus on 3D vision, including point cloud registration, pose estimation, and 3D representation learning.



Hao Yu is a PhD student at the Chair for Computer Aided Medical Procedures & Augmented Reality (CAMP) of TU Munich under supervision of PD. Dr. Slobodan Ilic and Dr. Benjamin Busam. He received his Master's degree in Computer Science from National University of Defense Technology, China, where he also completed his Bachelor's study in Network Engineering. His research interest includes 3D local descriptors and point cloud registration.



Changjian Wang received his Ph.D. degree in computer science from the School of Computer, National University of Defense Technology. He is currently an Associate Professor of the National University of Defense Technology (NUDT), Changsha, China. His current research interests include medical image analysis, natural language processing and big data.



Yulan Guo received the B.E. and Ph.D. degrees from National University of Defense Technology (NUDT) in 2008 and 2015, respectively. He has authored over 100 articles at highly referred journals and conferences. His current research interests focus on 3D vision, particularly on 3D feature learning, 3D modeling, 3D object recognition, and scene understanding. He served as an associate editor for *IEEE Transactions on Image Processing*, *IET Computer Vision*, *IET Image Processing*, and *Computers & Graphics*.

He also served as an area chair for *CVPR 2021*, *ICCV 2021*, and *ACM Multimedia 2021*. He organized several tutorials, workshops, and challenges in prestigious conferences, such as *CVPR 2016*, *CVPR 2019*, *ICCV 2021*, *3DV 2021*, *CVPR 2022*, *ICPR 2022*, and *ECCV 2022*. He is a Senior Member of IEEE and ACM.



Yuxing Peng received the Ph.D. degree from the National University of Defense Technology (NUDT), China, in 1996. He is currently a Professor with the College of Computer Science and Technology, NUDT. His research interests primarily focus on machine learning, data analysis and cloud computing.



Slobodan Ilic is currently senior key expert research scientist at Siemens Corporate Technology in Munich, Perlach. He is also a visiting researcher and lecturer at Computer Science Department of TUM and closely works with the CAMP Chair. From 2009 until end of 2013 he was leading the Computer Vision Group of CAMP at TUM, and before that he was a senior researcher at Deutsche Telekom Laboratories in Berlin. In 2005 he obtained his PhD at EPFL in Switzerland under supervision of Pascal Fua.

His research interests include: 3D reconstruction, deformable surface modelling and tracking, real-time object detection and tracking, human pose estimation and semantic segmentation.



Dewen Hu received the B.Sc. and M.Sc. degrees from Xi'an Jiaotong University, China, in 1983 and 1986, respectively, and the Ph.D. degree from the National University of Defense Technology, in 1999. In 1986, he was with the National University of Defense Technology. From October 1995 to October 1996, he was a Visiting Scholar with The University of Sheffield, U.K. In 1996, he was promoted as a Professor. He has authored more than 200 articles in journals, such as the *Brain*, the *Proceedings of the National Academy of Sciences of the United States of America*, the *NeuroImage*, the *Human Brain Mapping*, the *IEEE Transactions on Pattern Analysis and Machine Intelligence*, the *IEEE Transactions on Image Processing*, the *IEEE Transactions on Signal Processing*, the *IEEE Transactions on Neural Networks and Learning Systems*, the *IEEE Transactions on Medical Imaging*, and the *IEEE Transactions on Biomedical Engineering*. His research interests include pattern recognition and cognitive neuroscience. He is currently an Action Editor of *Neural Networks*, and an Associate Editor of *IEEE Transactions on Systems, Man, and Cybernetics: Systems*.



Kai Xu is a Professor at the College of Computer, NUDT, where he received his Ph.D. in 2011. He conducted visiting research at Simon Fraser University and Princeton University. His research interests include geometric modeling and shape analysis, especially on data-driven approaches to the problems in those directions, as well as 3D vision and its robotic applications. He has published over 80 research papers, including 20+ SIGGRAPH/TOG papers. He has co-organized several SIGGRAPH Asia courses

and Eurographics STAR tutorials. He serves on the editorial board of ACM Transactions on Graphics, Computer Graphics Forum, Computers & Graphics, and The Visual Computer. He also served as program co-chair of CAD/Graphics 2017, ICVRV 2017 and ISVC 2018, as well as PC member for several prestigious conferences including SIGGRAPH, SIGGRAPH Asia, Eurographics, SGP, PG, etc. His research work can be found in his personal website: www.kevinkaixu.net.