OAAFormer: Robust and Efficient Point Cloud Registration Through Overlapping-Aware Attention in Transformer

Junjie Gao[1]*, Qiujie Dong*[1]*, Ruian Wang*[1]*, Shuangmin Chen*[2]*, Shiqing Xin*[1]*, Changhe Tu*[1]*, Wenping Wang*[3]

[1]Shandong University, [2]Qingdao University of Science and Technology, [3]Texas A&M University

Abstract In the domain of point cloud registration, the coarse-to-fine feature matching paradigm has received significant attention due to its impressive performance. This paradigm involves a two-step process: first, the extraction of multi-level features, and subsequently, the propagation of correspondences from coarse to fine levels. However, this paradigm faces two notable limitations. Firstly, the use of the Dual Softmax operation may promote one-to-one correspondences between superpoints, inadvertently excluding valuable correspondences. Secondly, it is crucial to closely examine the overlapping areas between point clouds, as only correspondences within these regions decisively determine the actual transformation. Considering these issues, we propose *OAAFormer* to enhance correspondence quality. On one hand, we introduce a soft matching mechanism to facilitate the propagation of potentially valuable correspondences from coarse to fine levels. Additionally, we integrate an overlapping region detection module to minimize mismatches to the greatest extent possible. Furthermore, we introduce a region-wise attention module with linear complexity during the fine-level matching phase, designed to enhance the discriminative capabilities of the extracted features. Tests on the challenging 3DLoMatch benchmark demonstrate that our approach leads to a substantial increase of about 7% in the inlier ratio, as well as an enhancement of 2-4% in registration recall. Finally, to accelerate the prediction process, we replace the conventional RANSAC algorithm with the selection of a limited yet representative set of high-confidence correspondences, resulting in a 100x speedup while still maintaining comparable registration performance.

Keywords point cloud registration, coarse-to-fine, overlapping region, feature matching, transformer

1 Introduction

The task of point cloud registration involves determining a rigid transformation that aligns one point cloud with another. This challenge is of fundamental importance in the fields of computer vision and robotics and has wide-ranging applications, including 3D reconstruction[1–4], SLAM[5–8], and autonomous driving[9–12]. A common approach to this task involves two key stages: point feature matching and globally consistent refinement. During the point feature matching phase, the goal is to generate a set of initial correspondences with a high inlier ratio, ideally including as many true correspondences as possible while minimizing false ones. However, achieving this objective is a formidable challenge due to inherent noise and disparities in the input point clouds, as well as the possibility of partial overlap between them. Conversely, in the globally consistent refinement step, the focus shifts to rapidly identifying a subset of correspondences capable of consistently encoding the actual transformation through further refinement.

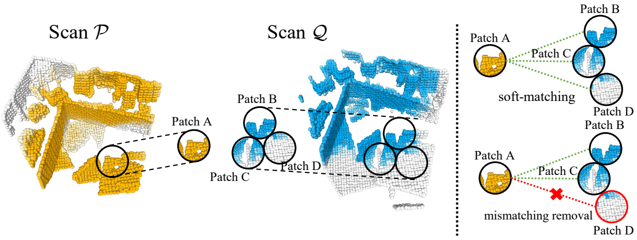


Fig.1. Considering that in coarse-level matching, the correspondence between a source superpoint and a target superpoint inherently embodies a patch-based mapping, there exists the possibility of overlooking potentially valuable correspondences due to the use of the Dual Softmax operation[13, 14]. To ameliorate this concern, we introduce a soft matching mechanism that permits one-to-many correspondences, effectively addressing this limitation. Moreover, our network incorporates a dedicated module for predicting overlap regions, which serves the purpose of filtering out significantly unhelpful correspondences. It is noted that the intensity of the color (yellow or blue) indicates the overlapping score.

While a substantial body of literature[15–19] has focused on the extraction of discriminative features to enhance correspondence quality, the inherent sparsity and disparities in point clouds, along with potential partial overlap, present persistent challenges. Recently, the coarse-to-fine matching paradigm[14, 20] has garnered significant attention for its impressive performance. This paradigm begins by downsampling the input point cloud into superpoints and establishing correspondences between these superpoints, where each superpoint inherently represents a point patch. Subsequently, sparse correspondences are propagated to encompass more points, resulting in the generation of dense correspondences.

However, accurately matching a superpoint from one scan to another can be challenging, as the corresponding point patches may not exhibit perfect alignment. As illustrated in Fig. 1, suppose we have two input point clouds $\mathcal{P}$ and $\mathcal{Q}$. The superpoint A is associated with B, C, D simultaneously. Yet, the use of the Dual Softmax operation[13, 14] within the coarse-to-fine paradigm has the potential to enforce one-to-one correspondences between superpoints, unintentionally excluding valuable correspondences. This represents the first limitation of the coarse-to-fine paradigm. On the other hand, it is crucial to examine the overlapping regions between point clouds, as only correspondences within these areas decisively determine the actual transformation. Consequently, there is a pressing need to enhance the discriminability of the features extracted from points within these overlapping regions to improve the overall performance of the coarse-to-fine paradigm.

Motivated by these considerations, we propose a robust matching network, named *OAAFormer*, with the explicit objective of augmenting the performance of the coarse-to-fine matching paradigm. This augmentation is achieved through the systematic integration of a suite of strategies meticulously designed to elevate the quality of correspondences. Firstly, OAAFormer employs a sophisticated soft matching mechanism, with the explicit purpose of seamlessly propagating potentially valuable correspondences from the coarse to the fine levels of the matching process. Secondly, OAAFormer incorporates an intricately designed overlapping region detection module, strategically engineered to minimize the probability of mismatches. Thirdly, it introduces a region-wise attention module characterized by linear computational complexity, meticulously designed to enhance the discriminative capabilities of the extracted features during the fine-level matching phase. Empirical validation underscores the efficacy of these strategies. For instance, tests on the exacting 3DLoMatch benchmark show that our approach yields a substantial increase of approximately 7% in the inlier ratio, as well as a discernible enhancement of 2-4% in registration recall. Furthermore, we replace the conventional RANSAC algorithm[21] with the selection of a limited yet representative set of high-confidence correspondences for accelerating the prediction process.

In summary, the main contributions of this work are as follows:

- We use a soft matching mechanism to facilitate the propagation of potentially valuable correspondences from coarse to fine levels, which finally results in a substantial increase in the inlier ratio and registration recall.
- We introduce a region-wise attention module with linear complexity during the fine-level matching phase, designed to enhance the discriminative capabilities of the extracted features.
- Through the replacement of the inefficient RANSAC algorithm with a more intelligent mechanism for selecting high-confidence correspondences, we achieve a remarkable 100x acceleration in the prediction process.

2 Related work

2.1 Point cloud registration

The construction of feature descriptors with specific characteristics proves to be an effective means of encoding the curvature of the underlying surface, providing valuable information for the alignment of point clouds. In previous research, a multitude of traditional methods[22–26] have relied on handcrafted features to craft such descriptors. With the proliferation of deep learning techniques, various learning-based descriptors[27–34] have been introduced to enhance the expressiveness of these feature descriptors. However, the task of identifying valuable correspondences between points based solely on geometric descriptors remains a challenging one, primarily due to the presence of various defects in the input point clouds, including noise, disparities, and partial overlapping. Consequently, approaches such as the Random Sample Consensus (RANSAC) algorithm[21, 35, 36] or meticulously designed neural networks[37–40] are frequently employed to address this challenge. These methods aim to eliminate mismatches, even when dealing with points possessing similar features, ultimately resulting in a more robust and accurate registration outcome.

Additionally, a variety of keypoint detectors tailored for rigid registration tasks have emerged. For instance, D3Feat[17] introduces a keypoint selection strategy that overcomes the inherent density variations of 3D point clouds. However, this approach does not fully account for overlapping areas and exhibits reduced robustness in scenarios with low overlap. Another noteworthy method, Predator[19], develops an overlap-attention block for early information exchange between the latent encodings of the two point clouds. Keypoints are selected based on both saliency and overlap scores. While Predator[19] demonstrates substantial improvements over existing methods across indoor and outdoor benchmarks, challenges persist in extracting a set of representative keypoints.

Recently, the coarse-to-fine paradigm has garnered attention for enhancing the quality of correspondences, not only in 2D image matching[41–44] but also in the domain of point cloud registration[14, 20]. For instance, CofiNet[20] incorporates an optimal transport[45, 46] matching layer to establish correspondences between mutually nearest patches and subsequently refines these correspondences at the fine-level stage. In a similar vein, Geotransformer[14] introduces a self-attention mechanism to learn geometric features, thereby improving the matching accuracy between superpoints based on whether their neighboring patches overlap.

In this paper, we further enhance the coarse-to-fine mechanism through a set of strategies, including (1) a soft matching mechanism that streamlines the propagation of potentially valuable correspondences from coarse to fine levels and (2) a region-wise attention module characterized by linear complexity during the fine-level matching phase.

2.2 Efficient Transformer

In the standard Transformer model[47], the memory cost exhibits a quadratic increase due to matrix multiplication, which has become a bottleneck when handling long sequences. Recently, several efficient Transformer variants[48–52] have been introduced. For example, the Linear Transformer[48] reformulates self-attention as a linear dot product of kernel feature maps and exploits the associativity property of matrix products to reduce computational complexity. BigBird[51] combines local and global attention mechanisms at specific positions and introduces random attention for selected token pairs. FastFormer[52] employs an additive attention mechanism to model global contexts, achiev-

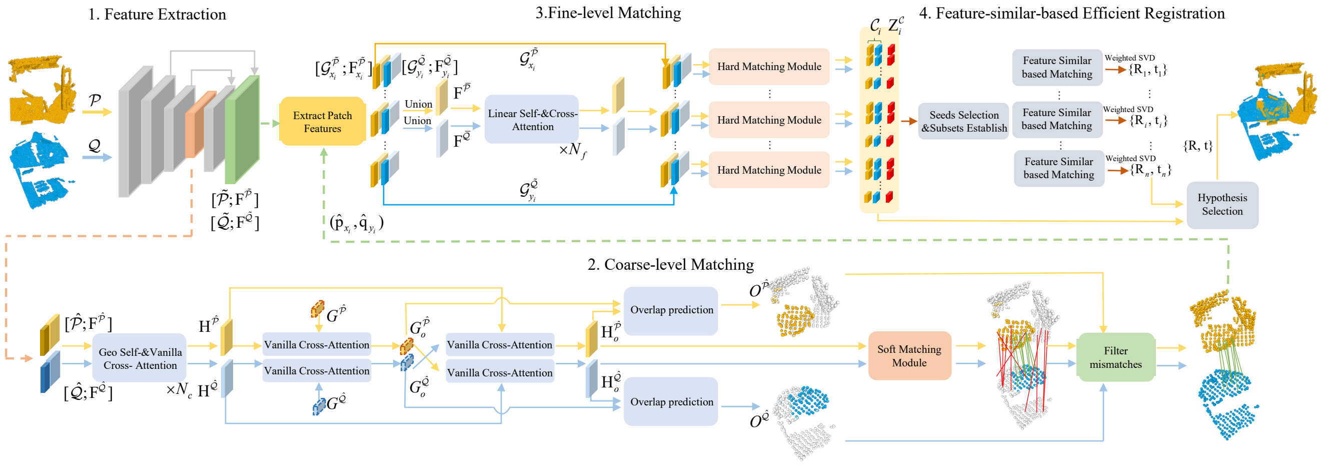

1. Feature Extraction
$\mathcal{P}$
$\mathcal{Q}$
$[\tilde{\mathcal{P}};\mathrm{F}^{\tilde{\mathcal{P}}}]$
$[\tilde{\mathcal{Q}};\mathrm{F}^{\tilde{\mathcal{Q}}}]$
Extract Patch Features
$(\hat{\mathrm{p}}_{x_i},\hat{\mathrm{q}}_{y_i})$
3.Fine-level Matching
$[\mathcal{G}^{\tilde{\mathcal{P}}}_{x_i};\mathrm{F}^{\tilde{\mathcal{P}}}_{x_i}]$
$[\mathcal{G}^{\tilde{\mathcal{Q}}}_{y_i};\mathrm{F}^{\tilde{\mathcal{Q}}}_{y_i}]$
Union
Union
$\mathrm{F}^{\tilde{\mathcal{P}}}$
$\mathrm{F}^{\tilde{\mathcal{Q}}}$
Linear Self-&Cross-Attention
$\times N_f$
$\mathcal{G}^{\tilde{\mathcal{P}}}_{x_i}$
$\mathcal{G}^{\tilde{\mathcal{Q}}}_{y_i}$
Hard Matching Module
Hard Matching Module
Hard Matching Module
$\mathcal{C}_i$ $Z^{\mathcal{C}}_i$
4. Feature-similar-based Efficient Registration
Seeds Selection &Subsets Establish
Feature Similar based Matching
Feature Similar based Matching
Feature Similar based Matching
Weighted SVD
Weighted SVD
Weighted SVD
$\{\mathrm{R}_1,\mathrm{t}_1\}$
$\{\mathrm{R}_i,\mathrm{t}_i\}$
$\{\mathrm{R}_n,\mathrm{t}_n\}$
$\{\mathrm{R},\mathrm{t}\}$
Hypothesis Selection
2. Coarse-level Matching
$[\hat{\mathcal{P}};\mathrm{F}^{\hat{\mathcal{P}}}]$
$[\hat{\mathcal{Q}};\mathrm{F}^{\hat{\mathcal{Q}}}]$
Geo Self-&Vanilla Cross- Attention
$\times N_c$
$\mathrm{H}^{\hat{\mathcal{P}}}$
$\mathrm{H}^{\hat{\mathcal{Q}}}$
$G^{\hat{\mathcal{P}}}$
$G^{\hat{\mathcal{Q}}}$
Vanilla Cross-Attention
Vanilla Cross-Attention
$G^{\hat{\mathcal{P}}}_o$
$G^{\hat{\mathcal{Q}}}_o$
Vanilla Cross-Attention
Vanilla Cross-Attention
$\mathrm{H}^{\hat{\mathcal{P}}}_o$
$\mathrm{H}^{\hat{\mathcal{Q}}}_o$
Overlap prediction
Overlap prediction
$O^{\hat{\mathcal{P}}}$
$O^{\hat{\mathcal{Q}}}$
Soft Matching Module
Filter mismatches

Fig. 2. The backbone network down-samples the input point cloud to extract features in multiple resolutions. In the coarse-level matching step, a soft matching mechanism is employed to establish one-to-many correspondences between superpoints, while the overlapping detection module is introduced to eliminate mismatches outside the overlapping region. In the fine-level matching step, the correspondences between superpoints propagate to the dense point sets $\tilde{\mathcal{P}}$ and $\tilde{\mathcal{Q}}$, and the matching capability of features is enhanced through linear attention modules. Ultimately, the transformation estimation is calculated using an efficient estimation module based on feature similarity.

ing effective context modeling with linear complexity. Inspired by these advancements, we propose a region-wise attention module with linear complexity during the fine-level matching phase, meticulously designed to enhance the discriminative capabilities of the extracted features for points within overlapping areas.

3 Method

3.1 Pipeline

Suppose that we have a source point cloud $\mathcal{P} = \{\mathbf{p}_i \in \mathbb{R}^3 \mid i = 1, \ldots, N\}$ and a target point cloud $\mathcal{Q} = \{\mathbf{q}_i \in \mathbb{R}^3 \mid i = 1, \ldots, M\}$. The objective of rigid registration is to estimate the unknown rigid transformation $\mathbf{T} = \{\mathbf{R},\ \mathbf{t}\}$, where $\mathbf{R} \in \mathrm{SO}(3)$ represents a rotation matrix and $\mathbf{t} \in \mathbb{R}^3$ represents a translation vector. Let

$$\mathcal{C}^* = \{\mathbf{p}_{i_k} \mapsto \mathbf{q}_{j_k}, k = 1, 2, \cdots, K\}$$

denote the set of ground-truth correspondences between $\mathcal{P}$ and $\mathcal{Q}$. The true transformation $\mathbf{T}$ should accurately map each $\mathbf{p}_{i_k} \in \mathcal{P}$ to $\mathbf{q}_{j_k} \in \mathcal{Q}$, meaning that it should minimize the difference vector

$$\mathbf{R}\mathbf{p}_{i_k} + \mathbf{t} - \mathbf{q}_{j_k}$$

to be nearly zero. In real-world scenarios, where the elusive optimal correspondences set $\mathcal{C}^*$ is challenging to obtain, the prevalent approach involves extracting a subset of correspondences that are deemed reasonably reliable between two point clouds. Subsequently, the estimation of the transformation matrix relies on the consistency of these correspondences.

As shown in Fig. 2, our algorithmic pipeline includes the following stages:

(1) During the feature extraction stage, we utilize KPConv [53] as the backbone to downsample the point cloud and extract multi-level features. Subsequently, we select sample points from both the first and last levels for the subsequent matching process.

(2) In the coarse-level matching stage, we utilize the Geotransformer[14] to generate the geometric features of the superpoints. Additionally, we estimate the overlap region using a dedicated detection module specifically designed for this purpose. Refer to Section 3.2.

(3) Subsequently, we introduce a soft matching mechanism to extract valuable correspondences at the patch level, followed by a filtering step to remove po-

tential mismatches. Refer to Section 3.2.

(4) In the fine-level matching stage, we introduce a region-wise attention module characterized by linear complexity. This module is designed to enhance the discriminative capabilities of the extracted features. Refer to Section 3.3.

(5) In the pose estimation stage, we have devised an efficient seeding mechanism for the identification of high-confidence correspondences, aiming to expedite the process. Refer to Section 3.4.

3.2 Coarse-level Matching

Intra- and Inter- Consistency Enhancement: In the coarse-level matching phase, considering superpoints $\hat{\mathcal{P}}$ and $\hat{\mathcal{Q}}$ with associated features $\mathbf{F}^{\hat{\mathcal{P}}} \in \mathbb{R}^{|\hat{\mathcal{P}}| \times d_t}$ and $\mathbf{F}^{\hat{\mathcal{Q}}} \in \mathbb{R}^{|\hat{\mathcal{Q}}| \times d_t}$, we alternately apply the self-attention layer within each point cloud and the cross-attention layer between point clouds N_c times to enhance the consistency. It's worth noting that we utilize the geometry-aware self-attention mechanism[14] instead of the standard self-attention[47], as the former is better suited for capturing long-range contextual information.

Overlap Region Detection Module: To enhance the distinction between overlapping and non-overlapping regions, we introduce a token-based attention mechanism. Specifically, we employ a feature token, denoted as $G^{\hat{\mathcal{P}}}$, to encapsulate information related to the overlapping region. The initialization of $G^{\hat{\mathcal{P}}}$ is accomplished through a max-pooling operation applied to the augmented feature set $\mathbf{H}^{\hat{\mathcal{P}}}$. Subsequently, we employ a cross-attention operation to update the token $G^{\hat{\mathcal{P}}}$, resulting in $G_o^{\hat{\mathcal{P}}}$. This updated token is instrumental in distinguishing between overlapping and non-overlapping regions. In the implementation, the query originates from the initialized token $G^{\hat{\mathcal{P}}}$, while both keys and values are derived from the feature set $\mathbf{H}^{\hat{\mathcal{P}}}$. Finally, the tokens obtained, namely $G_o^{\hat{\mathcal{P}}}$ and $G_o^{\hat{\mathcal{Q}}}$, serve as guiding elements for updating the original features $\mathbf{H}^{\hat{\mathcal{P}}}$ and $\mathbf{H}^{\hat{\mathcal{Q}}}$ through an additional cross-attention operation. This is formally represented as:

$$G_o^{\hat{\mathcal{P}}} = \text{vanillaTransformer}(Q = G^{\hat{\mathcal{P}}}, K = V = \mathbf{H}^{\hat{\mathcal{P}}}), \tag{1}$$

where $G_o^{\hat{\mathcal{Q}}}$ is computed in the same way.

Subsequently, the obtained tokens $G_o^{\hat{\mathcal{P}}}$ and $G_o^{\hat{\mathcal{Q}}}$ are used as guide items to update the original features $\mathbf{H}^{\hat{\mathcal{P}}}$ and $\mathbf{H}^{\hat{\mathcal{Q}}}$ through another cross-attention operation:

$$\mathbf{H}_o^{\hat{\mathcal{P}}} = \text{vanillaTransformer}(Q = \mathbf{H}^{\hat{\mathcal{P}}}, K = V = G_o^{\hat{\mathcal{Q}}}), \tag{2}$$

where $\mathbf{H}_o^{\hat{\mathcal{Q}}}$ is computed in the same way. During this process, $\mathbf{H}^{\hat{\mathcal{P}}}$ and $\mathbf{H}^{\hat{\mathcal{Q}}}$ are updated to $\mathbf{H}_{\text{o}}^{\hat{\mathcal{P}}}$ and $\mathbf{H}_{\text{o}}^{\hat{\mathcal{Q}}}$, respectively, such that they are aware of the overlapping region between $\hat{\mathcal{P}}$ and $\hat{\mathcal{Q}}$. The overlapping-aware mechanism is highly advantageous as it enhances the ability to effectively discriminate between the overlapping region and the non-overlapping region.

To further identify the location of the overlapping region, we have devised an additional module designed to assign a probability score indicating the likelihood that a point is situated within the overlap region. Specifically, we project the decoded tokens $G_{\text{o}}^{\hat{\mathcal{P}}}$ and $G_{\text{o}}^{\hat{\mathcal{Q}}}$ through matrix multiplication and the sigmoid function to create the weight mapping. The weight map $w^{\hat{\mathcal{P}}}$ is employed to enhance the overlap information within the features. Subsequently, a linear projection operator $\mathbf{W}^O \in \mathbb{R}^{d_t \times 1}$, and a sigmoid function are applied to obtain the overlapping confidence:

$$w^{\hat{\mathcal{P}}} = \text{sigmoid}((\mathbf{H}_{\text{o}}^{\hat{\mathcal{P}}})^T G_{\text{o}}^{\hat{\mathcal{P}}}), \tag{3}$$

$$O^{\hat{\mathcal{P}}} = \text{sigmoid}((w^{\hat{\mathcal{P}}} \odot \mathbf{H}_{\text{o}}^{\hat{\mathcal{P}}} + \mathbf{H}_{\text{o}}^{\hat{\mathcal{P}}})\mathbf{W}^O), \tag{4}$$

where $O^{\hat{\mathcal{Q}}}$ is then computed in the same way. To this end, we consider the points whose confidence is greater than a threshold θ_o to be within the overlap region.

Soft-Matching Module: For the output features $\mathbf{H}_o^{\hat{\mathcal{P}}}$ and $\mathbf{H}_o^{\hat{\mathcal{Q}}}$ generated by the overlapping region detection module, we first normalize them to the unit hypersphere. Subsequently, we calculate the similarity matrix $\mathbf{S} \in \mathbb{R}^{|\hat{\mathcal{P}}| \times |\hat{\mathcal{Q}}|}$, where each element is defined as $s_{i,j} = \exp\left(-\left\|\mathbf{h}_i^{\hat{\mathcal{P}}} - \mathbf{h}_j^{\hat{\mathcal{Q}}}\right\|_2^2\right)$. Accordingly, we apply the softmax operation to the similarity matrix $\mathbf{S}$ on two dimensions separately to allow one-to-many matching. Next, we extract purified correspondences by applying a threshold θ_m.

$$\mathbf{S}_k = \mathrm{softmax}(\mathbf{S}(i, \cdot))_j, \tag{5}$$

$$\hat{\mathcal{C}}_k = \{(\hat{\mathbf{p}}_i, \hat{\mathbf{q}}_j) | \mathbf{S}_k(i,j) \geq \theta_m \|\} \tag{6}$$

where $k \in 0, 1$, $\mathbf{S}_0$ and $\mathbf{S}_1$ are the matching probability matrix obtained by softmax operation along the first dimension and the zeroth dimension, $\hat{\mathcal{C}}_0$ and $\hat{\mathcal{C}}_1$ are the corresponding coarse-level correspondences proposals. Compared with the commonly used top-k selection strategy that needs to specify the number of matches, our strategy of using a tolerance can ensure that the number of selected correspondences is adaptive to the overlapping rate.

It is important to acknowledge that while the previously mentioned strategy generates a larger number of potentially beneficial correspondences, it may lead to a low inlier ratio. To enhance this inlier ratio, we introduce a procedure where, for each superpoint in the source point cloud, we initially identify the most closely matched target superpoint based on $\mathbf{S}$, as well as the k-nearest neighbors of the target superpoint. Out of these $k+1$ correspondences, only those that satisfy the condition defined in Eq. (5) are retained. Similarly, for each superpoint in the target point cloud, this process is repeated until a pruned correspondence set $\hat{\mathcal{C}}_k$ is obtained. Finally, we further filter out mismatches outside predicted overlap regions $O^{\hat{\mathcal{P}}}$ and $O^{\hat{\mathcal{Q}}}$.

3.3 Fine-level Matching

Linear Transformer: Linear Transformer[50] proposes to reduce the computation complexity by substituting the exponential kernel used in the original attention layer[47] with an alternative kernel function:

$$\mathrm{sim}(Q, K) = \phi(Q) \cdot \phi(K)^T, \tag{7}$$

where $\phi(\cdot) = elu(\cdot) + 1$. Utilizing the associativity property of matrix products, the multiplication between $\phi(K)^T$ and V can be carried out first. Since $d_t \ll |\mathcal{P}|$, the computation cost is reduced to $O(|\mathcal{P}|)$.

Thanks to our overlap region detection module, we perform linear attention operations to improve feature discrimination only for points within the overlap region and not for all dense points. This reduces the impact of points in non-overlapping region on the one hand, and reduces the cost of calculation on the other hand. To be specific, we only focused on the points $\bar{\mathcal{P}}$ within patch $\{\mathcal{G}_{\mathbf{p}_i}^{\tilde{\mathcal{P}}} | \mathbf{p}_i \in O^{\hat{\mathcal{P}}}\}$ instead of all dense points $\tilde{\mathcal{P}}$, and the relevant features note as $\mathbf{F}^{\bar{\mathcal{P}}}$. We perform the same operation to get overlapping region points $\bar{\mathcal{Q}}$ and relevant features $\mathbf{F}^{\bar{\mathcal{Q}}}$.

Next, we adopt the linear transformer[50] to perform the self- and cross-attention to collect the global information through intra- and inter-relationship between features $\mathbf{F}^{\bar{\mathcal{P}}}$ and $\mathbf{F}^{\bar{\mathcal{Q}}}$. The self-attention layer updates its message by:

$$\mathbf{Z}^{\bar{\mathcal{P}}} = \mathrm{LinearTransformer}(Q = K = V = \mathbf{F}^{\bar{\mathcal{P}}}), \tag{8}$$

and for $\mathbf{Z}^{\bar{\mathcal{Q}}}, Q = K = V = \mathbf{F}^{\bar{\mathcal{Q}}}$. The cross-attention layer updates messages with information collected from the inter-relationship between two frame features:

$$\mathbf{Z}^{\bar{\mathcal{P}}} = \mathrm{LinearTransformer}(Q = \mathbf{F}^{\bar{\mathcal{P}}}, K = V = \mathbf{F}^{\bar{\mathcal{Q}}}) \tag{9}$$

and for $\mathbf{Z}^{\bar{\mathcal{Q}}}, Q = \mathbf{F}^{\bar{\mathcal{Q}}}, K = V = \mathbf{F}^{\bar{\mathcal{P}}}$.

Relative Position Embedding: Unlike the previous work, which either chooses to reduce the point cloud resolution[54, 55] to decrease the computing overhead of the transformer or only aims to enhance the feature representation capability of superpoints[14, 54], our approach introduces a linear transformer[48] to augment the fine-level features. To improve the rotation invariance of the features, inspired by the work of Lepard[54], we integrate rotation-invariant information by adding rotation position embeddings[56] to the inputs at each transformer layer. This helps mitigate limitations on rotation datasets. For further details, please refer to the Appendix.

Hard-Matching Module: Through the aforementioned operations, we obtain a series of one-to-many superpoint correspondences situated in overlapping regions. The associated patches may have a low overlap rate, inevitably leading to a large number of dense point mismatches. Therefore, different from the soft matching strategy in Section 3.2, adopting a stricter matching strategy to suppress mismatches at the fine level is the key to obtaining robust registration. Hence, we employ the point matching module[14], which operates in conjunction with the optimal transmission strategy[45], to extract dense correspondences. The resultant correspondence set is denoted as $\mathcal{C}$. Additionally, the confidence score of $\mathcal{C}$ is denoted as $Z^{\mathcal{C}}$. For further details, please refer to the Appendix.

3.4 Feature-similar-based Efficient Registration

In robust pose estimators such as RANSAC[21], a large number of iterations is typically required to guarantee accuracy, leading to inefficiency. Considering the high inlier ratio of OAAFormer, we have designed an efficient estimator to achieve comparable performance while significantly reducing the computational cost. This design is motivated by the crucial observation that a well-distributed set of correspondences, which are more similar in the feature space, is beneficial for transform estimation.

Global sampling strategy: In order to obtain the global sampling distribution, we employ the spectral matching technique[57] to select reliable seeds. Correspondences with a local maximum confidence score within their neighborhood with radius R are then chosen. The number of seed points N_s is determined by the proportion of the whole correspondences $|\mathcal{C}|$. For each seed, we select its k-nearest neighbors in $Z^{\mathcal{C}}$ to expand into a consensus set. The total consensus sets can be noted as: $\mathcal{CS} \in \mathbb{R}^{N_s \times k}$.

Feature similarity compatibility: We conducted further analysis on the feature similarity of correspondences within each consensus set. The intra-difference of each correspondence in a consensus set is denoted as $\mathbf{D^F} \in \mathbb{R}^{k\times 1}$, and subsequently normalized as: $\mathbf{D^F} = 1 - \mathbf{D^F}/\max(\mathbf{D^F})$. Moreover, we employ a sigmoid operation to expand the inter-difference of correspondences as follows:

$$\mathbf{D^F} = \mathrm{sigmoid}((\mathbf{D^F} - \mathrm{mean}(\mathbf{D^F}))\cdot \sigma_s) \tag{10}$$

where σ_s is a parameter controlling the sensitivity to differences in features. Simultaneously, $\mathbf{D^F}$ serves as a feature similarity score. The closer the correspondence features are, the closer the score is to 1; otherwise, it approaches 0. Subsequently, we compute the compatibility matrix of this consensus set, denoted as $\mathbf{CM} \in \mathbb{R}^{k\times k}$, where each element of $\mathbf{CM}$ represents the minimum value of the two correspondence scores.

Hypothesis Selection: The association of each correspondence with the leading eigenvector is adopted as the weight for this correspondence and can be solved by power iteration algorithm[58]. Then we use the weighted SVD[59] on the consensus set to generate an estimation $(\mathbf{R}_i, \mathbf{t}_i)$ for each seed. Finally, we choose the transformation that allows the most correspondences in $\mathcal{C}$:

$$\mathbf{R},\mathbf{t} = \max_{\mathbf{R}_i,\mathbf{t}_i} \sum\nolimits_{(\tilde{\mathbf{p}}_j,\tilde{\mathbf{q}}_j)\in\mathcal{C}} [\![\|\mathbf{R}_i \cdot \tilde{\mathbf{p}}_j + \mathbf{t}_i - \tilde{\mathbf{q}}_j\|_2^2 < \tau_a]\!] \tag{11}$$

where $[\![\cdot]\!]$ is the Iverson bracket. τ_a is the acceptance radius.

3.5 Loss function:

The final loss consists of the coarse-fine-level loss and the overlap loss: $\mathcal{L} = \mathcal{L}_c + \mathcal{L}_f + 0.5 * \mathcal{L}_o$. As with geotransformer[14], we use overlap-aware circle loss[14] $\mathcal{L}_c$ and negative log-likelihood loss[60] $\mathcal{L}_f$ for coarse and fine level features, respectively. This also benefits us in allowing features to be closer between superpoints/patches with higher overlap ratios in coarse-level matching, rather than strictly limiting one-to-one matching. At the fine-level, stricter supervise can also help eliminate mismatches. Here, the overlap region estimation is regarded as a binary classification task, and the overlap loss $\mathcal{L}_o = \left(\mathcal{L}_o^{\hat{\mathcal{P}}} + \mathcal{L}_o^{\hat{\mathcal{Q}}}\right)/2$ is defined as:

$$\mathcal{L}_o^{\hat{\mathcal{P}}} = \frac{1}{|\hat{\mathcal{P}}|} \sum_{i=1}^{|\hat{\mathcal{P}}|} \bar{o}_{\hat{\mathbf{p}}_i} \log\left(o_{\hat{\mathbf{p}}_i}\right) + \left(1 - \bar{o}_{\hat{\mathbf{p}}_i}\right) \log\left(1 - o_{\hat{\mathbf{p}}_i}\right). \tag{12}$$

The ground truth label $\bar{o}_{\hat{\mathbf{p}}_i}$ of superpoint $\hat{\mathbf{p}}_i$ is defined according whether it is in the ground-truth coarse matches set $\mathcal{A}$:

$$\bar{o}_{\hat{\mathbf{p}}_i} = \begin{cases} 1, & \text{if } i \in \mathcal{A}(x,\cdot) \\ 0, & \text{otherwise} \end{cases} \tag{13}$$

The reverse loss $\mathcal{L}_o^{\hat{\mathcal{Q}}}$ and ground truth label $\bar{o}_{\hat{\mathbf{p}}_i}$ are computed in the same way.

Table 1. Evaluation results on 3DMatch and 3DLoMatch

# Samples=5,000	3DMatch		3DLoMatch	
	Origin	Rotated	Origin	Rotated
Feature Matching Recall (%)				
SpinNet[30]	97.4	97.4	75.5	75.2
Predator[19]	96.6	96.2	78.6	73.7
CoFiNet[20]	98.1	97.4	83.1	78.6
YOHO[32]	98.2	97.8	79.4	77.8
RIGA[34]	97.9	98.2	85.1	84.5
Lepard[53]	98.0	97.4	83.1	79.5
GeoTrans[14]	97.9	97.8	88.3	85.8
Ours	**98.6**	**98.2**	**89.8**	**89.5**
Inlier Ratio (%)				
SpinNet[30]	48.5	48.7	25.7	25.7
Predator[19]	58.0	52.8	26.7	22.4
CoFiNet[20]	49.8	46.8	24.4	21.5
YOHO[32]	64.4	64.1	25.9	23.2
RIGA[34]	68.4	68.5	32.1	32.1
Lepard[53]	58.6	53.7	28.4	24.4
GeoTrans[14]	71.9	68.2	43.5	40.0
Ours	**82.9**	**79.6**	**50.1**	**48.2**
Registration Recall (%)				
SpinNet[30]	88.8	93.2	58.2	61.8
Predator[19]	89.0	92.0	59.8	58.6
CoFiNet[20]	89.3	92.0	67.5	62.5
YOHO[32]	90.8	92.5	65.2	66.8
RIGA[34]	89.3	93.0	65.1	66.9
Lepard[53]	91.7	84.9	62.5	49.0
GeoTrans[14]	92.0	92.0	75.0	71.8
Ours	**94.2**	**93.8**	**77.2**	**76.0**

4 Experiments

In this section, we evaluate OAAFormer on indoor 3DMatch/3DLoMatch benchmarks (Section 4.1), the outdoor KITTI odometry benchmark (Section 4.2), and synthetic ModelNet/ModelLoNet benchmarks (Section 4.3). For the coarse-level matching module, we repeatedly alternate between the geometric self-attention module[14] and the vanilla cross-attention module[47] by setting $N_c = 3$ and then pass through the overlap region detection module. Regarding the threshold θ_m, we observed that $\theta_m = 0.05$ is safe to limit the number of superpoint matches to be within the range of $[256, 512]$. For k nearest neighbors, we find that $k = 3$ achieves the best results. For fine-level matching, we also interleave the linear self-/cross-attention module by setting $N_f = 3$ to enhance feature discrimination. For the proposed efficient pose estimator, $\sigma_s = 10$ is used to enhance the distinctiveness of correspondences, with $k = 20$ for establishing the minimum consensus set, and the number of seeds N_s set to 30% of the total sampled correspondence count. For specific experimental details and network architecture, please refer to the Appendix.

4.1 Indoor Benchmark: 3DMatch

Dataset. 3DMatch[27] is a collection of 62 scenes, of which we employ 46 scenes for training, 8 for validation, and 8 for testing. We utilize the training data preprocessed by [19] and conduct evaluations on both the 3DMatch and 3DLoMatch benchmarks. The former features a 30% overlap, while the latter exhibits low overlap in the range of 10% to 30%. To assess robustness to arbitrary rotations, we follow [32] to create rotated benchmarks, where full-range rotations are independently applied to the two frames of each point cloud pair.

Metrics. We follow [14, 19, 20] to employ three metrics for evaluation: (1) *Inlier Ratio* (**IR**), which computes the ratio of putative correspondences with a residual distance smaller than a threshold (*i.e.*, 0.1m) under the ground-truth transformation; (2) *Feature Matching Recall* (**FMR**), which calculates the fraction of point cloud pairs with an **IR** exceeding a threshold (*i.e.*, 5%); and (3) *Registration Recall* (**RR**) , which quantifies the fraction of point cloud pairs that are accurately registered (*i.e.*, with a root mean square error, **RMSE** <0.2m).

Table 2. Evaluation results on 3DMatch and 3DLoMatch with a varying number of correspondences

# Samples	3DMatch					3DLoMatch				
	5000	2500	1000	500	250	5000	2500	1000	500	250
Feature Matching Recall (%)										
PMatch[28]	95.0	94.3	92.9	90.1	82.9	63.6	61.7	53.6	45.2	34.2
FCGF[29]	97.4	97.3	97.0	96.7	96.6	76.6	75.4	74.2	71.7	67.3
D3Feat[17]	95.6	95.4	94.5	94.1	93.1	67.3	66.7	67.0	66.7	66.5
SpinNet[30]	97.6	97.2	96.8	95.5	94.3	75.3	74.9	72.5	70.0	63.6
Predator[19]	96.6	96.6	96.5	96.3	96.5	78.6	77.4	76.3	75.7	75.3
YOHO[32]	98.2	97.6	97.5	97.7	96.0	79.4	78.1	76.3	73.8	69.1
CoFiNet[20]	98.1	98.3	98.1	98.2	98.3	83.1	83.5	83.3	83.1	82.6
GeoTrans[14]	97.9	97.9	97.9	97.9	97.6	88.3	88.6	88.8	88.6	88.3
Ours	**98.6**	**98.6**	**98.5**	**98.5**	**98.2**	**89.8**	**89.9**	**90.1**	**90.1**	**89.9**
Inlier Ratio (%)										
PMatch[28]	36.0	32.5	26.4	21.5	16.4	11.4	10.1	8.0	6.4	4.8
FCGF[29]	56.8	54.1	48.7	42.5	34.1	21.4	20.0	17.2	14.8	11.6
D3Feat[17]	39.0	38.8	40.4	41.5	41.8	13.2	13.1	14.0	14.6	15.0
SpinNet[30]	47.5	44.7	39.4	33.9	27.6	20.5	19.0	16.3	13.8	11.1
Predator[19]	58.0	58.4	57.1	54.1	49.3	26.7	28.1	28.3	27.5	25.8
YOHO[32]	64.4	60.7	55.7	46.4	41.2	25.9	23.3	22.6	18.2	15.0
CoFiNet[20]	49.8	51.2	51.9	52.2	52.2	24.4	25.9	26.7	26.8	26.9
GeoTrans[14]	71.9	75.2	76.0	82.2	85.1	43.5	45.3	46.2	52.9	57.7
Ours	**82.9**	**83.1**	**83.3**	**85.5**	**86.1**	**50.1**	**52.4**	**55.6**	**58.6**	**60.1**
Registration Recall (%)										
PMatch[28]	78.4	76.2	71.4	67.6	50.8	33.0	29.0	23.3	17.0	11.0
FCGF[29]	85.1	84.7	83.3	81.6	71.4	40.1	41.7	38.2	35.4	26.8
D3Feat[17]	81.6	84.5	83.4	82.4	77.9	37.2	42.7	46.9	43.8	39.1
SpinNet[30]	88.6	86.6	85.5	83.5	70.2	59.8	54.9	48.3	39.8	26.8
Predator[19]	89.0	89.9	90.6	88.5	86.6	59.8	61.2	62.4	60.8	58.1
YOHO[32]	90.8	90.3	89.1	88.6	84.5	65.2	65.5	63.2	56.5	48.0
CoFiNet[20]	89.3	88.9	88.4	87.4	87.0	67.5	66.2	64.2	63.1	61.0
GeoTrans[14]	92.0	91.8	91.8	91.4	91.2	75.0	74.8	74.2	74.1	73.5
Ours	**94.2**	**94.2**	**93.8**	**93.2**	**93.0**	**77.2**	**77.2**	**77.0**	**76.8**	**76.4**

Correspondence results. We begin by comparing the results of OAAFormer with the recent state-of-the-art in Table 1, and then proceed to analyze the impact of varying the number of correspondences in Table 2 and Table 3. Notably, our method excels in terms of **FMR**, outperforming all baselines significantly, particularly in the case of 3DLoMatch. This implies a substantial increase in the likelihood of achieving correct registration with our robust pose estimator in low-overlap scenarios, where we consistently find more than 5% inliers. Furthermore, for **IR**, our approach exhibits even more substantial improvements, surpassing all benchmarks by over 10% on 3DMatch and more than 7% on 3DLoMatch. It is worth mentioning that our method maintains a stable performance even when the number of correspondences varies. Additionally, due to our incorporation of rotational invariance position information during fine-level matching, we perform admirably on the rotated datasets.

Registration results. As Table 1 shows, the primary metric related to the ultimate objective of point cloud registration is **RR**. For this metric, we compute the transformation using RANSAC[21] with 50K iterations. OAAFormer excels in terms of **RR**, outperforming the competition with significant margins. Specifically, we achieve improvements of 2.2% on both the standard and rotated datasets for 3DMatch and even more remarkable enhancements of 2.2% and 4.2% on 3DLoMatch.

Additionally, we report registration recall under different numbers of correspondences in Table 2 and Table 3. It's evident that our method's performance is remarkably stable, eliminating the need for extensive correspondence sampling, as seen in previous methods aimed at performance improvement.

We then compare **RR** using RANSAC-free estimators in Table 4 . We begin with weighted SVD[59] over correspondences to solve for the alignment transformation. Thanks to high values of **FMR** and **IR**, OAAFormer achieves **RR** scores of 88.4% and 62.1% on 3DMatch and 3DLoMatch, respectively, while the results of the baseline methods deteriorate significantly. This can be explained by the fact that, on one hand, the coarse-to-fine mechanism constrains the correspondences to specific patches rather than the global domain. On the other hand, our model further narrows down the correspondences to the overlapping region and enhances the discriminative capabilities of fine-level features.

Subsequently, we employ the local-to-global registration module (LGR) [14] and (FSR) in Section 3.3 separately to compute the transformation. In comparison with LGR, the FSR in our method maintains a similar time cost but significantly improves the sampling distribution, making it more effective for transformation estimation and yielding higher **RR** scores. This efficient estimator delivers performance on par with the robust pose estimator (RANSAC) but with significantly lower time costs, offering over 100 times acceleration.

Table 3. Evaluation results on Rotated 3DMatch and 3DLoMatch with a varying number of correspondences

# Samples	3DMatch(Rotated)					3DLoMatch(Rotated)				
	5000	2500	1000	500	250	5000	2500	1000	500	250
Feature Matching Recall (%)										
SpinNet[30]	97.4	97.4	96.7	96.5	94.1	75.2	74.9	72.6	69.2	61.8
Predator[19]	96.2	96.2	96.6	96.0	96.0	73.7	74.2	75.0	74.8	73.5
YOHO[32]	97.8	97.8	97.4	97.6	96.4	77.8	77.8	76.3	73.9	67.3
CoFiNet[20]	97.4	97.4	97.2	97.2	97.3	78.6	78.8	79.2	78.9	79.2
GeoTrans[14]	97.8	97.9	98.1	97.7	97.3	85.8	85.7	86.5	86.6	86.1
Ours	**98.2**	**98.2**	**98.2**	**98.1**	**98.1**	**89.8**	**89.6**	**89.6**	**89.4**	**89.2**
Inlier Ratio (%)										
SpinNet[30]	48.7	46.0	40.6	35.1	29.0	25.7	23.9	20.8	17.9	15.6
Predator[19]	52.8	53.4	52.5	50.0	45.6	22.4	23.5	23.0	23.2	21.6
YOHO[32]	64.1	60.4	53.5	46.3	36.9	23.2	23.2	19.2	15.7	12.1
CoFiNet[20]	46.8	48.2	49.0	49.3	49.3	21.5	22.8	23.6	23.8	23.8
GeoTrans[14]	68.2	72.5	73.3	79.5	82.3	40.0	40.3	42.7	49.5	54.1
Ours	**82.9**	**82.9**	**83.3**	**83.3**	**83.5**	**48.2**	**48.5**	**50.4**	**52.3**	**54.6**
Registration Recall (%)										
SpinNet[30]	93.2	93.2	91.1	87.4	77.0	61.8	59.1	53.1	44.1	30.7
Predator[19]	92.0	92.8	92.0	92.2	89.5	58.6	59.5	60.4	58.6	55.8
YOHO[32]	92.5	92.3	92.4	90.2	87.4	66.8	67.1	64.5	58.2	44.8
CoFiNet[20]	92.0	91.4	91.0	90.3	89.6	62.5	60.9	60.9	59.9	56.5
GeoTrans[14]	92.0	91.9	91.8	91.5	91.4	71.8	72.0	72.0	71.6	70.9
Ours	**93.8**	**93.8**	**93.6**	**93.6**	**93.2**	**76.0**	**75.4**	**75.4**	**75.3**	**74.9**

Table 4. Registration results w/o RANSAC on 3DMatch (3DM) and 3DLoMatch (3DLM). The time overhead for transformation estimation is also provided

Model	Estimator	#Samples	RR(%)		Time(s)
			3DM	3DLM	Pose
SpinNet[30]	RANSAC-50k	5000	88.6	59.8	2.344
Predator[19]	RANSAC-50k	5000	89.0	59.8	
CoFiNet[20]	RANSAC-50k	5000	89.3	67.5	
GeoTrans[14]	RANSAC-50k	5000	92.0	75.0	
Ours	RANSAC-50k	5000	**94.2**	**77.2**	
SpinNet[30]	weighted SVD	250	34.0	2.5	0.008
Predator[19]	weighted SVD	250	50.0	6.4	
CoFiNet[20]	weighted SVD	250	64.6	21.6	
GeoTrans[14]	weighted SVD	250	86.5	59.9	
Ours	weighted SVD	250	**88.4**	**62.1**	
CoFiNet[20]	LGR	5000	85.5	63.2	0.019
GeoTrans[14]	LGR	5000	91.2	73.4	
Ours	LGR	5000	**93.2**	**76.2**	
CoFiNet[20]	FSR	5000	85.8	64.2	0.022
GeoTrans[14]	FSR	5000	91.5	73.8	
Ours	FSR	5000	**93.4**	**76.8**	

Ablation studies. To gain a more comprehensive understanding of the individual modules within our method, we conducted a series of ablation studies. Following the methodology outlined in [14], we introduced the metric *Patch Inlier Ratio* (**PIR**) to measure the fraction of patch matches with actual overlap. Additionally, we introduced another metric, *Patch Overlap Precision* (**POP**), to assess the precision of patches within the actual overlap. It's worth noting that the metrics **FMR** and **IR** were reported with correspondences in the set $\mathcal{C}$, while RANSAC[21] was employed for the registration process.

To investigate the effectiveness of the overlap detection module (ODM), we compared it with the MLP-directly[19] module (MLP) in Table 5. Leveraging the attention mechanism, our module has the capability to model the global overlap position, allowing for better perception of the overlap region. With a well-designed re-weighted prediction module, we obtained more accurate detection results for the overlap region. As accurate overlap estimation is pivotal for eliminating mismatches, our proposed module outperforms alternatives across all metrics.

Moving forward, to explore the interactions between the soft-matching module (SMM), overlapping detection module (ODM), and linear transformer module (LTM), we conducted relevant ablation experiments in Table 6. When all modules were removed, OAAFormer reverted to Geotransformer[14] and served as the baseline. In general, when we replaced the strict matching mechanism of the original implementation with SMM, due to the introduction of a one-to-many matching paradigm, while introducing a prior for local-to-local matching, some mismatches were inevitably introduced, resulting in a decline in all metrics. The introduction of ODM and LTM, on the other hand, enhanced the accuracy of coarse- and fine-level matching, respectively, and outperformed the original implementation. When all three modules were introduced simultaneously, SMM mined more potential patch matches, ODM eliminated mismatches distributed outside the estimated overlapping regions, and LTM made the dense features of the overlapping region more discriminative, achieving the best performance.

To better elucidate the impact of each module, we present qualitative results of coarse/fine-level correspondences under different module ablations, as delineated in Fig. 3. (a) showcases the outcomes of Geotransformer[14], which extracts a fixed number of coarse and dense correspondences. (b) illustrates the outcomes when solely the SMM module is incorporated. Due to the introduction of one-to-many matching and adaptive threshold settings, more matches are established at the coarse-level matching stage, inevitably introducing some outliers that propagate to the fine-level matching stage. However, more inliers are fortunately discerned at this stage. (c) demonstrates that introducing the ODM module can preserve inliers while predominantly eliminating outliers introduced by the SMM module. In comparison to Geotransformer[14], which samples a fixed number of coarse correspondences, we can adaptively sample fewer correspondences in low-overlap scenes, thereby enhancing PIR and diminishing unnecessary interference in the fine-level matching stage. (d) illustrates that introducing the LTM module notably enhances the feature matching capability in the overlapping regions, thereby further refining the matching capability at the fine-level matching stage.

In addition, we replaced the relative position embedding[56] with the absolute position embedding[47] in the linear attention module and conducted relevant ablation experiments. As shown in Table 7, in the context of the rotated version benchmark within the 3DMatch/3DLoMatch dataset, it is evident that the inclusion of relative position embedding resulted in superior performance. This observation suggests that incorporating relative positional information not only assists the neural network in effectively modeling distant spatial relationships but also enhances the network's capacity to discriminate between features within regions that are otherwise similar. Furthermore, it contributes to the augmentation of feature rotation invari-

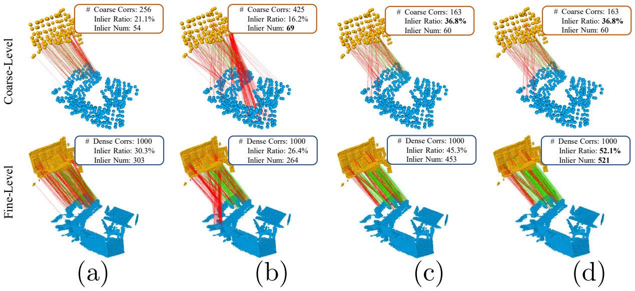


Fig. 3. Qualitative results of coarse-/fine-level correspondences under different module ablations. (a) depicts results of Geotransformer[14]. (b) illustrates results with only SMM. (c) showcases results with SMM+ODM. (d) demonstrates results of the complete model with SMM+ODM+LTM. Green/red lines indicate inliers/outliers.

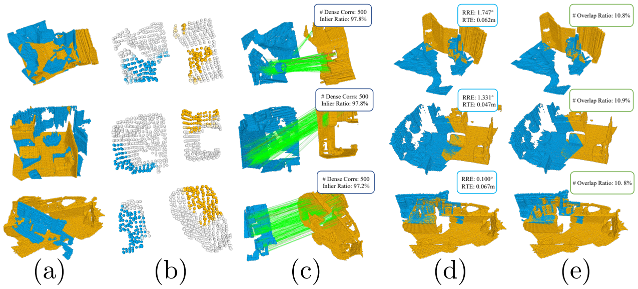


Fig. 4. Qualitative results on 3DLoMatch. (a) illustrates the input point cloud. (b) shows the predicted overlap region. (c) represents the established correspondences. (d) demonstrates the registration results. Green/red lines indicate inliers/outliers.

Table 5. Ablation experiments of the overlapping region detection module

Model	3DMatch						3DLoMatch			
	POP	PIR	FMR	IR	RR	OP	PIR	FMR	IR	RR
MLP[19]	89.6	84.2	98.2	73.4	92.5	84.5	53.4	88.5	45.2	75.5
ODM	**93.5**	**85.6**	**98.6**	**82.9**	**94.2**	**88.1**	**54.2**	**89.8**	**50.1**	**77.2**

Table 6. Ablation experiments of main modules

SMM	ODM	LTM	3DMatch				3DLoMatch			
			PIR	FMR	IR	RR	PIR	FMR	IR	RR
			86.1	97.9	71.9	92.0	54.9	88.3	43.5	75.0
✓			82.7	97.4	68.0	91.3	46.4	86.1	38.1	73.5
	✓		**86.4**	98.1	73.6	92.7	**55.3**	88.7	44.8	75.5
		✓	86.1	98.4	79.2	93.4	54.9	89.3	46.4	75.8
✓	✓	✓	85.6	**98.6**	**82.9**	**94.2**	54.4	**89.8**	**50.1**	**77.2**

Table 7. Ablation experiments of position embedding

Model	3DMatch(Rotated)			3DLoMatch(Rotated)		
	FMR	IR	RR	FMR	IR	RR
Absolute[47]	**98.0**	80.2	93.2	88.4	43.8	75.2
Relative[55]	**98.2**	**82.9**	**93.8**	**89.8**	**48.2**	**76.0**

Qualitative results. Fig. 4 offers a visualization of the overlap region prediction in the coarse level and the dense correspondence results in the fine level. The overlapping region detection module excels in perceiving the global position, and the interaction module aids in determining whether superpoints are situated within the overlap region. Moreover, the Linear Transformer module with the relative position embedding strategy enhances the discriminative ability for dense correspondences, resulting in more reliable correspondences.

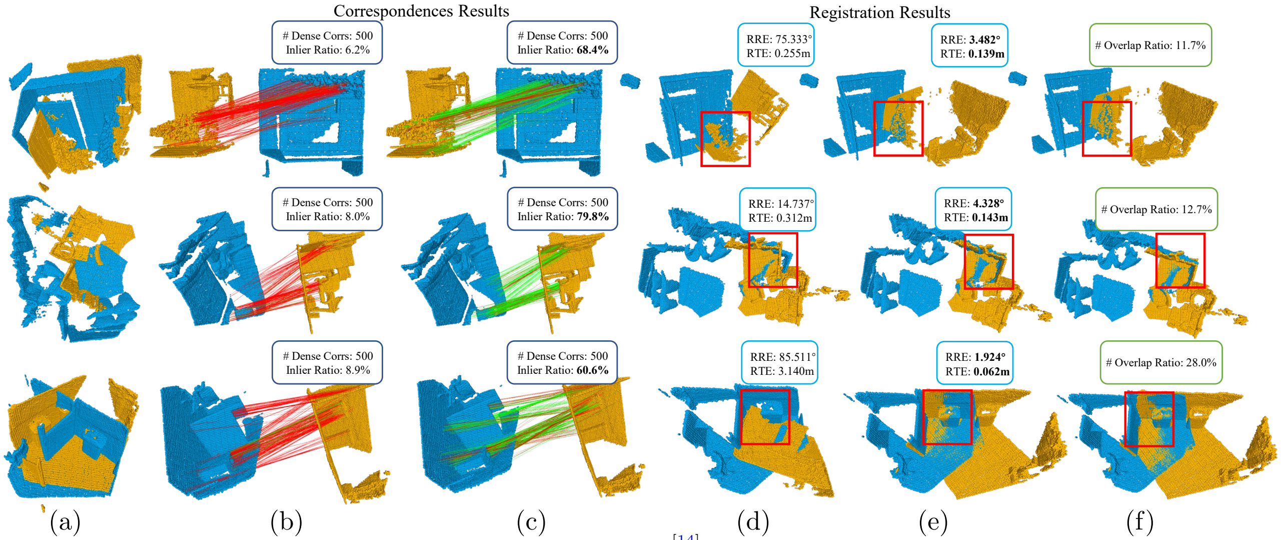


Fig.5. Qualitative comparison results on 3DLoMatch. Geotransformer[14] serves as the baseline. (a) illustrates the input point cloud. (b) and (c) respectively depict correspondence results of Geotransformer[14] and our method. (d) and (e) respectively illustrate registration results of Geotransformer[14] and our method. (f) represents the ground-truth. Green/red lines indicate inliers/outliers.

A gallery of registration and matching comparison results with state-of-the-art methods is shown in Fig. 5. It is evident that our method can establish more accurate correspondences across a broader spectrum of domains, yielding robust registration outcomes.

4.2 Outdoor Benchmark: KITTI

Dataset. The KITTI odometry dataset[60] comprises 11 sequences of LiDAR-scanned outdoor driving scenarios. For training, we adhere to the setup of[17, 37], utilizing sequences 0-5, while sequences 6-7 are reserved for validation, and sequences 8-10 are designated for testing. In line with the approach described in[19], we refine the ground-truth poses using ICP, and restrict the evaluation to point cloud pairs that are within a maximum distance of 10 meters.

Metrics. We adhere to the evaluation metrics established by[17, 19], which include the following: (1) *Relative Rotation Error* (**RRE**): This metric quantifies the geodesic distance between the estimated and ground-truth rotation matrices. (2) *Relative Translation Error* (**RTE**): It calculates the Euclidean distance between the estimated and ground-truth translation vectors. (3) *Registration Recall* (**RR**): This metric measures the fraction of point cloud pairs for which both **RRE** and **RTE** fall below specific thresholds, typically set as **RRE**<5° and **RTE**<2 meters.

Table 8. Registration results on KITTI odometry

Model	RTE(cm)	RRE(°)	RR(%)
3DFeat-Net[15]	25.9	0.25	96.0
FCGF[29]	9.5	0.30	96.6
D3Feat[17]	7.2	0.30	**99.8**
SpinNet[30]	9.9	0.47	99.1
Predator[19]	6.8	0.27	**99.8**
CoFiNet[20]	8.2	0.41	**99.8**
GeoTrans[14]	7.4	0.27	**99.8**
Ours (RANSAC)	**6.6**	**0.24**	**99.8**
FMR[61]	∼66	1.49	90.6
DGR[38]	∼32	0.37	98.7
HRegNet[37]	∼12	0.29	99.7
GeoTrans (LGR)[14]	6.8	0.24	**99.8**
Ours (FSR)	**6.0**	**0.21**	**99.8**

Registration results. In Table 8 (top), we compare OAAFormer with recent state-of-the-art methods, employing RANSAC as the pose estimator: D3Feat[17], SpinNet[30], Predator[19], CoFiNet[20], and Geotransformer[14]. Our method performs comparably to these methods on **RR** but outperforms the baseline by approximately 0.7 cm in terms of **RTE** and 0.03° in **RRE**. We also compare our method to three RANSAC-free methods in Table 8 (bottom): FMR[61], DGR[38], HRegNet[37], and Geotransformer (with LGR)[14]. Our method outperforms all the baselines significantly. Furthermore, when using FSR as an estimator, our method surpasses all the RANSAC-based methods.

4.3 Synthetic Benchmark: ModelNet

Dataset. ModelNet comprises 12,311 CAD models of synthetic objects spanning 40 distinct categories. We adhere to the practice of employing 5,112 samples for training, 1,202 samples for validation, and 1,266 samples for testing. Similar to [19], we conduct evaluations under two partial overlap scenarios: ModelNet, characterized by an average pairwise overlap of 73.5%, and ModelLoNet, which exhibits a lower average overlap of 53.6%.

Metrics. We adhere to the methodology outlined in [19, 54] for performance evaluation, employing three key metrics: (1) **RRE** (2) **RTE** (with definitions consistent with those in Section 4.2), and (3) *Chamfer distance* (**CD**), which quantifies the chamfer distance between two registered scans.

Table 9. Registration results on ModelNet dataset

Model	ModelNet			ModelLoNet		
	RRE	RTE	CD	RRE	RTE	CD
Predator[19]	1.739	0.019	0.00089	5.235	0.132	0.0083
Ours (RANSAC)	**1.484**	**0.016**	**0.00081**	**4.143**	**0.091**	**0.0044**
PointNetLK[63]	29.725	0.297	0.0235	48.567	0.507	0.0367
OMNet[64]	2.947	0.032	0.0015	6.517	0.129	0.0074
DCP-v2[65]	11.975	0.171	0.0117	16.501	0.300	0.0268
RPM-Net[66]	1.712	0.018	0.00085	7.342	0.124	0.0050
REGTR[54]	1.473	0.014	0.00078	3.930	0.087	0.0037
Ours (FSR)	**1.366**	**0.012**	**0.00074**	**3.884**	**0.074**	**0.0032**

Registration results. In Table 9, we conduct a comparative analysis of OAAFormer against state-of-the-art RANSAC-based methods and RANSAC-free methods. Notably, a few RANSAC-free methods are optimized primarily for ModelNet, and these models exhibit rapid performance deterioration in real-world scenarios. In contrast, OAAFormer demonstrates a substantial performance advantage over all baseline methods across all metrics, whether in the context of high overlap (ModelNet) or low overlap (ModelLoNet) scenarios.

5 Conclusions

In this paper, we have enhanced the coarse-to-fine matching mechanism through a series of strategies. Key enhancements include (1) the development of a soft matching module to preserve valuable correspondences among superpoints, (2) the introduction of an overlapping region detection module for the elimination of mismatches and (3) the incorporation of a region-wise attention module with linear complexity to bolster the discriminative capabilities of the extracted features. Furthermore, we propose a technique to accelerate the prediction process by carefully selecting limited but representative correspondences with high-confidence. Our method's effectiveness and robustness are validated through experiments conducted on three publicly available datasets.

Conflict of Interest

The authors declare that they have no conflict of interest.

References

[1] Azinović D, Martin-Brualla R, Goldman D B, Nießner M, Thies J. Neural RGB-D surface reconstruction. In *Proc. the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, June. 2022, pp.6290-6301. DOI: 10.1109/CVPR52688.2022.00619.

[2] Deng K, Liu A, Zhu J, Ramanan D. Depth-supervised NeRF: Fewer Views and Faster Training for Free. In

Proc. the IEEE/CVF Conference on Computer Vision and Pattern Recognition, June. 2022, pp.12882-12891. DOI: 10.1109/CVPR52688.2022.01254.

[3] Li K, Tang Y, Prisacariu V A, Torr P H. BNV-Fusion: Dense 3D Reconstruction using Bi-level Neural Volume Fusion. In *Proc. the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, June. 2022, pp. 6156-6165. DOI: 10.1109/CVPR52688.2022.00607.

[4] Wang Z, Wang P, Wang P, Dong Q, Gao J, Chen S, Xin S, Tu C, Wang W. Neural-IMLS: Self-supervised Implicit Moving Least-Squares Network for Surface Reconstruction. *IEEE Trans. visualization and computer graphics*, 2023, 1(1):1-16. DOI: 10.1109/TVCG.2023.3284233.

[5] Barros A M, Michel M, Moline Y, Corre G, Carrel F. A Comprehensive Survey of Visual SLAM Algorithms. *Robotics*, 2022, 11(1):24. DOI: 10.3390/robotics11010024.

[6] Chaplot D S, Gandhi D, Gupta S, Gupta A K, Salakhutdinov, R. Learning to Explore using Active Neural SLAM. arXiv:2004.05155, 2020. https://arxiv.org/abs/2004.05155, Apr. 2020.

[7] Mur-Artal R, Montiel J M, Tardós J D. ORB-SLAM: A Versatile and Accurate Monocular SLAM System. *IEEE Trans. Robotics*. 2015, 31(5):1147-1163. DOI: 10.1109/TRO.2015.2463671.

[8] Teed Z, Deng J. DROID-SLAM: Deep Visual SLAM for Monocular, Stereo, and RGB-D Cameras. In *Proc. the 35th Conference on Neural Information Processing Systems*, Dec. 2021, pp.16558-16569.

[9] Chitta K, Prakash A, Geiger A. NEAT: Neural Attention Fields for End-to-End Autonomous Driving. In *Proc. the IEEE/CVF International Conference on Computer Vision*, Oct. 2021, pp.15773-15783. DOI: 10.1109/ICCV48922.2021.01550.

[10] Hu S, Chen L, Wu P, Li H, Yan J, Tao D. ST-P3: End-to-end Vision-based Autonomous Driving via Spatial-Temporal Feature Learning. In *Proc. the European Conference on Computer Vision*, Oct. 2022. pp.533-549.

[11] Prakash A, Chitta K, Geiger A. Multi-Modal Fusion Transformer for End-to-End Autonomous Driving. In *Proc. the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, June. 2021, pp.7073-7083. DOI: 10.1109/CVPR46437.2021.00700.

[12] Yang Z, Chai Y, Anguelov D, Zhou Y, Sun P, Erhan D, Rafferty S M, Kretzschmar H. SurfelGAN: Synthesizing Realistic Sensor Data for Autonomous Driving. In *Proc. the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, June. 2020, pp.11115-11124. DOI: 10.1109/CVPR42600.2020.01113.

[13] Cheng X, Lin H, Wu X, Yang F, Shen D. Improving Video-Text Retrieval by Multi-Stream Corpus Alignment and Dual Softmax Loss. arXiv:2109.04290, 2021. https://arxiv.org/abs/2109.04290, Nov. 2021.

[14] Qin Z, Yu H, Wang C, Guo Y, Peng Y, Xu K. Geometric Transformer for Fast and Robust Point Cloud Registration. In *Proc. the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, Feb. 2022, pp.11133-11142. DOI: 10.1109/CVPR52688.2022.01086.

[15] Yew Z J, Lee G H. 3DFeat-Net: Weakly Supervised Local 3D Features for Point Cloud Registration. In *Proc. the European conference on computer vision*, Sep. 2018. pp.607-623. DOI: 10.1007/978-3-030-01267-0-37.

[16] Li J, Lee G H. USIP: Unsupervised Stable Interest Point Detection From 3D Point Clouds. In *Proc. the IEEE/CVF International Conference on Computer Vision*, Oct. 2019. pp.361-370. DOI: 10.1109/ICCV.2019.00045.

[17] Bai X, Luo Z, Zhou L, Fu H, Quan L, Tai C. D3Feat: Joint Learning of Dense Detection and Description of 3D Local Features. In *Proc. the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, June. 2020, pp.6358-6366. DOI: 10.1109/CVPR42600.2020.00639.

[18] Wu W, Zhang Y, Wang D, Lei Y. SK-Net: Deep Learning on Point Cloud via End-to-End Discovery of Spatial Keypoints. In *Proc. the AAAI Conference on Artificial Intelligence*. Feb. 2020, pp.6422-6429. DOI: 10.1609/aaai.v34i04.6113.

[19] Huang S, Gojcic Z, Usvyatsov M, Wieser A, Schindler K. PREDATOR: Registration of 3D Point Clouds with Low Overlap. In *Proc. the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, Jun. 2020, pp.4265-4274. DOI: 10.1109/CVPR46437.2021.00425.

[20] Yu H, Li F, Saleh M, Busam B, Ilic S. CoFiNet: Reliable Coarse-to-fine Correspondences for Robust Point Cloud Registration. In *Proc. the Neural Information Processing Systems*, Dec. 2021, pp.23872-23884.

[21] Fischler M A, Bolles, R C. Random sample consensus: a paradigm for model fitting with applications to im-

age analysis and automated cartography.*ACM. Communications*, 1981, 24(6): 381-395. DOI: 10.1016/B978-0-08-051581-6.50070-2.

[22] Johnson A E, Hebert M. Using Spin Images for Efficient Object Recognition in Cluttered 3D Scenes. *IEEE Trans. pattern analysis and machine intelligence*, 1999, 21(5): 433-449. DOI: 10.1109/34.765655.

[23] Tombari F, Salti S, Stefano L D. Unique shape context for 3d data description. In *Proc. the ACM workshop on 3D object retrieval* , Oct. 2010, pp.57. DOI: 10.1145/1877808.1877821.

[24] Tombari F, Salti S, Stefano L D. Unique Signatures of Histograms for Local Surface Description. In *Proc. the European Conference on Computer Vision*, Sep. 2010, pp.356-369. DOI: 10.1007/978-3-642-15558-1-26.

[25] Rusu R B, Blodow N, Beetz M. Fast Point Feature Histograms (FPFH) for 3D registration. In *Proc. the IEEE International Conference on Robotics and Automation*, May. 2009, pp.3212-3217. DOI: 10.1109/ROBOT.2009.5152473.

[26] Guo Y, Bennamoun S F, Wan J, Lu M. 3D free form object recognition using rotational projection statistics. In *Proc. the IEEE Workshop on Applications of Computer Vision*, Jan. 2013, PP.1-8. DOI: 10.1109/WACV.2013.6474992.

[27] Zeng A, Song, S, Nießner M, Fisher M, Xiao J, Funkhouser T A. 3DMatch: Learning Local Geometric Descriptors from RGB-D Reconstructions. In *Proc. the IEEE Conference on Computer Vision and Pattern Recognition*, Jul. 2016, pp.199-208. DOI: 10.1109/CVPR.2017.29.

[28] Gojcic Z, Zhou C, Wegner J D, Wieser A. The Perfect Match: 3D Point Cloud Matching With Smoothed Densities. In *Proc. the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, Jun. 2019, pp.5540-5549. DOI: 10.1109/CVPR.2019.00569.

[29] Choy C B, Park J, Koltun V. Fully Convolutional Geometric Features. In *Proc. the IEEE/CVF International Conference on Computer Vision*, Jun. 2019, pp.8957-8965. DOI: 10.1109/CVPR.2019.00569.

[30] Ao S, Hu Q, Yang B, Markham A, Guo Y. SpinNet: Learning a General Surface Descriptor for 3D Point Cloud Registration. In *Proc. the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, Jun. 2021, pp.11748-11757. DOI: 10.1109/CVPR46437.2021.01158.

[31] Thomas H, Qi C, Deschaud J, Marcotegui B, Goulette F, Guibas L J. KPConv: Flexible and Deformable Convolution for Point Clouds. In *Proc. the IEEE/CVF International Conference on Computer Vision* , Oct. 2019, pp.6410-6419. DOI: 10.1109/ICCV.2019.00651.

[32] Wang H, Liu Y, Dong Z, Wang W, Yang B. You Only Hypothesize Once: Point Cloud Registration with Rotation-equivariant Descriptors. In *Proc. the 30th ACM International Conference on Multimedia*, Oct. 2022, pp.1630-1641. DOI: 10.1145/3503161.3548023.

[33] Wang H, Liu Y, Hu Q, Wang B, Chen J, Dong Z, Guo Y, Wang W, Yang B. RoReg: Pairwise Point Cloud Registration With Oriented Descriptors and Local Rotations. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 2023, 45(8):10376-10393. DOI: 10.1109/TPAMI.2023.3244951.

[34] Yu H, Hou J, Qin Z, Saleh M, Shugurov I S, Wang K, Busam B, Ilic S. RIGA: Rotation-Invariant and Globally-Aware Descriptors for Point Cloud Registration. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 2024, 1(1):1-17. DOI: 10.1109/TPAMI.2023.3349199.

[35] Myatt D R, Torr P H, Nasuto S J, Bishop J M, Craddock R. (2002). NAPSAC: High Noise, High Dimensional Robust Estimation - it's in the Bag. British Machine Vision Conference. In *Proc. the British Machine Vision Conference*, Sep. 2022, pp.1-10. DOI: 10.5244/C.16.44.

[36] Baráth D, Matas J. Graph-Cut RANSAC. In *Proc. the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, Jun. 2018, pp.6733-6741. DOI: 10.1109/CVPR.2018.00704.

[37] Pais G D, Ramalingam S, Govindu V M, Nascimento J C, Chellappa R, Miraldo P. 3DRegNet: A Deep Neural Network for 3D Point Registration. In *Proc. the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, Jun. 2020, pp.7191-7201. DOI: 10.1109/CVPR42600.2020.00722.

[38] Choy C B, Dong W, Koltun V. Deep Global Registration. In *Proc. the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, Jun. 2020, pp.2511-2520. DOI: 10.1109/CVPR42600.2020.00259.

[39] Bai X, Luo Z, Zhou L, Chen H, Li L, Hu Z, Fu H, Tai C. PointDSC: Robust Point Cloud Registration using Deep Spatial Consistency. In *Proc. the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, Jun. 2021, pp.15854-15864. DOI: 10.1109/CVPR46437.2021.01560.

[40] Chen Z, Sun K, Yang F, Tao W. (2022). SC2-PCR: A Second Order Spatial Compatibility for Efficient and Robust Point Cloud Registration. In *Proc. the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, Jun. 2022, pp.13211-13221. DOI: 10.1109/CVPR52688.2022.01287.

[41] Li X, Han K, Li S, Prisacariu V. Dual-resolution correspondence networks. In *Proc. the 34th International Conference on Neural Information Processing*, Dec. 2020, pp.17346-17357.

[42] Zhou Q, Sattler T, Leal-Taixé L. Patch2Pix: Epipolar-Guided Pixel-Level Correspondences. *Proc. the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, Jun. 2020, pp.4667-4676. DOI: 10.1109/CVPR46437.2021.00464.

[43] Sun J, Shen Z, Wang Y, Bao H, Zhou X. LoFTR: Detector-Free Local Feature Matching with Transformers. *Proc. the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, Jun. 2021, pp.8918-8927. DOI: 10.1109/CVPR46437.2021.00881.

[44] Huang D, Chen Y, Xu S, Liu Y, Wu W, Ding Y, Wang C, Tang F. Adaptive Assignment for Geometry Aware Local Feature Matching. *Proc. the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, Jun. 2022. pp.5425-5434. DOI: 10.1109/CVPR52729.2023.00525.

[45] Cuturi M. Sinkhorn distances: Lightspeed computation of optimal transport. *Proc. the 26th International Conference on Neural Information Processing Systems*, Dec. 2013, pp.2292-2300.

[46] Peyré G, Cuturi M. Computational optimal transport: With applications to data science. *Foundations and Trends in Machine Learning*, 2019, 11(5-6):355-607. DOI: 10.1561/2200000073.

[47] Vaswani A, Shazeer N, Parmar N, Uszkoreit J, Jones L, Gomez A N, Polosukhin I. Attention is all you need. *Proc. the 31st International Conference on Neural Information Processing Systems*, Dec. 2017, pp.6000-6010.

[48] Katharopoulos A, Vyas A, Pappas N, Fleuret F. Transformers are rnns: Fast autoregressive transformers with linear attention. *Proc. the International conference on machine learning*, Nov. 2020, pp.5156-5165.

[49] Shen Z, Zhang M, Zhao H, Yi S, Li H. Efficient attention: Attention with linear complexities. *Proc. the IEEE/CVF winter conference on applications of computer vision*, Jan. 2021, pp.3531-3539. DOI: 10.1109/WACV48630.2021.00357.

[50] Wang S, Li B Z, Khabsa M, Fang H, Ma H. Linformer: Self-attention with linear complexity. arXiv:2006.04768, 2020. https://arxiv.org/abs/2006.04768, Jun. 2020.

[51] Zaheer M, Guruganesh G, Dubey K A, Ainslie J, Alberti C, Ontanon S, Ahmed A. Big bird: Transformers for longer sequences. *Proc. the 34th International Conference on Neural Information Processing*, Dec. 2020, pp.17283-17297.

[52] Wu C, Wu F, Qi T, et al. Fastformer: Additive attention can be all you need. arXiv:2108.09084, 2021. https://arxiv.org/abs/2108.09084, Sep. 2021.

[53] Thomas H, Qi C R, Deschaud J E, Marcotegui B, Goulette F, Guibas L J. Kpconv: Flexible and deformable convolution for point clouds. *Proc. the IEEE/CVF international conference on computer vision*, Oct 2019, pp.6411-6420. DOI: 10.1109/ICCV.2019.00651.

[54] Li Y, Harada T. Lepard: Learning partial point cloud matching in rigid and deformable scenes. *Proc. the IEEE/CVF conference on computer vision and pattern recognition*, Jun. 2022, pp.5554-5564. DOI: 10.1109/CVPR52688.2022.00547.

[55] Yew Z J, Lee G H. Regtr: End-to-end point cloud correspondences with transformers. *Proc. the IEEE/CVF conference on computer vision and pattern recognition*, Jun. 2022, pp.6677-6686. DOI: 10.1109/CVPR52688.2022.00656.

[56] Su J, Ahmed M, Lu Y, Pan S, Bo W, Liu Y. Roformer: Enhanced transformer with rotary position embedding. arXiv:2104.09864, 2021. https://arxiv.org/abs/2104.09864, Apr. 2021.

[57] Leordeanu M, Hebert M. A spectral technique for correspondence problems using pairwise constraints. *Proc. the IEEE International Conference on Computer Vision*, Oct. 2005, pp.1482-1489. DOI: 10.1109/ICCV.2005.20.

[58] Mises R V, Pollaczek-Geiringer H. Praktische Verfahren der Gleichungsauflosung. *Applied Mathematics and Mechanics/Zeitschrift fur Angewandte Mathematik und Mechanik*, 1929, 9(1):58-77.

[59] Arun K S, Huang T S, Blostein S D. Least-squares fitting of two 3-D point sets. *IEEE Trans. pattern analysis and machine intelligence*, 1987, (5):698-700.

[60] Wang A, Pruksachatkun Y, Nangia N, Singh A, Michael J, Hill F, Bowman S. (2019). Superglue: A stickier benchmark

for general-purpose language understanding systems. *Proc. the 33th International Conference on Neural Information Processing Systems*, Dec. 2019, pp.3266-3280.

[61] Geiger A, Lenz P, Urtasun R. Are we ready for autonomous driving? the kitti vision benchmark suite. *Proc. the IEEE conference on computer vision and pattern recognition*, Jun. 2012, pp.3354-3361. DOI: 10.1109/CVPR.2012.6248074.

[62] Huang X, Mei G, Zhang J. Feature-metric registration: A fast semi-supervised approach for robust point cloud registration without correspondences. *Proc. the IEEE/CVF conference on computer vision and pattern recognition*, Jun. 2020, pp.11366-11374. DOI: 10.1109/CVPR42600.2020.01138.

[63] Li X, Pontes J K, Lucey S. Pointnetlk revisited. *Proc. the IEEE/CVF conference on computer vision and pattern recognition*, Jun. 2021, pp.12763-12772. DOI: 10.1109/CVPR46437.2021.01257.

[64] Xu H, Liu S, Wang G, Liu G, Zeng B. Omnet: Learning overlapping mask for partial-to-partial point cloud registration. *Proc. the IEEE/CVF International Conference on Computer Vision*, Jun. 2021, pp.3132-3141. DOI: 10.1109/ICCV48922.2021.00312.

[65] Wang Y, Solomon J M. Deep closest point: Learning representations for point cloud registration. *Proc. the IEEE/CVF international conference on computer vision*. Oct. 2019, pp.3523-3532. DOI: 10.1109/ICCV.2019.00362.

[66] Yew Z J, Lee G H. Rpm-net: Robust point matching using learned features. *Proc. the IEEE/CVF conference on computer vision and pattern recognition*, Jun 2020, pp.11824-11833. DOI: 10.1109/CVPR42600.2020.01184.

1 APPENDIX

In this appendix, we first provide the implementation details in Section 1.1, then details of the Geometric Structure Embedding and the hard-matching module are illustrated in Section 1.2 and Section 1.3 respectively. Following this, we elaborate on the evaluation metrics in Section 1.4. In addition, the detailed network architecture is shown in Section 1.5. Moreover, more qualitative experimental results are demonstrated in Section 1.6. Finally, the limitations are further discussed in Section 1.7.

1.1 Implementation Details

We implement and evaluate OAAFormer with Pytorch[1] on an NVIDIA RTX 3090 GPU. The network is trained with Adam optimizer and the learning rate starts from 1e-4 and decays exponentially by 0.05 every epoch. We use the matching radius of τ = 10cm for 3DMatch, τ = 60cm for KITTI and τ = 10cm for ModelNet to determine overlapping during the generation of both coarse-level and fine-level ground-truth matches.

In the training stage, we randomly sample 128 ground-truth superpoint matches, and in the inference stage, the related parameters for overlap region and confident threshold are set as $\theta_o = 0.4$ and $\theta_m = 0.05$ respectively. In addition, the upper and lower bounds for selecting superpoint matches are set as $[256, 512]$. In order to obtain a higher quality correspondences set, we vary the hyper-parameter k in the multual top-k selection and set the confidence score as 0.05 of the hard matching module to control the number of the dense correspondences for OAAFormer. *i.e.*, $k = 1$ for 250/500/1000 matches, $k = 2$ for 2500 matches, and $k = 3$ for 5000 matches. And we use top-k selection to sample a certain number of the correspondences to report the performances of our method.

1.2 Geometric Structure Embedding

In [2], the geometric structure embedding encodes the relative distances and angles across superpoints. The details are as follows:

Given the relative distance $\rho_{i,j}$, the pair-wise distance embedding $\mathbf{r}^D_{i,j}$ is computed by the sinusoidal function[3]:

$$\begin{cases} r^D_{i,j,2k} = \sin\left(\dfrac{d_{i,j}/\sigma_d}{10000^{2k/d_t}}\right) \\ r^D_{i,j,2k+1} = \cos\left(\dfrac{d_{i,j}/\sigma_d}{10000^{2k/d_t}}\right) \end{cases} \tag{1}$$

where d_t is the feature dimention, and σ_d is a temperature which controls the sensitivity to distance variations.

The relative angular embedding can be computed in the same way. Given the angle $\alpha^k_{i,j}$, the angular embedding $\mathbf{r}^A_{i,j}$ is computed as:

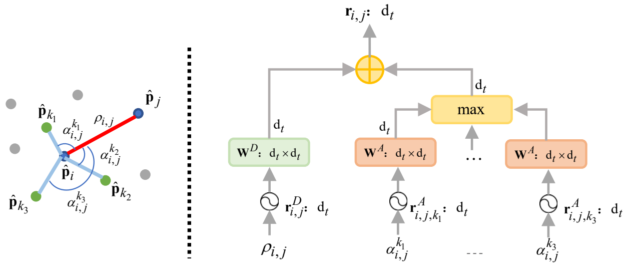
$$\begin{cases} r^A_{i,j,k,2x} = \sin\left(\dfrac{\alpha^k_{i,j}/\sigma_a}{10000^{2x/d_t}}\right) \\ r^A_{i,j,k,2x+1} = \cos\left(\dfrac{\alpha^k_{i,j}/\sigma_a}{10000^{2x/d_t}}\right) \end{cases} \tag{2}$$

where σ_a is another temperature which controls the sensitivity to angular variations.

The geometric embedding $\mathbf{r}_{i,j}$ finally represents as:

$$\mathbf{r}_{i,j} = \mathbf{r}^D_{i,j}\mathbf{W}^D + \max_x \left\{\mathbf{r}^A_{i,j,x}\mathbf{W}^A\right\} \tag{3}$$

where $\mathbf{W}^D$,$\mathbf{W}^A \in \mathbb{R}^{d_t \times d_t}$ are two learnable matrices. Fig. 1 illustrates the computation of geometric structure embedding.


$\mathbf{r}_{i,j}$: d_t
$\hat{\mathbf{p}}_j$
$\hat{\mathbf{p}}_{k_1}$
$\rho_{i,j}$
$\alpha_{i,j}^{k_1}$
$\alpha_{i,j}^{k_2}$
$\hat{\mathbf{p}}_i$
$\hat{\mathbf{p}}_{k_2}$
$\hat{\mathbf{p}}_{k_3}$
$\alpha_{i,j}^{k_3}$
d_t
max
d_t
d_t
$\mathbf{W}^D$: $\mathrm{d}_t \times \mathrm{d}_t$
$\mathbf{W}^A$: $\mathrm{d}_t \times \mathrm{d}_t$
...
$\mathbf{W}^A$: $\mathrm{d}_t \times \mathrm{d}_t$
$\mathbf{r}_{i,j}^D$: d_t
$\mathbf{r}_{i,j,k_1}^A$: d_t
$\mathbf{r}_{i,j,k_3}^A$: d_t
$\rho_{i,j}$
$\alpha_{i,j}^{k_1}$

$\alpha_{i,j}^{k_3}$

Fig.1. An illustration of the distance-and-angle-based geometric structure encoding and its computation

1.3 Hard-Matching Module

Unlike the soft allocation strategy employed during coarse-level matching, applying a stricter matching strategy to suppress mismatches at the fine-level is crucial for obtaining robust registration. Consequently, for each coarse-level match $\hat{\mathcal{C}}_i = (\hat{\mathbf{p}}_{x_i}, \hat{\mathbf{q}}_{y_i})$, we extract local dense point correspondences between patch $\mathcal{G}_{x_i}^{\tilde{\mathcal{P}}}$ and patch $\mathcal{G}_{x_i}^{\tilde{\mathcal{Q}}}$ by an optimal transport layer. To be specific, a coherence matrix $\mathbf{C}_i \in \mathbb{R}^{|\mathcal{G}_{x_i}| \times |\mathcal{G}_{y_i}|}$ is first computed as:

$$\mathbf{C}_i = \mathbf{F}_{x_i}^{\mathcal{P}} \left(\mathbf{F}_{y_i}^{\mathcal{Q}}\right)^T / \sqrt{\tilde{d}} \tag{4}$$

To suppress mismatches, we augment each set with a dustbin so that mismatches are explicitly assigned to it. Specifically, we augment the coherence matrix $\mathbf{C}_i$ to $\bar{\mathbf{C}}_i$ by appending a new row and column, filled with a single learnable parameter α:

$$\bar{\mathbf{C}}_{i,|\mathcal{G}_{x_i}|+1} = \bar{\mathbf{C}}_{|\mathcal{G}_{y_i}|+1,j} = \bar{\mathbf{C}}_{|\mathcal{G}_{x_i}|+1,|\mathcal{G}_{y_i}|+1} = \alpha \in \mathbb{R} \tag{5}$$

Then, we iteratively utilize Sinkhorn algorithm for $\bar{\mathbf{C}}_i$ to compute the assignment matrix $\bar{\mathbf{Z}}_i$ which is then recovered to $\mathbf{Z}_i$ by dropping the last row and the last column. and then, the mutual top-k selection strategy to extract correspondences where selected if the match among the k largest entries for the row and column in $\mathbf{Z}$:

$$\mathcal{C}_i = \left\{\left(\mathcal{G}_{x_i}^{\mathcal{P}}(x_j), \mathcal{G}_{y_i}^{\mathcal{Q}}(y_j)\right) \mid (x_j, y_j) \in \text{mutual\_topk}_{x,y}\left(z_{x,y}^i\right)\right\} \tag{6}$$

and the fine-level correspondences set $\mathcal{C}$ is the collection of each subset $\mathcal{C}_i$: $\mathcal{C} = \bigcup_{i=1}^{N_i} \mathcal{C}_i$. For the convenience of subsequent calculation, the confidence of each $\mathcal{C}_i$ is saved as: $\mathbf{Z}_i^{\mathcal{C}} = \{\mathbf{Z}_i(x,y) | (x,y) \in \mathcal{C}_i\}$, similarly, the confidence $\mathbf{Z}^{\mathcal{C}}$ for all correspondences is noted as: $\mathbf{Z}^{\mathcal{C}} = \bigcup_{i=1}^{N_i} \mathbf{Z}_i^{\mathcal{C}}$.

1.4 Evaluation metrics

Following common practice, we use different evaluation metrices for 3DMatch, KITTI and ModelNet. On 3DMatch, we report *Inlier Ratio, Feature Matching Recall* and *Registration Recall*. Following with [2], We also report *Patch Inlier Ratio* to evaluate the superpoint (patch) correspondences. In addition, another metric *Patch Overlap Precision* is introduced to measure the precision of patches within the actual overlap. On KITTI, we report *Relative Rotation Error, Relative Translation Error* and *Registration Recall*. On ModelNet, we report *Relative Rotation Error, Relative Translation Error* and *Chamfer Distance*.

Relative Rotation Error (**RRE**) that measures the geodesic distance in degrees between estimated and ground-truth rotation matrices. In practice, it measures the differences between the predicted and the ground-truth rotation matrices.

$$\mathbf{RRE} = \arccos\left(\frac{\operatorname{trace}\left(\mathbf{R}^T \cdot \bar{\mathbf{R}} - 1\right)}{2}\right) \tag{7}$$

Relative Translation Error (**RTE**) that measures the Euclidean distance between estimated and ground-truth translation vectors. In practice, it measures the differences between the predicted and the ground-truth translation vectors.

$$\mathbf{RTE} = \|\mathbf{t} - \hat{\mathbf{t}}\|_2 \tag{8}$$

Inlier Ratio (**IR**) that computes the ratio of putative correspondences whose residual distance is smaller than a threshold ($i.e.$, $\tau_1 = 0.1$m) under the ground-truth transformation $\bar{\mathbf{T}}_{\mathbf{P}\rightarrow\mathcal{Q}}$:

$$\mathbf{IR} = \frac{1}{|\mathcal{C}|} \sum_{(\mathbf{p}_{x_i}, \mathbf{q}_{y_i}) \in \mathcal{C}} [\![\left\| \bar{\mathbf{T}}_{\mathbf{P}\rightarrow\mathcal{Q}} \left(\mathbf{p}_{x_i} \right) - \mathbf{q}_{y_i} \right\|_2 < \tau_1]\!] \quad (9)$$

where $[\![\cdot]\!]$ is the Iverson bracket.

Feature Matching Recall (**FMR**)) that calculates the fraction of point cloud pairs whose **IR** is larger than a threshold ($i.e.$, $\tau_2 = 5\%$). **FMR** measures the likelihood of recovering the accurate transformation by using a robust estimator such as RANSAC[4].

$$\mathbf{FMR} = \frac{1}{M} \sum_{i=1}^{M} [\![\mathbf{IR}_i > \tau_2]\!] \quad (10)$$

where M is the number of all point cloud pairs.

Registration Recall (**RR**)) that counts the fraction of point cloud pairs that are correctly registered.

For 3DMatch ($i.e.$, with **RMSE** <0.2m). **RMSE** is the root mean square error of the ground-truth correspondences $\mathcal{C}^*$ after applying the estimated transformation $\bar{\mathbf{T}}_{\mathbf{P}\rightarrow\mathcal{Q}}$:

$$\mathbf{RMSE} = \sqrt{\frac{1}{|\mathcal{C}^*|} \sum_{(\mathbf{p}_{x_i}, \mathbf{q}_{y_4}) \in \mathcal{C}^*} \left\| \mathbf{T}_{\mathcal{P}\rightarrow\mathcal{Q}} \left(\mathbf{p}^*_{x_i} \right) - \mathbf{q}^*_{y_i} \right\|_2^2} \quad (11)$$

$$\mathbf{RR} = \frac{1}{M} \sum_{i=1}^{M} [\![\mathbf{RMSE}_i < 0.2\text{m}]\!] \quad (12)$$

For KITTI ($i.e.$, with **RRE** <5° and **RTE** <2m).

$$\mathbf{RR} = \frac{1}{M} \sum_{i=1}^{M} [\![\mathbf{RRE}_i < 5^\circ \wedge \mathbf{RTE}_i < 2m]\!] \quad (13)$$

Patch Inlier Ratio (**PIR**))) that counts the fraction of superpoint (patch) matches with actual overlap under the ground-truth transformation $\bar{\mathbf{T}}_{\mathbf{P}\rightarrow\mathcal{Q}}$. It reflects the quality of the putative superpoint (patch) correspondences:

$$\mathbf{PIR} = \frac{1}{|\mathcal{C}|} \sum_{(\hat{\mathbf{p}}_{x_i}, \hat{\mathbf{q}}_{y_i}) \in \hat{\mathcal{C}}} [\![\exists \tilde{\mathbf{p}} \in \bar{\mathbf{T}}_{\mathbf{P}\rightarrow\mathcal{Q}}(\mathcal{G}^{\tilde{\mathcal{P}}}_{x_i}), \tilde{\mathbf{q}} \in \mathcal{G}^{\tilde{\mathcal{Q}}}_{y_i} \quad s.t. \|\tilde{\mathbf{p}} - \tilde{\mathbf{q}}\|_2 < \tau]\!] \quad (14)$$

where the matching radius is τ.

Chamfer Distance (**CD**)) that measures the quality of registration on synthetic data. We follow [5] and use the modified Chamfer distance metric.

$$\begin{aligned} \mathbf{CD}(\mathcal{P}, \mathcal{Q}) = & \frac{1}{|\mathcal{P}|} \sum_{\mathbf{p} \in \mathcal{P}} \min_{\mathbf{q} \in \mathcal{Q}} \left\| \bar{\mathbf{T}}_{\mathcal{P}\rightarrow\mathcal{Q}} - \mathbf{q} \right\|_2^2 + \\ & \frac{1}{|\mathcal{Q}|} \sum_{\mathbf{q} \in \mathcal{Q}} \min_{\mathbf{p} \in \mathcal{P}} \left\| \mathbf{q} - \bar{\mathbf{T}}_{\mathcal{P}\rightarrow\mathcal{Q}}(\mathbf{p}) \right\|_2^2 \end{aligned} \quad (15)$$

1.5 Network Architecture

The detailed network configurations are shown in Table 1.

Backbone. We adopt KPConv[6] as a backbone to downsample the point cloud and extract the point-wise features simultaneously. Before being fed into the backbone, the input point clouds are first downsampled with a voxel size of 2.5cm on 3DMatch, 30cm on KITTI and 5cm on ModelNet. The voxel size is then doubled in each down-sampling operation. Since the distribution density of point cloud is different, we use a 4-stage backbone for 3DMatch, a 5-stage backbone for KITTI and 3-stage backbone for ModelNet. The configurations of KPConv are the same as in [7] and the group normalization[8] is used after the KPConv layers.

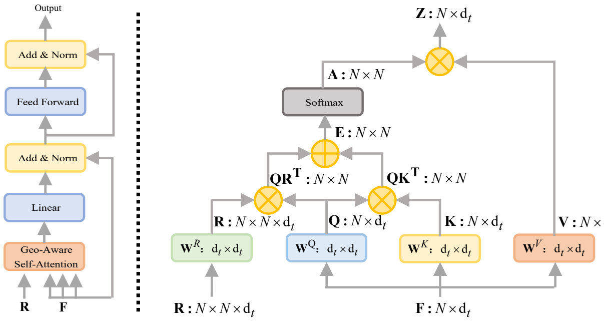


Fig. 2. Left: The structure of geometric self-attention module. Right: The computation graph of geometric self-attention.

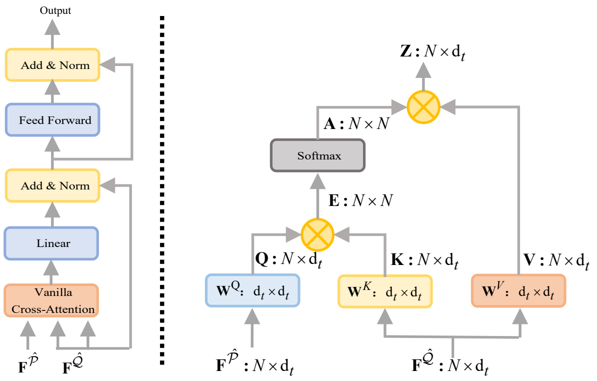

Output
Add & Norm
Feed Forward
Add & Norm
Linear
Vanilla Cross-Attention
$\mathbf{F}^{\hat{\mathcal{P}}}$
$\mathbf{F}^{\hat{\mathcal{Q}}}$
$\mathbf{Z}: N\times d_t$
$\mathbf{A}: N\times N$
Softmax
$\mathbf{E}: N\times N$
$\mathbf{Q}: N\times d_t$
$\mathbf{K}: N\times d_t$
$\mathbf{V}: N\times d_t$
$\mathbf{W}^Q$: $d_t\times d_t$
$\mathbf{W}^K$: $d_t\times d_t$
$\mathbf{W}^V$: $d_t\times d_t$
$\mathbf{F}^{\hat{\mathcal{P}}}: N\times d_t$
$\mathbf{F}^{\hat{\mathcal{Q}}}: N\times d_t$

Fig.3. Left: The structure of feature-based cross-attention module. Right: The computation graph of Vanilla cross-attention.

Coarse-level Matching Module. At the onset of the coarse-level matching module, a linear projection is employed to compress feature dimensions and mitigate memory consumption. For 3DMatch and ModelNet, the feature dimension is $d_t = 256$. Since KITTI has larger number of points, the feature dimension here is set to $d_t = 128$. Then, we repeatedly interchanging the geometry-aware self-attention layer and vanilla cross-attention layer for $N_c = 3$ times. All attention modules have 4 attention heads. In the geometric structure embedding, we use $\sigma_d = 0.2m$ on 3DMatch and ModelNet, and $\sigma_d = 4.8m$ on KITTI, while $\sigma_a = 15°$ on all datasets. The computations of the geometric structure embedding, geometric-aware self-attention and vanilla cross-attention for inputs are shown in Fig. 1, Fig. 2 and Fig. 3. Afterwards, we use another linear projection to to get hybrid features $\mathbf{H}^{\hat{\mathcal{P}}}$ and $\mathbf{H}^{\hat{\mathcal{Q}}}$ with dimension of 256, and then fed into overlapping region detection module. In order to get a suitable overlap region for mismatches filtering, we used $\theta_o = 0.4$ on all datasets. The specific structure is shown in Fig. 4.

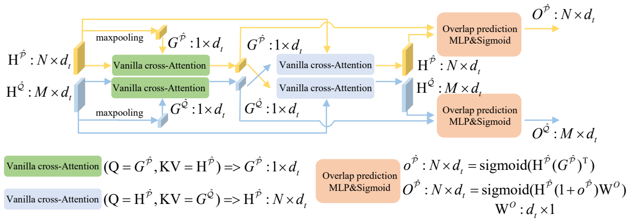

$\mathrm{H}^{\hat{\mathcal{P}}}: N\times d_t$
$\mathrm{H}^{\hat{\mathcal{Q}}}: M\times d_t$
maxpooling
$G^{\hat{\mathcal{P}}}: 1\times d_t$
maxpooling
$G^{\hat{\mathcal{Q}}}: 1\times d_t$
Vanilla cross-Attention
Vanilla cross-Attention
$G^{\hat{\mathcal{P}}}: 1\times d_t$
$G^{\hat{\mathcal{Q}}}: 1\times d_t$
Vanilla cross-Attention
Vanilla cross-Attention
$\mathrm{H}^{\hat{\mathcal{P}}}: N\times d_t$
$\mathrm{H}^{\hat{\mathcal{Q}}}: M\times d_t$
Overlap prediction MLP&Sigmoid
Overlap prediction MLP&Sigmoid
$O^{\hat{\mathcal{P}}}: N\times d_t$
$O^{\hat{\mathcal{Q}}}: M\times d_t$
Vanilla cross-Attention $(\mathrm{Q}=G^{\hat{\mathcal{P}}}, \mathrm{KV}=\mathrm{H}^{\hat{\mathcal{P}}}) => G^{\hat{\mathcal{P}}}: 1\times d_t$
Vanilla cross-Attention $(\mathrm{Q}=\mathrm{H}^{\hat{\mathcal{P}}}, \mathrm{KV}=G^{\hat{\mathcal{Q}}}) => \mathrm{H}^{\hat{\mathcal{P}}}: N\times d_t$
Overlap prediction MLP&Sigmoid
$o^{\hat{\mathcal{P}}}: N\times d_t = \mathrm{sigmoid}(\mathrm{H}^{\hat{\mathcal{P}}}(G^{\hat{\mathcal{P}}})^{\mathrm{T}})$
$O^{\hat{\mathcal{P}}}: N\times d_t = \mathrm{sigmoid}(\mathrm{H}^{\hat{\mathcal{P}}}(1+o^{\hat{\mathcal{P}}})\mathrm{W}^O)$
$\mathrm{W}^O: d_t\times 1$

Fig.4. The structure of overlapping region detection module.

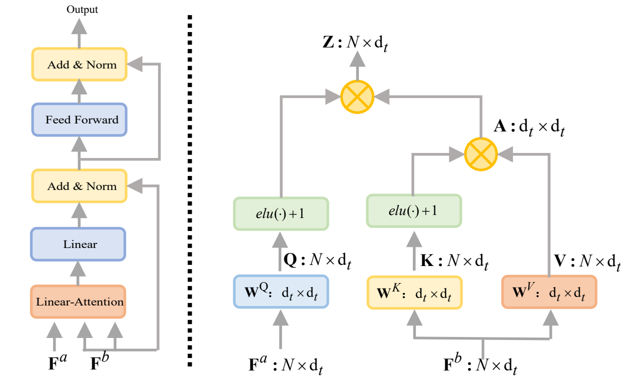

Output
Add & Norm
Feed Forward
Add & Norm
Linear
Linear-Attention
$\mathbf{F}^a$
$\mathbf{F}^b$
$\mathbf{Z}: N\times d_t$
$\mathbf{A}: d_t\times d_t$
$elu(\cdot)+1$
$elu(\cdot)+1$
$\mathbf{Q}: N\times d_t$
$\mathbf{K}: N\times d_t$
$\mathbf{V}: N\times d_t$
$\mathbf{W}^Q$: $d_t\times d_t$
$\mathbf{W}^K$: $d_t\times d_t$
$\mathbf{W}^V$: $d_t\times d_t$
$\mathbf{F}^a: N\times d_t$
$\mathbf{F}^b: N\times d_t$

Fig.5. Left: The structure of feature-based linear-attention module. Right: The computation graph of Linear attention.

Fine-level Matching Module. Through the coarse-level matching module, a series of superpoint correspondences are extracted. However, due to the inevitable partial overlap between two patches caused by the point-to-node grouping strategy. It is also necessary to strengthen the differentiation of fine-level features, which are always neglected by existing work. While the memory cost of vanilla transformer is the second order $O(N^2)$ of sequence length due to matrix multiplication, which results in being impractical in the context of fine-level feature matching. To solve this problem, Linear attention is introduced as an alternative, which reduces the complexity by replacing the exponential kernel of the original attention layer with a kernel function $sim(Q, K) = f(Q)\cdot f(K)^T, f(\cdot) = elu(\cdot)+1$. Utilizing the associativity property of matrix products, the multiplication between K and V can be carried out first. Due to $d_t \ll N$, the complexity is reduced from $O(N^2)$ to $O(N)$. The computation of the linear attention module is shown in Fig. 5. In practically, we repeatedly exchange self- and cross-attention layer to update fine-level features for $N_f = 3$ times to get reliable dense point matching.($i.e.$ for self-attention, $\mathbf{F}^a = \mathbf{F}^b = \mathbf{F}^{\bar{\mathcal{P}}}/\mathbf{F}^{\bar{\mathcal{Q}}}$; for cross-attention, $\mathbf{F}^a = \mathbf{F}^{\bar{\mathcal{P}}}, \mathbf{F}^b = \mathbf{F}^{\bar{\mathcal{Q}}}$ or $\mathbf{F}^a = \mathbf{F}^{\bar{\mathcal{Q}}}, \mathbf{F}^b = \mathbf{F}^{\bar{\mathcal{P}}}$).

Feature-similar-based Efficient Registration. The number of seed points N_s is set to $0.3 * |\mathcal{C}|$, where $|\mathcal{C}|$ is number of the correspondences set $\mathcal{C}$. For each seed, we use $ck = 20$ nearest neighbors search in $\mathbf{Z}^{\mathcal{C}}$ to expand into a consensus set. The total consensus sets can be noted as: $\mathcal{CS} \in \mathbb{R}^{N_s \times ck}$. When compute the feature similarity of correspondences, the hyper-parameter θ_m is set to 10. To select the best transformation, the acceptance radius is $\tau_a = 10$cm on 3DMatch and ModelNet, and $\tau_a = 60$cm on KITTI.

1.6 More qualitative Results

Indoor benchmark: 3DLoMatch. In Fig. 6, we show more qualitative comparative experimental results with Geotransformer[2], and our approach performs quite well in these low-overlap cases. It is worth pointing out that, due to our soft matching module allowing one-to-many matching in coarse-level matching phase, OAAFormer can mine more potential inliers located in overlapping regions, such as 1, 3, 5-th rows and 3-th cols. Morover, Our method can distinguish similar regions in close spatial positions, such as 6,7-th rows and 3-th cols, which is mainly because our linear attention module and relative position embedding can better identify the spatial positions of dense points, so that the features in similar regions have stronger discrimination ability. Additionally, we show more challenging qualitative results for 3DLoMatch with overlap rate below 15% in Fig. 7, our method can accurately identify overlapping regions and obtain a set of reliable dense correspondences.

Outdoor benchmark: KITTI. Fig. 8 visualizes the patch overlap prediction, dense correspondences and registration results of OAAFormer on KITTI. We can see that our method is still robust and effective even in sparse outdoor scenes. Here, accurate registration results obtained by using FSR as an estimator.

Synthetic Benchmark: ModelNet. As shown in the Fig. 9, thanks to the powerful local and global geometric modeling capabilities, OAAFormer also has excellent performance on the synthetic object dataset with fewer points.

1.7 Limitations

We further show some failure cases in Fig. 10. It can be observed that the main reasons for the failure of 3DLoMatch are as follows: In the first case, the overlap region has a similar ambiguous object, such as the yellow chair in the first row, while the blue area just has a similar chair. Unfortunately, it is not the overlap region, so the mismatches are established under the case of low overlap ratio. In the second case, the yellow table corner is located in the overlap region of the two point clouds, but the corresponding blue table corner is partial missing due to real scanning. Interestingly, the table corner just matches the "false corner" formed by the table and the missing area. In the third case, the overlap region is only on the floor, although 5% inliers are obtained, the matches between two plane cannot determine the consistent direction, resulting in the point cloud fitting in the opposite direction.

References

[1] Paszke A, Gross S, Massa F, Lerer A, Bradbury J, Chanan G, Chintala S. Pytorch: An imperative style, high-performance deep learning library. In *Proc. the 33th International Conference on Neural Information Processing Systems*, Dec. 2019, pp.8026-8037.

[2] Qin Z, Yu H, Wang C, Guo Y, Peng Y, Xu K. Geometric Transformer for Fast and Robust Point Cloud Registration. In *Proc. the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, Feb. 2022, pp.11133-11142. DOI: 10.1109/CVPR52688.2022.01086.

[3] Vaswani A, Shazeer N, Parmar N, Uszkoreit J, Jones L, Gomez A N, Polosukhin I. Attention is all you need. *Proc. the 31st International Conference on Neural Information Processing Systems*, Dec. 2017, pp.6000-6010.

[4] Fischler M A, Bolles, R C. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography.*ACM. Communications*, 1981, 24(6): 381-395. DOI: 10.1016/B978-0-08-051581-6.50070-2.

[5] Yew Z J, Lee G H. Regtr: End-to-end point cloud correspondences with transformers. *Proc. the IEEE/CVF conference on computer vision and pattern recognition*, Jun. 2022, pp.6677-6686. DOI: 10.1109/CVPR52688.2022.00656.

[6] Thomas H, Qi C R, Deschaud J E, Marcotegui B, Goulette F, Guibas L J. Kpconv: Flexible and deformable convolution for point clouds. *Proc. the IEEE/CVF international conference on computer vision*, Oct 2019, pp.6411-6420. DOI: 10.1109/ICCV.2019.00651.

[7] Huang S, Gojcic Z, Usvyatsov M, Wieser A, Schindler K. PREDATOR: Registration of 3D Point Clouds with Low Overlap. In *Proc. the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, Jun. 2020, pp.4265-4274. DOI: 10.1109/CVPR46437.2021.00425.

[8] Wu Y, He K. Group normalization. In *Proc. the European conference on computer vision*, Sep 2018, pp.3-19.

Table 1. Detailed network architecture for 3DMatch, KITTI and ModelNet.

Stage	3DMatch	KITTI	ModelNet
Backbone			

1	KPConv(1 → 64)
	ResBlock(64 → 128)
		KPConv(1 → 64)
		ResBlock(64 → 128)
			KPConv(1 → 64)
			ResBlock(64 → 128)

2	ResBlock(64 → 128, strided)
	ResBlock(128 → 256)
	ResBlock(256 → 256)
		ResBlock(64 → 128, strided)
		ResBlock(128 → 256)
		ResBlock(256 → 256)
			ResBlock(64 → 128, strided)
			ResBlock(128 → 256)
			ResBlock(256 → 256)

3	ResBlock(256 → 256, strided)
	ResBlock(256 → 512)
	ResBlock(512 → 512)
		ResBlock(256 → 256, strided)
		ResBlock(256 → 512)
		ResBlock(512 → 512)
			ResBlock(256 → 256, strided)
			ResBlock(256 → 512)
			ResBlock(512 → 512)

4	ResBlock(512 → 512, strided)
	ResBlock(512 → 1024)
	ResBlock(1024 → 1024)
		ResBlock(512 → 512, strided)
		ResBlock(512 → 1024)
		ResBlock(1024 → 1024)
			-

5	-	ResBlock(1024 → 1024, strided)
		ResBlock(1024 → 2048)
		ResBlock(2048 → 2048)
			-

6	-	NearestUpsampling UnaryConv(3072 → 1024)	-
7	NearestUpsampling UnaryConv(1536 → 512)	NearestUpsampling UnaryConv(1536 → 512)	NearestUpsampling UnaryConv(768 → 256)
8	NearestUpsampling UnaryConv(768 → 264)	NearestUpsampling UnaryConv(768 → 264)	NearestUpsampling UnaryConv(768 → 264)
Coarse-level Matching Module			
1	Linear(1024 → 256)	Linear(2048 → 128)	Linear(512 → 256)

2	GeometricSelfAttention(256, 4)
	VanillaCrossAttention(256, 4)
		GeometricSelfAttention(128, 4)
		VanillaCrossAttention(128, 4)
			GeometricSelfAttention(256, 4)
			VanillaCrossAttention(256, 4)

3	GeometricSelfAttention(256, 4)
	VanillaCrossAttention(256, 4)
		GeometricSelfAttention(128, 4)
		VanillaCrossAttention(128, 4)
			GeometricSelfAttention(256, 4)
			VanillaCrossAttention(256, 4)

4	GeometricSelfAttention(256, 4)
	VanillaCrossAttention(256, 4)
		GeometricSelfAttention(128, 4)
		VanillaCrossAttention(128, 4)
			GeometricSelfAttention(256, 4)
			VanillaCrossAttention(256, 4)

5	Linear(256 → 256)	Linear(128 → 128)	Linear(256 → 256)
Fine-level Matching Module			
1	Linear(264 → 264)	Linear(264 → 264)	Linear(264 → 264)

2	LinearSelfAttention(264, 4)
	LinearCrossAttention(264, 4)
		LinearSelfAttention(264, 4)
		LinearCrossAttention(264, 4)
			LinearSelfAttention(264, 4)
			LinearCrossAttention(264, 4)

3	LinearSelfAttention(264, 4)
	LinearCrossAttention(264, 4)
		LinearSelfAttention(264, 4)
		LinearCrossAttention(264, 4)
			LinearSelfAttention(264, 4)
			LinearCrossAttention(264, 4)

4	LinearSelfAttention(264, 4)
	LinearCrossAttention(264, 4)
		LinearSelfAttention(264, 4)
		LinearCrossAttention(264, 4)
			LinearSelfAttention(264, 4)
			LinearCrossAttention(264, 4)

5	Linear(264 → 264)	Linear(264 → 264)	Linear(264 → 264)

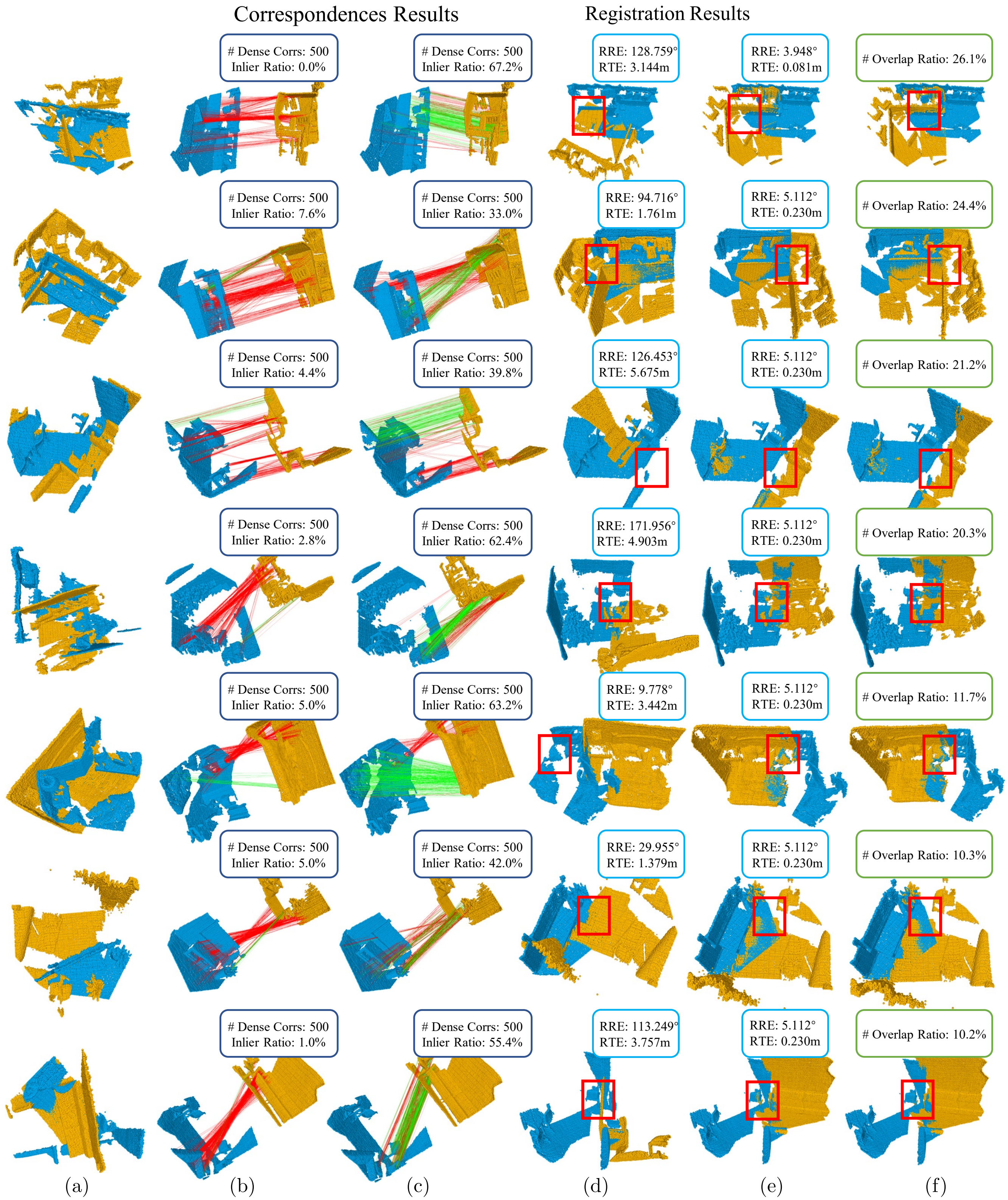


Fig.6. More qualitative results on 3DLoMatch. Geotransformer[?] serves as the baseline. (a) illustrates the input point cloud. (b) and (c) respectively depict correspondence results of Geotransformer[?] and our method. (d) and (e) respectively illustrate registration results of Geotransformer[?] and our method. (f) represents the ground-truth. Green/red lines indicate inliers/outliers.

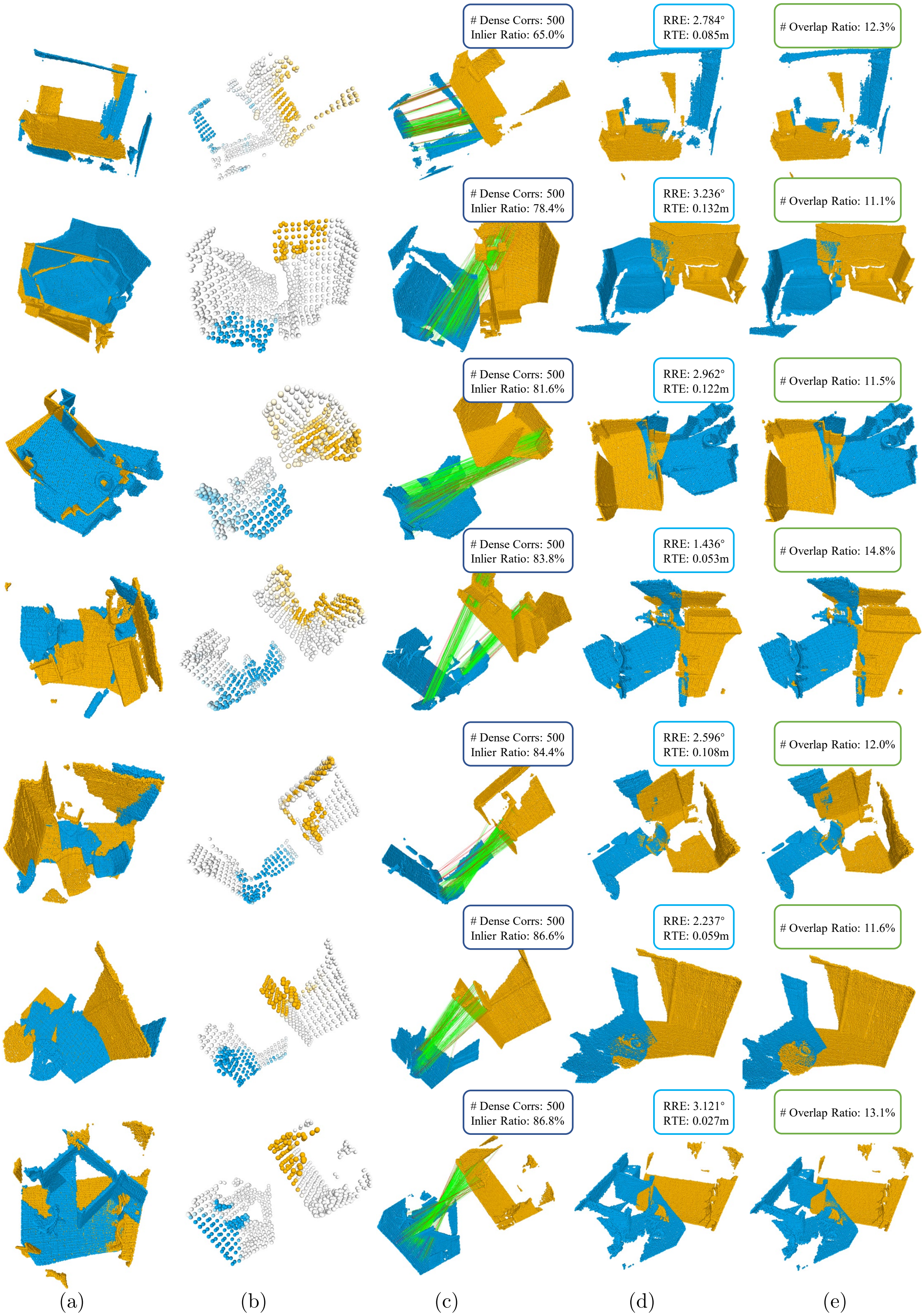
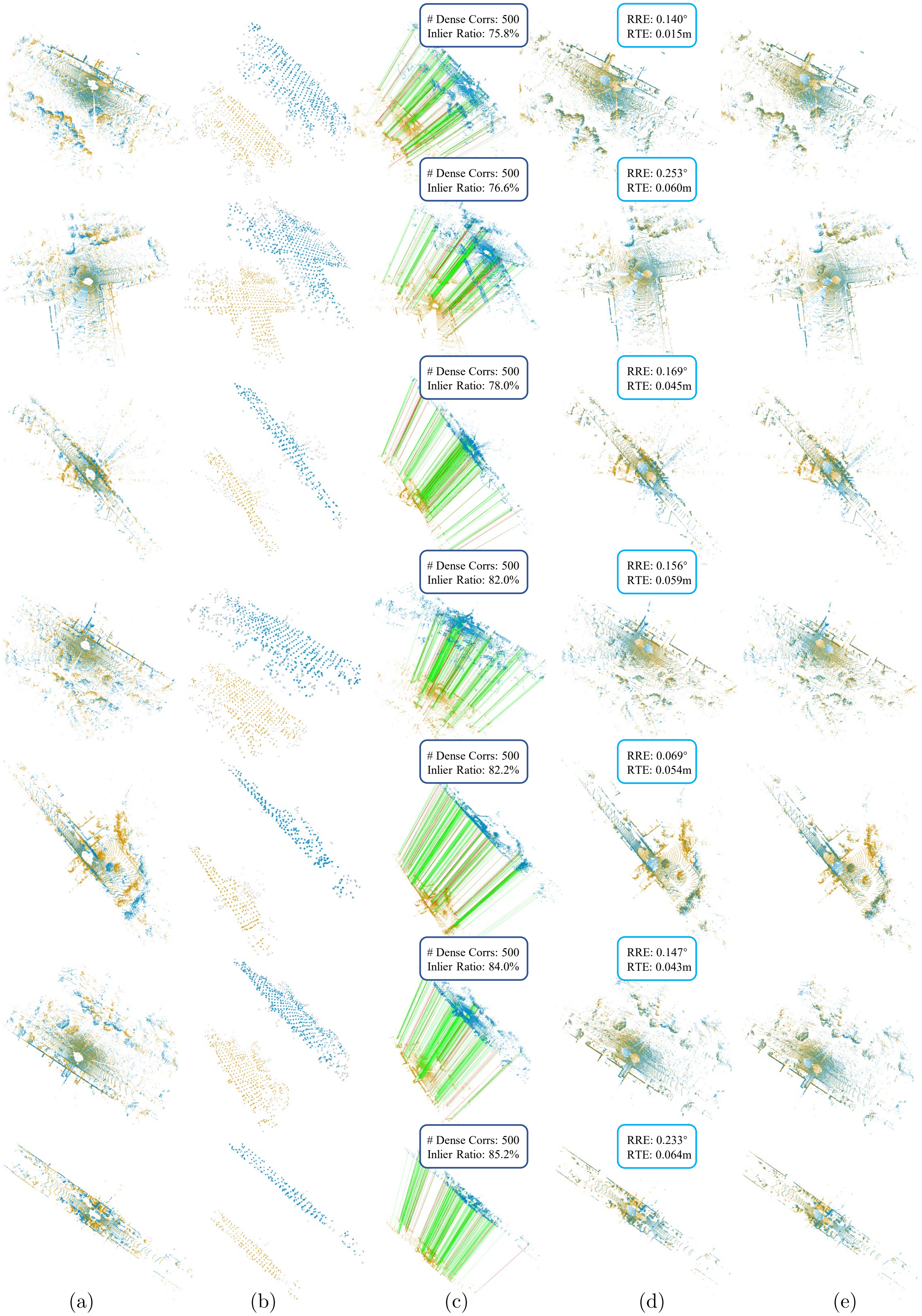


Fig. 7. More qualitative results on 3DLoMatch. (a) illustrates the input point cloud. (b) shows the predicted overlap region. (c) represents the established correspondences. (d) demonstrates the registration results. Green/red lines indicate inliers/outliers.



(a) (b) (c) (d) (e)

Fig.8. More qualitative results on KITTI. (b) shows the predicted overlap region, (c) represents the established correspondences, and (d) demonstrates the registration results. (f) represents the ground-truth. Green/red lines indicate inliers/outliers.

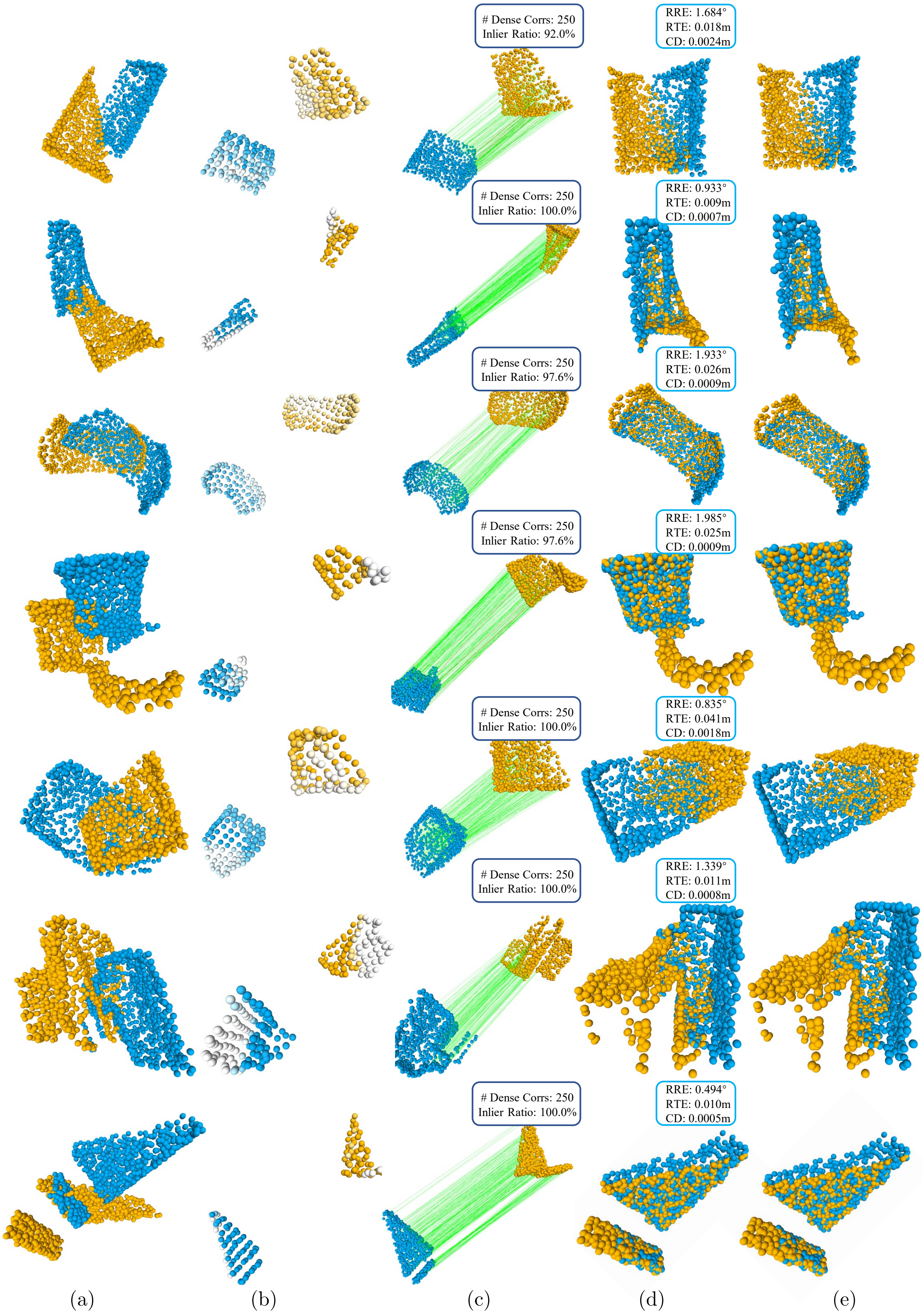


Fig. 9. More qualitative results on ModelLoNet. (a) illustrates the input point cloud. (b) shows the predicted overlap region. (c) represents the established correspondences. (d) demonstrates the registration results. Green/red lines indicate inliers/outliers.

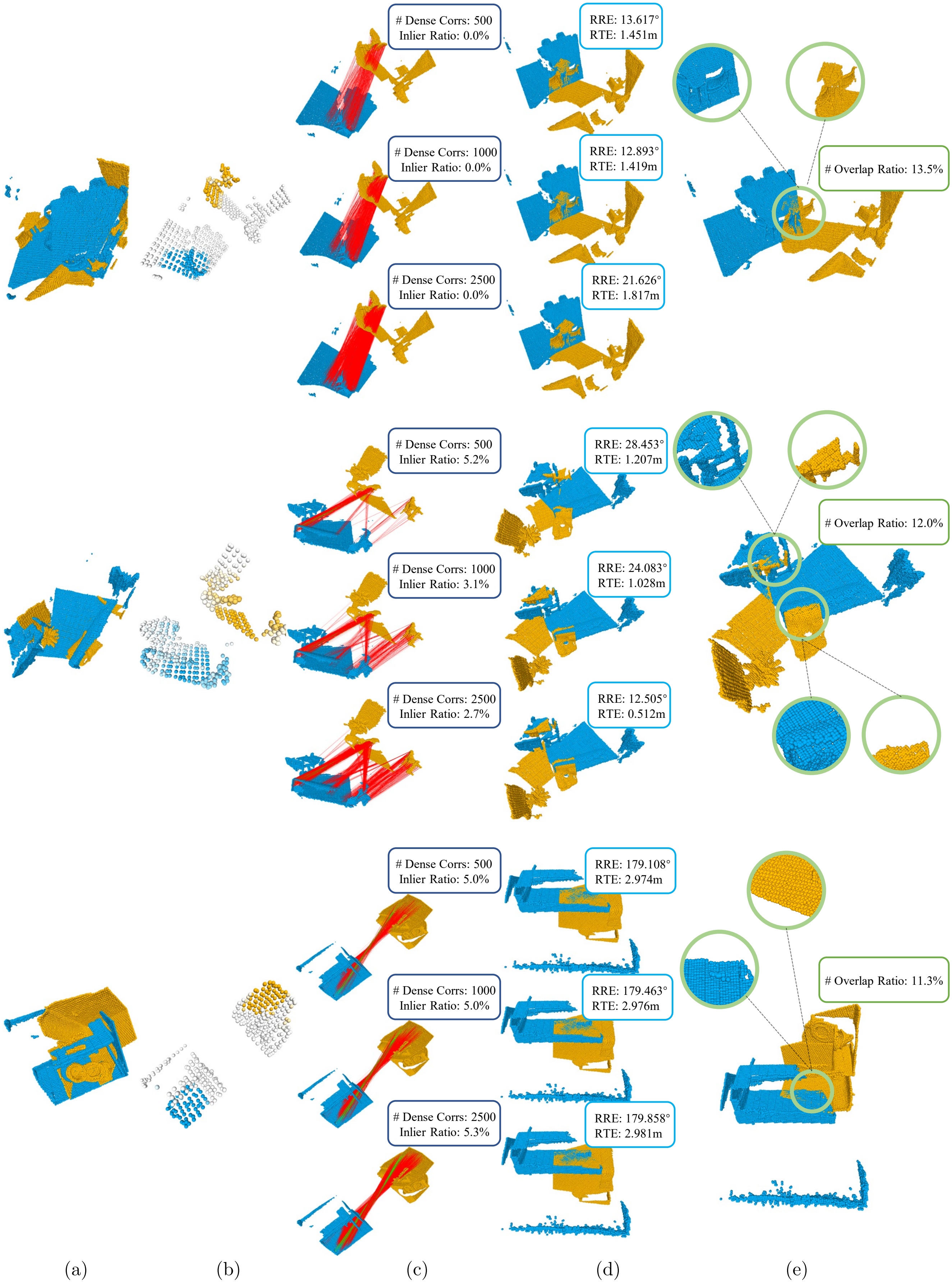


Fig.10. Failed cases on 3DLoMatch. (a) illustrates the input point cloud. (b) shows the predicted overlap region. (c) represents the established correspondences. (d) demonstrates the registration results. Green/red lines indicate inliers/outliers.