Rotation-Invariant Transformer for Point Cloud Matching
Supplementary Material

In this appendix, we first provide the implementation details in Sec. A. Then the network architecture is detailed in Sec. B. Moreover, the definition and ablation study of the geometry representation are illustrated in Sec. C. The details of the loss function and the evaluation metrics are given in Sec. D and Sec. E, respectively. Furthermore, more quantitative and qualitative results are demonstrated in Sec. F and Sec. G, respectively. Finally, the runtime is analyzed in Sec. H, and the limitations are discussed in Sec. I.

A. Implementation Details

We implement RoITr with PyTorch [9]. The matching model can be trained end-to-end on a single Nvidia RTX 3090 with 24G memory. In practice, we train the model on-parallel using 4× Nvidia 3090 GPUs for ∼35 epochs on both 3DMatch/3DLoMatch [6, 19] and 4DMatch/4DLoMatch [8]. It takes ∼35 hours and ∼30 hours for full convergence on 3DMatch/3DLoMatch and 4DMatch/4DLoMatch, respectively. The batch size is set to 1. We use an Adam optimizer [7] with an initial learning rate of 1e-4, which is exponentially decayed by 0.05 after each epoch. On 3DMatch/3DLoMatch, we select $|\mathcal{C}'| = 256$ superpoint correspondences with the highest scores. Based on each superpoint correspondence, we further extract the mutual top-3 point correspondences whose confidence scores are larger than 0.05 as the point correspondences. For non-rigid matching, we first pick the superpoint correspondences whose Euclidean distance is smaller than 0.75 (pick the top-128 instead if the number of selected correspondences is smaller than 128) and extract the mutual top-2 point correspondences with scores larger than 0.05.

B. Network Architecture

PPFTrans Encoder-Decoder. We detail the architecture of PPFTrans in Tab. 1. The encoder part has $4\times$ encoder blocks. In each block, AAL first downsamples the points and aggregates the information in a local vicinity. PAL further enhances the features with both the pose-agnostic local geometry and highly-representative learned context. The decoder part also comprises $4\times$ decoder blocks. In each block (except for Block_4), TUL first subsamples the points and incorporates the information flowing from the encoder via skip connections. The obtained features are further enhanced by the following PAL.

Global Transformer. The details of the global transformer are demonstrated in Tab. 2. It has $3\times$ transformer blocks, each comprising a geometry-aware self-attention module (GSM) followed by a position-aware cross-attention module (PCM). In each transformer block, GSM first aggregates the global context individually for each point cloud. Then in PCM, the global context flows from the second frame to the first one and then from the first frame to the second one.

Feed-Forward Network. The structure of the feed-forward network is illustrated in Fig. 1. It details the feed-forward network in the context branch of GSM in Fig. 4 of the main paper.

C. Geometry Representation

Taking superpoints $\mathbf{P}' \in \mathbb{R}^{n' \times 3}$ as an instance, the geometry representation $\mathbf{R}' \in \mathbb{R}^{n' \times n' \times c'}$ proposed in [10] depicts the pairwise geometric relationship among superpoints in a rotation-invariant fashion. It comprises a distance-based part $\mathbf{R}'_D \in \mathbb{R}^{n' \times n' \times c'}$ as well as an angle-based part $\mathbf{R}'_A \in \mathbb{R}^{n' \times n' \times 3 \times c'}$.

Euclidean Distance. The pairwise Euclidean distance is defined as $\rho_{i,j} = \|\mathbf{p}'_i - \mathbf{p}'_j\|_2$, which is projected to a c'-dimension (note that c' must be an even number) embedding via the sinusoidal function [14]:

$$\begin{cases} \mathbf{R}'_D(i,j,2l+1) = \sin(\frac{\rho_{i,j}/\sigma_d}{10000^{2l/c'}}), \\ \mathbf{R}'_D(i,j,2l+2) = \cos(\frac{\rho_{i,j}/\sigma_d}{10000^{2l/c'}}), \end{cases} \tag{1}$$

with $0 \leq l < c'/2$ and $\sigma_d = 0.2$.

Angles. Given a superpoint pair $(\mathbf{p}'_i, \mathbf{p}'_j)$, the 3-nearest neighbors of $\mathbf{p}'_i$ w.r.t. $\mathbf{P}'$ is first retrieved and denoted as $\mathcal{N}(i)$. For each $k \in \mathcal{N}(i)$, we calculate the angle between two vectors by $\alpha^k_{i,j} = \angle(\mathbf{p}'_\mathbf{k} - \mathbf{p}'_\mathbf{i}, \mathbf{p}'_\mathbf{j} - \mathbf{p}'_\mathbf{i})$ [3, 4], upon

Stage	Block	Operation
Input		$\mathcal{P} = (\mathbf{P}, \mathbf{N}, \mathbf{X} \in \mathbb{R}^{n \times 1})$
Encoder	$\text{Block}_1^e(\mathcal{P}) \rightarrow \mathcal{P}_1$	$\text{AAL}(n \times 1) \rightarrow n \times 64$ $\text{PAL}(n \times 64) \rightarrow n \times 64$
	$\text{Block}_2^e(\mathcal{P}_1) \rightarrow \mathcal{P}_2$	$\text{AAL}(n \times 64) \rightarrow n/4 \times 128$ $\text{PAL}(n/4 \times 128) \rightarrow n/4 \times 128$
	$\text{Block}_3^e(\mathcal{P}_2) \rightarrow \mathcal{P}_3$	$\text{AAL}(n/4 \times 128) \rightarrow n/16 \times 256$ $\text{PAL}(n/16 \times 256) \rightarrow n/16 \times 256$
	$\text{Block}_4^e(\mathcal{P}_3) \rightarrow \mathcal{P}'$	$\text{AAL}(n/16 \times 256) \rightarrow n/64 \times 256$ $\text{PAL}(n/64 \times 256) \rightarrow n/64 \times 256$
Decoder	$\text{Block}_4^d(\mathcal{P}') \rightarrow \hat{\mathcal{P}}_4$	$\text{TUL}\ (n/64 \times 256) \rightarrow n/64 \times 256$ $\text{PAL:}\ n/64 \times 256 \rightarrow n/64 \times 256$
	$\text{Block}_3^d(\hat{\mathcal{P}}_4, \mathcal{P}_3) \rightarrow \hat{\mathcal{P}}_3$	$\text{TUL}(n/64 \times 256,\ n/16 \times 256) \rightarrow n/16 \times 256$ $\text{PAL}(n/16 \times 256) \rightarrow n/16 \times 256$
	$\text{Block}_2^d(\hat{\mathcal{P}}_3, \mathcal{P}_2) \rightarrow \hat{\mathcal{P}}_2$	$\text{TUL}(n/16 \times 256,\ n/4 \times 128) \rightarrow n/4 \times 128$ $\text{PAL}(n/4 \times 128) \rightarrow n/4 \times 128$
	$\text{Block}_1^d(\hat{\mathcal{P}}_2, \mathcal{P}_1) \rightarrow \hat{\mathcal{P}}$	$\text{TUL}(n/4 \times 128,\ n \times 64) \rightarrow n \times 64$ $\text{PAL}(n \times 64) \rightarrow n \times 64$
Output		$\mathcal{P}' = (\mathbf{P}', \mathbf{N}', \mathbf{X}');\quad \hat{\mathcal{P}} = (\hat{\mathbf{P}}, \hat{\mathbf{N}}, \hat{\mathbf{X}})$

Table 1. Detailed architecture of the PPFTrans encoder-decoder.

Block	Module	Operation	
Input		$\mathcal{P}' = (\mathbf{P}', \mathbf{N}', \mathbf{X}')$	$\mathcal{Q}' = (\mathbf{Q}', \mathbf{M}', \mathbf{Y}')$
Trans_1	$\text{Self}_1(\mathcal{P}') \rightarrow \widetilde{\mathcal{P}}_1'$ | $\text{Self}_1(\mathcal{Q}') \rightarrow \widetilde{\mathcal{Q}}_1'$ $\text{Cross}_1(\widetilde{\mathcal{P}}_1', \widetilde{\mathcal{Q}}_1') \rightarrow \mathcal{P}_1'$ $\text{Cross}_1(\widetilde{\mathcal{Q}}_1', \mathcal{P}_1') \rightarrow \mathcal{Q}_1'$	$\text{GSM}(n' \times c') \rightarrow n' \times c'$ | $\text{GSM}(m' \times c') \rightarrow m' \times c'$ $\text{PCM}(n' \times c', m' \times c') \rightarrow n' \times c'$ $\text{PCM}(m' \times c', n' \times c') \rightarrow m' \times c'$	
Trans_2	$\text{Self}_2(\mathcal{P}_1') \rightarrow \widetilde{\mathcal{P}}_2'$ | $\text{Self}_2(\mathcal{Q}_1') \rightarrow \widetilde{\mathcal{Q}}_2'$ $\text{Cross}_2(\widetilde{\mathcal{P}}_2', \widetilde{\mathcal{Q}}_2') \rightarrow \mathcal{P}_2'$ $\text{Cross}_2(\widetilde{\mathcal{Q}}_2', \mathcal{P}_2') \rightarrow \mathcal{Q}_2'$	$\text{GSM}(n' \times c') \rightarrow n' \times c'$ | $\text{GSM}(m' \times c') \rightarrow m' \times c'$ $\text{PCM}(n' \times c', m' \times c') \rightarrow n' \times c'$ $\text{PCM}(m' \times c', n' \times c') \rightarrow m' \times c'$	
Trans_3	$\text{Self}_3(\mathcal{P}_2') \rightarrow \widetilde{\mathcal{P}}_3'$ | $\text{Self}_3(\mathcal{Q}_2') \rightarrow \widetilde{\mathcal{Q}}_3'$ $\text{Cross}_3(\widetilde{\mathcal{P}}_3', \widetilde{\mathcal{Q}}_3') \rightarrow \widetilde{\mathcal{P}}'$ $\text{Cross}_3(\widetilde{\mathcal{Q}}_3', \mathcal{P}_3') \rightarrow \widetilde{\mathcal{Q}}'$	$\text{GSM}(n' \times c') \rightarrow n' \times c'$ | $\text{GSM}(m' \times c') \rightarrow m' \times c'$ $\text{PCM}(n' \times c', m' \times c') \rightarrow n' \times c'$ $\text{PCM}(m' \times c', n' \times c') \rightarrow m' \times c'$	
Output		$\widetilde{\mathcal{P}}' = (\mathbf{P}', \mathbf{N}', \widetilde{\mathbf{X}}')$	$\widetilde{\mathcal{Q}}' = (\mathbf{Q}', \mathbf{M}', \widetilde{\mathbf{Y}}')$

Table 2. Detailed architecture of the global transformer.

which the c'-dimension angle-based embedding is defined as:

$$\begin{cases} \mathbf{R}'_A(i,j,k,2l+1) = \sin(\frac{\alpha_{i,j}^k / \sigma_a}{10000^{2l/c'}}), \\ \mathbf{R}'_A(i,j,k,2l+2) = \cos(\frac{\alpha_{i,j}^k / \sigma_a}{10000^{2l/c'}}), \end{cases} \tag{2}$$

with $0 \leq l < c'/2$ and $\sigma_a = 15$.

The pairwise geometry representation $\mathbf{R}'$ finally reads as:

$$\mathbf{R}' = \mathbf{R}'_D \mathbf{W}_D + \max_k (\mathbf{R}'_A \mathbf{W}_A), \tag{3}$$

where $\max_k(\mathbf{R}'_A \mathbf{W}_A)$ indicates the max-pooling operation over the second last dimension, and $\mathbf{W}_D, \mathbf{W}_A \in \mathbb{R}^{c' \times c'}$ stand for two learnable matrices.

Our Model with	3DMatch FMR	3DMatch IR	3DMatch RR	3DLoMatch FMR	3DLoMatch IR	3DLoMatch RR	Size	Time
Point Pair Features	97.9	81.8	91.6	88.6	52.4	73.0	**9.5M**	**0.213s**
GeoTrans	**98.0**	**82.6**	**91.9**	**89.6**	**54.3**	**74.7**	10.1M	0.233s

Table 3. Ablation study of different global geometric embedding.

Ablation Study. Using the geometry representation proposed in [10] (instead of the point pair features [5]) in the global transformer moderately improves the results (see Tab. 3) despite a slightly larger memory footprint. Using it, RoITr has a comparable model size with GeoTrans [10] (10.1M v.s. 9.8M) and is significantly more

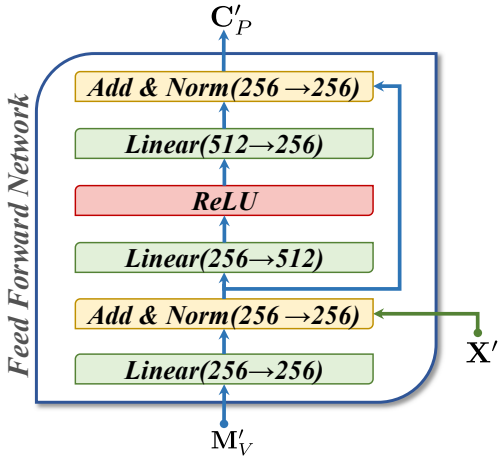

Figure 1. Detailed architecture of the feed-forward network. LayerNorm [2] is used for normalization.

lightweight than Lepard [8] (10.1M v.s. 37.6M).

D. Loss Function

Superpoint Matching Loss. We use the Circle Loss [12] for superpoint matching. Following [10], we use the overlap ratio between the vicinity of superpoints to weigh different ground truth superpoint correspondences. More specifically, for each superpoint $\mathbf{p}'_i \in \mathbf{P}'$ with an associated feature $\tilde{\mathbf{x}}'_i$ (unit vector after normalization), we sample a positive set of superpoints from $\mathbf{Q}'$, denoted as $\mathcal{E}_i^P = \{\mathbf{q}'_j \in \mathbf{Q}' | \mathcal{O}(\mathbf{p}'_i, \mathbf{q}'_j) > \tau_r\}$, where $\mathcal{O}$ is the function that calculates the overlap ratio between the vicinity of two superpoints, and τ_r is the threshold to select positive samples (τ_r=0.1 by default). The overlap ratio function is defined as:

$$\mathcal{O}(\mathbf{p}'_i, \mathbf{q}'_j) = \frac{|\{\hat{\mathbf{p}}_u \in \hat{\mathbf{G}}_i^P | \exists \hat{\mathbf{q}}_v \in \hat{\mathbf{G}}_j^Q \text{ s.t. } \hat{\mathbf{p}}_u \Leftrightarrow \hat{\mathbf{q}}_v\}|}{|\{\hat{\mathbf{p}}_u \in \hat{\mathbf{G}}_i^P\}|}, \tag{4}$$

where $\Leftrightarrow$ denotes the correspondence relationship and $\hat{\mathbf{G}}_i^P$ is the group of points from $\hat{\mathbf{P}}$ assigned to $\mathbf{p}'_i$ by the Point-to-Node grouping strategy [17].

We further sample a negative set of superpoints $\mathcal{F}_i^P = \{\mathbf{q}'_j \in \mathbf{Q}' | \mathcal{O}(\mathbf{p}'_i, \mathbf{q}'_j) = 0\}$. Then for $\mathbf{P}'$, the superpoint matching loss is computed as:

$$\mathcal{L}_s^P = \frac{1}{n'} \sum_{i=1}^{n'} \log[1 + \sum_{\mathbf{q}'_j \in \mathcal{E}_i^P} e^{r_i^j \beta_e^{i,j} (d_i^j - \Delta_e)} \cdot \sum_{\mathbf{q}'_k \in \mathcal{F}_i^P} e^{\beta_f^{i,k} (\Delta_f - d_i^k)}], \tag{5}$$

with $r_i^j := \mathcal{O}(\mathbf{p}'_i, \mathbf{q}'_j)$ and $d_i^j = \|\tilde{\mathbf{x}}'_i - \tilde{\mathbf{y}}'_j\|_2$. Moreover, Δ_e and Δ_f are the positive and negative margins (Δ_e=0.1 and Δ_f=1.4 by default). $\beta_e^{i,j} = \gamma(d_i^j - \Delta_e)$ and $\beta_f^{i,k} = \gamma(\Delta_f - d_i^k)$ are the weights individually determined for different samples, with γ a hyper-parameter. The same loss for $\mathbf{Q}'$ is defined in a similar way, and the overall loss reads as $\mathcal{L}_s = (\mathcal{L}_s^P + \mathcal{L}_s^Q)/2$.

Point Matching Loss. For each superpoint correspondence $\mathcal{C}'_l = (\mathbf{p}'_i, \mathbf{q}'_j) \in \mathcal{C}'$, we adopt a negative log-likelihood loss [11] on its corresponding normalized $\hat{\mathbf{S}}_l \in \mathbb{R}^{(|\hat{\mathbf{G}}_i^P|+1)\times(|\hat{\mathbf{G}}_j^Q|+1)}$. We define $\mathcal{M}_l = \{(u,v) | \hat{\mathbf{p}}_u \Leftrightarrow \hat{\mathbf{q}}_v$ with $\hat{\mathbf{p}}_u \in \hat{\mathbf{G}}_i^P, \hat{\mathbf{q}}_v \in \hat{\mathbf{G}}_j^Q\}$ as the set comprising the indices of corresponding points. We further define $\mathcal{I}_l = \{u | \hat{\mathbf{p}}_u \nLeftrightarrow \hat{\mathbf{q}}_v, \forall \hat{\mathbf{q}}_v \in \hat{\mathbf{G}}_j^Q\}$ and $\mathcal{J}_l = \{v | \hat{\mathbf{q}}_v \nLeftrightarrow \hat{\mathbf{p}}_u, \forall \hat{\mathbf{p}}_u \in \hat{\mathbf{G}}_i^P\}$ as the sets of indices of points that have no correspondence in the opposite frame, with $\nLeftrightarrow$ depicting the non-correspondence relationship. Then the point matching loss on $\mathcal{C}'_l$ can be defined as:

$$\mathcal{L}_p^l = -\sum_{(u,v)\in\mathcal{M}_l} \log \hat{s}_{u,v}^l - \sum_{u\in\mathcal{I}_l} \log \hat{s}_{u,|\hat{\mathbf{G}}_j^Q|+1}^l - \sum_{v\in\mathcal{J}_l} \log \hat{s}_{|\hat{\mathbf{G}}_i^P|+1,v}^l, \tag{6}$$

where $\hat{s}_{u,v}^l := \hat{\mathbf{S}}_l(u,v)$ stands for the entry on the u^{th} row and v^{th} column of $\hat{\mathbf{S}}_l$. The overall point matching loss reads as $\mathcal{L}_p = \frac{1}{|\mathcal{C}'|} \sum_{l=1}^{|\mathcal{C}'|} \mathcal{L}_p^l$.

E. Detailed Metrics

Given a point cloud pair $\mathbf{P} \in \mathbb{R}^{n\times 3}$ and $\mathbf{Q} \in \mathbb{R}^{m\times 3}$, RoITr generates a correspondence set $\hat{\mathcal{C}}$ by matching the downsampled point cloud pair $\hat{\mathbf{P}} \in \mathbb{R}^{\hat{n}\times 3}$ and $\hat{\mathbf{Q}} \in \mathbb{R}^{\hat{m}\times 3}$. We detail all the metrics for evaluation hereafter.

Inlier Ratio (IR). IR counts the fraction of putative correspondences $(\hat{\mathbf{p}}_i, \hat{\mathbf{q}}_j) \in \hat{\mathcal{C}}$ whose Euclidean distance is under a threshold τ_1 (0.1m on 3DMatch/3DLoMatch, 0.04m on 4DMatch/4DLoMatch) under the ground-truth transformation $\mathbf{T}^*$:

$$\mathcal{I}(\hat{\mathcal{C}} | \mathbf{T}^*) = \frac{1}{|\hat{\mathcal{C}}|} \sum_{(\hat{\mathbf{p}}_i, \hat{\mathbf{q}}_j)\in\hat{\mathcal{C}}} \mathbb{1}(\|\mathbf{T}^*(\hat{\mathbf{p}}_i) - \hat{\mathbf{q}}_j\|_2 < \tau_1), \tag{7}$$

with $\mathbb{1}(\cdot)$ the indicator function.

Feature Matching Recall (FMR). FMR counts the fraction of point cloud pairs whose IR is larger than a threshold $\tau_2 = 0.05$:

$$\mathcal{F}(\mathcal{T}) = \frac{1}{|\mathcal{T}|} \sum_{t=1}^{|\mathcal{T}|} \mathbb{1}(\mathcal{I}(\hat{\mathcal{C}}_t | \mathbf{T}_t^*) > \tau_2), \tag{8}$$

with $\mathcal{T}$ the testing set and $\mathcal{T}_t$ the t^{th} point cloud pair in the dataset.

Registration Recall (RR). RR computes the fraction of point cloud pairs that are registered correctly based on the putative correspondences, measured by the *Root-Mean-Square Error* (RMSE). Following [6], we define RMSE on the original 3DMatch/3DLoMatch as:

$$\mathcal{R}_1(\hat{\mathcal{C}}|\mathcal{C}^*) = \sqrt{\frac{1}{|\mathcal{C}^*|}\sum_{(\mathbf{p}_i,\mathbf{q}_j)\in\mathcal{C}^*} \|\hat{\mathbf{T}}(\mathbf{p}_i) - \mathbf{q}_j\|_2^2}, \quad (9)$$

with $\mathcal{C}^*$ the ground-truth correspondence set established upon $\mathbf{P}$ and $\mathbf{Q}$, and $\hat{\mathbf{T}}$ the transformation estimated based on $\hat{\mathcal{C}}$. On Rotated 3DMatch/3DLoMatch, we follow [16, 18] and define the RMSE as:

$$\mathcal{R}_2(\hat{\mathcal{C}}|\mathbf{T}^*, \mathbf{P}) \approx \frac{1}{n}\sqrt{\sum_{\mathbf{p}_i\in\mathbf{P}} \|\hat{\mathbf{T}}(\mathbf{p}_i) - \mathbf{T}^*(\mathbf{p}_i)\|_2^2}, \quad (10)$$

with $\hat{\mathbf{T}}$ the transformation estimated based on $\hat{\mathcal{C}}$ and $\mathbf{T}^*$ the ground-truth transformation. RR is finally calculated as:

$$\begin{aligned}\mathcal{R}(\mathcal{T}) = &\frac{1}{|\mathcal{T}|}\sum_{t=1}^{\mathcal{T}} \mathbb{1}(\mathcal{R}_1(\hat{\mathcal{C}}|\mathcal{C}^*) < \tau_3) \quad \text{or} \\ &\frac{1}{|\mathcal{T}|}\sum_{t=1}^{\mathcal{T}} \mathbb{1}(\mathcal{R}_2(\hat{\mathcal{C}}|\mathbf{T}^*, \mathbf{P})) < \tau_3),\end{aligned} \quad (11)$$

with τ_3 = 0.2m.

Non-Rigid Feature Matching Recall (NFMR). NFMR counts the fraction of ground-truth correspondences $\mathcal{C}^*$ that can be recovered by the putative correspondences $\hat{\mathcal{C}}$. The deformation flow $\hat{\mathbf{d}}_u$ for each putative correspondence $(\hat{\mathbf{p}}_u, \hat{\mathbf{q}}_v) \in \hat{\mathcal{C}}$ is defined as $\hat{\mathbf{d}}_u = \hat{\mathbf{q}}_v - \hat{\mathbf{p}}_u$. Then for each $(\mathbf{p}_i, \mathbf{q}_j) \in \mathcal{C}^*$, the deformation flow can be computed via interpolation:

$$\mathbf{d}_i = \frac{\sum_{u\in\mathcal{N}(i)} w_u^i \hat{\mathbf{d}}_u}{\sum_{u\in\mathcal{N}(i)} w_u^i}, \text{ with } w_u^i = \frac{1}{\|\mathbf{p}_i - \hat{\mathbf{p}}_u\|_2}, \quad (12)$$

where $\mathcal{N}(i)$ indicates the k-nearest neighbor ($k = 3$ in practice) of $\mathbf{p}_i$ w.r.t. points $\hat{\mathbf{p}}_u$ s.t. $(\hat{\mathbf{p}}_u, \hat{\mathbf{q}}_v) \in \hat{\mathcal{C}}$. NFMR is finally computed by:

$$\mathcal{F}_N(\mathcal{C}^*|\hat{\mathcal{C}}) = \frac{1}{|\mathcal{C}^*|}\sum_{(\mathbf{p}_i,\mathbf{q}_j)\in\mathcal{C}^*} \mathbb{1}(\|\mathbf{d}_i - \mathbf{d}_i^*\|_2 < \tau_4), \quad (13)$$

with $\mathbf{d}_i^*$ the ground-truth deformation flow and τ_4=0.04m in practice.

F. More Quantitative Results

Varying Correspondence Number on Rotated Data. We further analyze the performance of different methods w.r.t. the varying number of correspondences on rotated data. The superiority of RoITr can be observed in Tab. 4.

	Rotated 3DMatch					Rotated 3DLoMatch				
# Samples	5000	2500	1000	500	250	5000	2500	1000	500	250
Feature Matching Recall (%) ↑										
SpinNet [1]	97.4	97.4	96.7	96.5	94.1	75.2	74.9	72.6	69.2	61.8
Predator [6]	96.2	96.2	96.6	96.0	96.0	73.7	74.2	75.0	74.8	73.5
CoFiNet [17]	97.4	97.4	97.2	97.2	97.3	78.6	78.8	79.2	78.9	79.2
YOHO [15]	97.8	97.8	97.4	97.6	96.4	77.8	77.8	76.3	73.9	67.3
RIGA [16]	**98.2**	**98.2**	**98.2**	98.0	**98.1**	84.5	84.6	84.5	84.2	84.4
GeoTrans [10]	97.8	97.9	98.1	97.7	97.3	85.8	85.7	86.5	86.6	86.1
RoITr (*Ours*)	**98.2**	98.1	98.1	**98.1**	**98.1**	**89.4**	**89.2**	**89.1**	**89.1**	**89.0**
Inlier Ratio (%) ↑										
SpinNet [1]	48.7	46.0	40.6	35.1	29.0	25.7	23.9	20.8	17.9	15.6
Predator [6]	52.8	53.4	52.5	50.0	45.6	22.4	23.5	23.0	23.2	21.6
CoFiNet [17]	46.8	48.2	49.0	49.3	49.3	21.5	22.8	23.6	23.8	23.8
YOHO [15]	64.1	60.4	53.5	46.3	36.9	23.2	23.2	19.2	15.7	12.1
RIGA [16]	68.5	69.8	70.7	71.0	71.2	32.1	33.5	34.3	34.7	35.0
GeoTrans [10]	68.2	72.5	73.3	79.5	82.3	40.0	40.3	42.7	49.5	54.1
RoITr (*Ours*)	**82.3**	**82.3**	**82.6**	**82.6**	**82.6**	**53.2**	**54.9**	**55.1**	**55.2**	**55.3**
Registration Recall (%) ↑										
SpinNet [1]	93.2	93.2	91.1	87.4	77.0	61.8	59.1	53.1	44.1	30.7
Predator [6]	92.0	92.8	92.0	92.2	89.5	58.6	59.5	60.4	58.6	55.8
CoFiNet [17]	92.0	91.4	91.0	90.3	89.6	62.5	60.9	60.9	59.9	56.5
YOHO [15]	92.5	92.3	92.4	90.2	87.4	66.8	67.1	64.5	58.2	44.8
RIGA [16]	93.0	93.0	92.6	91.8	92.3	66.9	67.6	67.0	66.5	66.2
GeoTrans [10]	92.0	91.9	91.8	91.5	91.4	71.8	72.0	72.0	71.6	70.9
RoITr (*Ours*)	**94.7**	**94.9**	**94.4**	**94.4**	**94.2**	**77.2**	**76.5**	**76.6**	**76.5**	**76.0**

Table 4. Quantitative results on Rotated 3DMatch & 3DLoMatch with a varying number of points/correspondences.

Ablation Study on (Rotated) 3DMatch. We also conduct ablation study on (Rotated) 3DMatch, as shown in Tab. 5. Similar to the ablation study on (Rotated) 3DLoMatch shown in the main paper, our default model achieves the best performance, which further confirms the significance of each individual design of RoITr.

		3DMatch					
		Origin			Rotated		
Category	Model	FMR	IR	RR	FMR	IR	RR
a. Local	1. PT [20]	96.7	71.0	87.6	96.4	69.5	90.5
	*2. PPF+PT [20]	97.9	80.1	91.2	97.8	79.8	93.9
	3. Δxyz+Ours	-	-	-	-	-	-
	*4. *Ours*	**98.0**	**82.6**	**91.9**	**98.2**	**82.3**	**94.7**
b. Aggregation	*1. max pooling	97.9	80.8	90.7	97.8	80.8	94.1
	*2. avg pooling	**98.1**	81.8	**92.1**	**98.2**	81.8	**94.8**
	*3. *Ours*	98.0	**82.6**	91.9	**98.2**	**82.3**	94.7
c. Backbone	1. KPConv [13]	97.9	74.6	91.1	97.3	72.8	94.3
	*2. *Ours*	**98.0**	**82.6**	**91.9**	**98.2**	**82.3**	**94.7**
d. Global	*1. GeoTrans [10]	97.9	**82.6**	90.8	98.0	**82.3**	94.5
	*2. *Ours*	**98.0**	**82.6**	**91.9**	**98.2**	**82.3**	**94.7**
e. #Global	*1. $g = 0$	**98.5**	65.8	90.9	98.3	65.9	93.7
	*2. $g = 1$	98.4	74.8	90.8	**98.5**	74.8	94.2
	*3. $g = 3$ (*Ours*)	98.0	**82.6**	**91.9**	98.2	**82.3**	**94.7**
	*4. $g = 5$	98.1	82.0	91.7	98.0	82.0	94.6

Table 5. Ablation study on (Rotated) 3DMatch. 5,000 points/correspondences are leveraged. * indicates the methods with intrinsic rotation invariance.

Method	Data (s)↓	Model (s)↓	Total (s)↓
Lepard [8]	0.444	**0.051**	0.495
GeoTrans [10]	0.194	0.076	0.270
RoITr (*Ours*)	**0.023**	0.210	**0.233**

Table 6. Runtime comparison.

G. More Qualitative Results

Indoor Scenes: 3DLoMatch. We show more qualitative results on the challenging 3DLoMatch benchmark in Fig. 2.
Deformable Objects: 4DLoMatch. More qualitative results of the 4DLoMatch benchmark consisting of partially-scanned deformable objects are demonstrated in Fig. 3.

H. Runtime

We show the runtime comparison with Lepard [8] and GeoTrans [10] in Tab. 6. We run all the methods on a machine with a single Nvidia RTX 3090 GPU and an AMD Ryzen 5800X 3.80GHz CPU. All the models are tested without CPU parallel and with a batch size of 1. All the reported time is averaged over the 3DMatch testing set that consists of 1,623 point cloud pairs. The column "Data" counts the runtime for data preparation, and the column "Model" reports the time for generating descriptors from the prepared data. As shown in Tab. 6, RoITr has the highest data preparation and overall speed while the lowest model speed. That is mainly due to the relatively low speed of the attention mechanism compared to convolutions, e.g., KPConv [13] used in both Lepard and GeoTrans, and also because we do the Farthest Point Sampling (FPS) and k-nearest neighbor search on GPU, which is counted into the model time.

I. Limitations

Further Discussion. Although RoITr achieves remarkable performance on both the rigid and non-rigid scenarios, we also notice the drawbacks of our method. The first is the efficiency of the attention mechanism. Although our local attention mechanism runs faster compared to that of Point Transformer [20], its running speed is still lower than that of convolutions, as shown in Tab. 6. Moreover, the intrinsic rotation invariance comes at the cost of losing the ability to match symmetric structures (see the 4DLoMatch data of Fig. 4). Furthermore, RoITr mainly relies on feature distinctiveness to implicitly filter out the occluded areas during the matching procedure, which makes it fail in cases with extremely limited overlap (see the 3DLoMatch data of Fig. 4). Finally, as normal data augmentation cannot work on intrinsically rotation-invariant methods, more data is required to train a larger model.

Failure Cases. We further show some failure cases in Fig. 4. It can be observed that the failure on 3DLoMatch is caused by an extremely limited overlap on the flattened areas. In the first row, the overlap ratio is only 17.6%, and the overlap region is mainly on the floor. In the second row, the overlap region is even more limited (with an overlap ratio of 10.7%) and mainly on a wall. For the 4DLoMatch, the failure is mainly due to the extremely limited overlap and the ambiguity caused by the symmetric structure. The first row shows a case with the two frames of point cloud showing a horse's left and right parts, with only 18.1% overlap in the middle. The second row with 17.9% overlap ratio also has a strong left-right ambiguity due to the symmetric structure of a pig, which accounts for many left-right mismatches.

References

[1] Sheng Ao, Qingyong Hu, Bo Yang, Andrew Markham, and Yulan Guo. Spinnet: Learning a general surface descriptor for 3d point cloud registration. In *CVPR*, 2021. 4

[2] Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. Layer normalization. *arXiv preprint arXiv:1607.06450*, 2016. 3

[3] Tolga Birdal and Slobodan Ilic. Point pair features based object detection and pose estimation revisited. In *3DV*, 2015. 1

[4] Haowen Deng, Tolga Birdal, and Slobodan Ilic. Ppfnet: Global context aware local features for robust 3d point matching. In *CVPR*, 2018. 1

[5] Bertram Drost, Markus Ulrich, Nassir Navab, and Slobodan Ilic. Model globally, match locally: Efficient and robust 3d object recognition. In *CVPR*, 2010. 2

[6] Shengyu Huang, Zan Gojcic, Mikhail Usvyatsov, Andreas Wieser, and Konrad Schindler. Predator: Registration of 3d point clouds with low overlap. In *CVPR*, 2021. 1, 4

[7] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 1

[8] Yang Li and Tatsuya Harada. Lepard: Learning partial point cloud matching in rigid and deformable scenes. In *CVPR*, 2022. 1, 3, 5, 7

[9] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. *NeurIPS*, 2019. 1

[10] Zheng Qin, Hao Yu, Changjian Wang, Yulan Guo, Yuxing Peng, and Kai Xu. Geometric transformer for fast and robust point cloud registration. In *CVPR*, 2022. 1, 2, 3, 4, 5, 6

[11] Paul-Edouard Sarlin, Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. Superglue: Learning feature matching with graph neural networks. In *CVPR*, 2020. 3

[12] Yifan Sun, Changmao Cheng, Yuhan Zhang, Chi Zhang, Liang Zheng, Zhongdao Wang, and Yichen Wei. Circle loss: A unified perspective of pair similarity optimization. In *CVPR*, 2020. 3

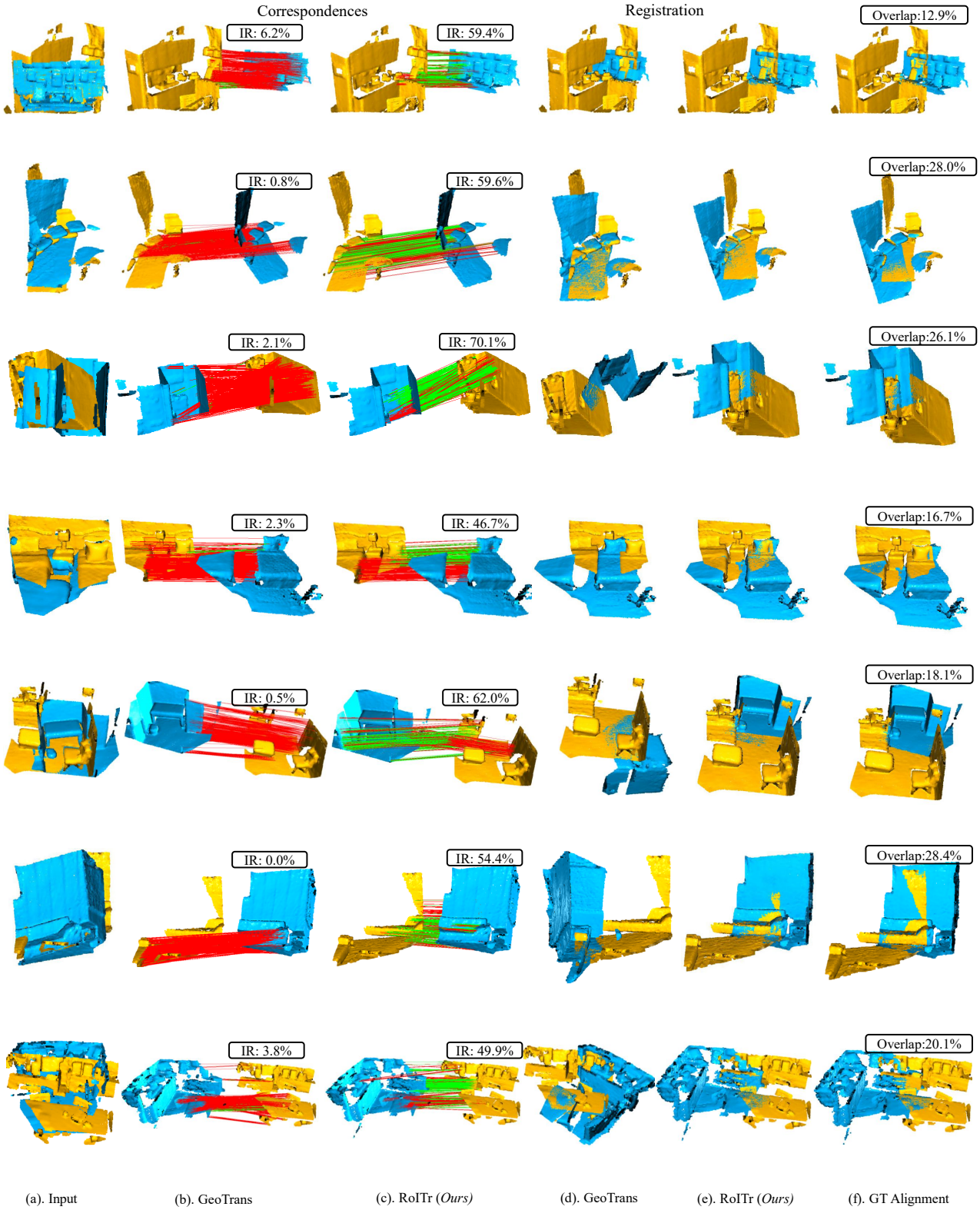

Figure 2. More qualitative results on 3DLoMatch. GeoTrans [10] is used as the baseline.

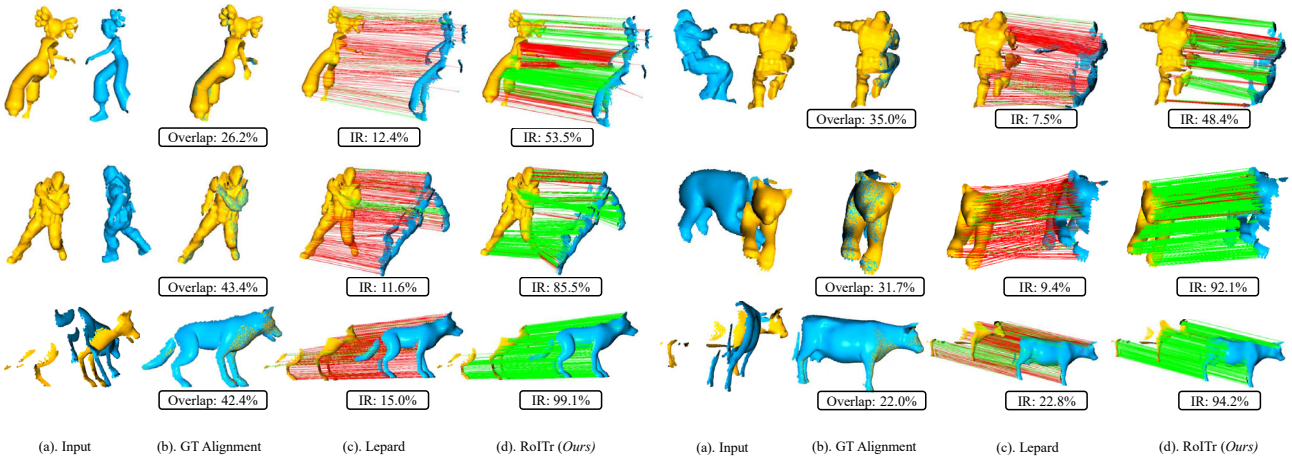

Figure 3. More qualitative results on 4DLoMatch. Lepard [8] is used as the baseline.

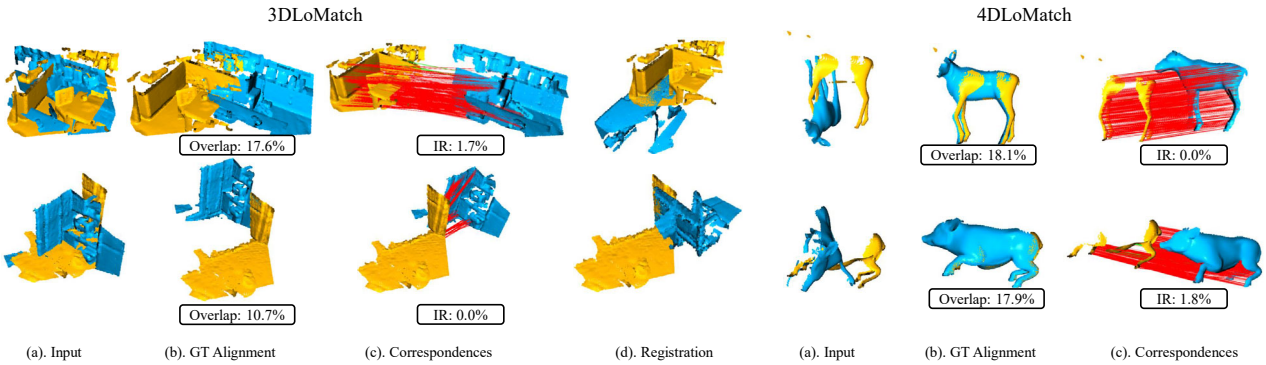

Figure 4. Failed cases on 3DLoMatch and 4DLoMatch.

[13] Hugues Thomas, Charles R Qi, Jean-Emmanuel Deschaud, Beatriz Marcotegui, François Goulette, and Leonidas J Guibas. Kpconv: Flexible and deformable convolution for point clouds. In *ICCV*, 2019. 4, 5

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

[15] Haiping Wang, Yuan Liu, Zhen Dong, and Wenping Wang. You only hypothesize once: Point cloud registration with rotation-equivariant descriptors. In *ACM MM*, 2022. 4

[16] Hao Yu, Ji Hou, Zheng Qin, Mahdi Saleh, Ivan Shugurov, Kai Wang, Benjamin Busam, and Slobodan Ilic. Riga: Rotation-invariant and globally-aware descriptors for point cloud registration. *arXiv preprint arXiv:2209.13252*, 2022. 4

[17] Hao Yu, Fu Li, Mahdi Saleh, Benjamin Busam, and Slobodan Ilic. Cofinet: Reliable coarse-to-fine correspondences for robust pointcloud registration. *NeurIPS*, 2021. 3, 4

[18] Wentao Yuan, Benjamin Eckart, Kihwan Kim, Varun Jampani, Dieter Fox, and Jan Kautz. Deepgmr: Learning latent gaussian mixture models for registration. In *ECCV*, 2020. 4

[19] Andy Zeng, Shuran Song, Matthias Nießner, Matthew Fisher, Jianxiong Xiao, and Thomas Funkhouser. 3dmatch: Learning local geometric descriptors from rgb-d reconstructions. In *CVPR*, 2017. 1

[20] Hengshuang Zhao, Li Jiang, Jiaya Jia, Philip HS Torr, and Vladlen Koltun. Point transformer. In *ICCV*, 2021. 4, 5