Supplementary Material
Learning Instance-Aware Correspondences for Robust Multi-Instance Point Cloud Registration in Cluttered Scenes

Zhiyuan Yu[1] Zheng Qin[2] Lintao Zheng[1] Kai Xu[1]
[1] National University of Defense Technology
[2] Defense Innovation Institute, Academy of Military Sciences

A. Implementation Details

A.1. Network Architecture

Backbone. We use a KPConv-FPN backbone [11] for feature extraction. We apply the grid subsampling scheme of [11] to downsample the point clouds and generate superpoints and dense points before feeding them into the network. The input point clouds are first downsampled with a voxel-grid filter of the size of 2.5cm on Scan2CAD, ShapeNet and 0.15cm on ROBI. We adopt a 4-stage backbone in all benchmarks. After each stage, the voxel size is doubled to further downsample the point clouds. The first and last (coarsest) levels of downsampled points represent the dense points and superpoints intended for matching. The detailed network configurations are shown in Tab. 1.

Instance-Aware Geometric Transformer. Given the superpoint features $\hat{\mathbf{F}}^{\mathcal{P}}$ and $\hat{\mathbf{F}}^{\mathcal{Q}}$ from the backbone, we first use a linear projection $\mathbf{W}_{\text{in}}$ to compress the feature dimension from 1024 to 256. $\mathbf{M}^{\mathcal{Q}}_{(0)}$ is initialized as all zeros. We adopt $N_t = 3$ instance-aware geometric transformer modules to iteratively extract superpoint features and predict instance masks:

$$\hat{\mathbf{F}}^{\mathcal{P}}_{\text{self},(0)} = \hat{\mathbf{F}}^{\mathcal{P}}\mathbf{W}_{\text{in}} \tag{1}$$

$$\hat{\mathbf{F}}^{\mathcal{Q}}_{\text{self},(0)} = \hat{\mathbf{F}}^{\mathcal{Q}}\mathbf{W}_{\text{in}} \tag{2}$$

$$\hat{\mathbf{F}}^{\mathcal{P}}_{\text{self},(t)} = \text{GeometricEncoder}(\hat{\mathcal{P}}, \hat{\mathbf{F}}^{\mathcal{P}}_{\text{cross},(t-1)}), \tag{3}$$

$$\hat{\mathbf{F}}^{\mathcal{Q}}_{\text{self},(t)} = \text{GeometricEncoder}(\hat{\mathcal{Q}}, \hat{\mathbf{F}}^{\mathcal{Q}}_{\text{cross},(t-1)}, \mathbf{M}^{\mathcal{Q}}_{(t-1)}), \tag{4}$$

$$\hat{\mathbf{F}}^{\mathcal{P}}_{\text{cross},(t)} = \text{CrossAtt}(\hat{\mathbf{F}}^{\mathcal{P}}_{\text{self},(t)}, \hat{\mathbf{F}}^{\mathcal{Q}}_{\text{self},(t)}), \tag{5}$$

$$\hat{\mathbf{F}}^{\mathcal{Q}}_{\text{cross},(t)} = \text{CrossAtt}(\hat{\mathbf{F}}^{\mathcal{Q}}_{\text{self},(t)}, \hat{\mathbf{F}}^{\mathcal{P}}_{\text{cross},(t)}) \tag{6}$$

$$\mathbf{M}^{\mathcal{Q}}_{(t)} = \text{InstanceMask}(\hat{\mathcal{Q}}, \hat{\mathbf{F}}^{\mathcal{Q}}_{\text{cross},(t)}, \mathbf{M}^{\mathcal{Q}}_{(t-1)}). \tag{7}$$

For the geometric structure embedding, we use $\sigma_d = 0.2$m on Scan2CAD, ShapeNet and $\sigma_d = 0.02$m on ROBI. We use the $\sigma_a = 15°$ on all benchmarks. In the geodesic embedding, we use $\sigma_{geo} = 0.1$m on Scan2CAD, ShapeNet and $\sigma_{geo} = 0.01$m on ROBI. Given the geodesic distance $G_{i,j}$ between $\hat{\mathbf{p}}_i$ and $\hat{\mathbf{p}}_j$, the pair-wise geodesic distance embedding $\mathbf{g}^G_{i,j}$ is computed as:

$$\begin{cases} \mathbf{g}^G_{i,j,2k} = \sin(\frac{G_{i,j}/\sigma_{geo}}{10000^{2k/d_t}}) \\ \mathbf{g}^G_{i,j,2k+1} = \cos(\frac{G_{i,j}/\sigma_{geo}}{10000^{2k/d_t}}) \end{cases}, \tag{8}$$

where d_t is the feature dimension. The geodesic embedding $\mathbf{g}_{i,j}$ is computed as:

$$\mathbf{g}_{i,j} = \mathbf{g}^G_{i,j}\mathbf{W}^G, \tag{9}$$

where $\mathbf{W}^G \in \mathbb{R}^{d_t \times d_t}$ is the projection matrices for the geodesic embedding.

At last, the final $\hat{\mathbf{Z}}^{\mathcal{P}}$ and $\hat{\mathbf{Z}}^{\mathcal{Q}}$ are obtained by adopting another linear projection with 256 channels.

$$\hat{\mathbf{Z}}^{\mathcal{P}} = \hat{\mathbf{F}}^{\mathcal{P}}_{\text{cross},(N_t)}\mathbf{W}_{\text{out}}, \tag{10}$$

$$\hat{\mathbf{Z}}^{\mathcal{Q}} = \hat{\mathbf{F}}^{\mathcal{Q}}_{\text{cross},(N_t)}\mathbf{W}_{\text{out}}. \tag{11}$$

A.2. Loss Functions

Our model is trained with three loss functions, an overlap-aware circle loss $\mathcal{L}_{\text{circle}}$, a negative log-likelihood loss $\mathcal{L}_{\text{nll}}$, and a mask prediction loss $\mathcal{L}_{\text{mask}}$. The overall loss is computed as: $\mathcal{L} = \mathcal{L}_{\text{circle}} + \mathcal{L}_{\text{nll}} + \mathcal{L}_{\text{mask}}$.

Overlap-aware Circle Loss. To supervise the superpoint feature features, we follow [8] and use the overlap-aware circle loss which weights the loss of each superpoint (patch) match according to their overlap ratio. Given the set of anchor patches $\mathcal{A}$, it consists of the patches in $\mathcal{Q}$ which have at least one positive patch in $\mathcal{P}$. For each anchor patch $\mathcal{G}^{\mathcal{Q}}_i \in \mathcal{A}$, we denote the set of its positive patches in $\mathcal{P}$ which share at least 10% overlap with $\mathcal{G}^{\mathcal{Q}}_i$ as ε^i_p, and the set of its negative patches which do not overlap with $\mathcal{G}^{\mathcal{Q}}_i$ as ε^i_n. The overlap-aware circle loss on $\mathcal{Q}$ is then computed as:

$$\mathcal{L}^{\mathcal{Q}}_{\text{circle}} = \frac{1}{|\mathcal{A}|}\sum_{\mathcal{G}^{\mathcal{Q}}_i \in \mathcal{A}} \log[1+\sum_{\mathcal{G}^{\mathcal{P}}_j \in \varepsilon^i_p} e^{\lambda^j_i \beta^{i,j}_p (d^j_i - \Delta_p)} . \sum_{\mathcal{G}^{\mathcal{P}}_k \in \varepsilon^i_n} e^{\beta^{i,k}_n (\Delta_n - d^k_i)}], \tag{12}$$

Stage	Scan2CAD	ROBI
	Backbone	

1	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)

3	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	NearestUpsampling UnaryConv(1536 → 512)	NearestUpsampling UnaryConv(1536 → 512)
6	NearestUpsampling UnaryConv(768 → 256)	NearestUpsampling UnaryConv(768 → 256)
	Instance-Aware Geometric Attention	
1	Linear(1024 → 256)	Linear(1024 → 256)
2	Geometric encoding block(256, 4) Cross-attention block(256, 4) Instance masking block(256, 4)	Geometric encoding block(256, 4) Cross-attention block(256, 4) Instance masking block(256, 4)
3	Geometric encoding block(256, 4) Cross-attention block(256, 4) Instance masking block(256, 4)	Geometric encoding block(256, 4) Cross-attention block(256, 4) Instance masking block(256, 4)
4	Geometric encoding block(256, 4) Cross-attention block(256, 4) Instance masking block(256, 4)	Geometric encoding block(256, 4) Cross-attention block(256, 4) Instance masking block(256, 4)
5	Linear(256 → 256)	Linear(256 → 256)

Table 1. Network architecture for Scan2CAD and ROBI.

where $d_i^j = \|\hat{\mathbf{h}}_i^{\mathcal{Q}} - \hat{\mathbf{h}}_j^{\mathcal{P}}\|_2$ is the distance in feature space, $\lambda_i^j = (o_i^j)^{\frac{1}{2}}$ and o_i^j is the overlap ratio between $\mathcal{G}_i^{\mathcal{P}}$ and $\mathcal{G}_j^{\mathcal{Q}}$. The weights $\beta_p^{i,j} = \gamma(d_i^j - \Delta_p)$ and $\beta_n^{i,k} = \gamma(\Delta_n - d_i^k)$ are determined individually for each positive and negative example, using the margin hyper-parameters $\Delta_p = 0.1$ and $\Delta_n = 1.4$. The loss $\mathcal{L}_{\text{circle}}^{\mathcal{P}}$ on $\mathcal{P}$ is computed in the same way. And the overall loss is $\mathcal{L}_{\text{circle}} = (\mathcal{L}_{\text{circle}}^{\mathcal{P}} + \mathcal{L}_{\text{circle}}^{\mathcal{Q}})/2$.

Negative Log-likelihood Loss. Following [9], we use a negative log-likelihood loss on the assignment matrix $\bar{\mathbf{Z}}_i$ of each ground-truth superpoint correspondence $\hat{\mathcal{C}}_i^*$.

For each $\hat{\mathcal{C}}_i$, we calculate the inlier ratio between matched patches using each ground-truth transformation. Subsequently, we select the transformation corresponding to the highest inlier ratio to estimate a set of ground-truth point correspondences $\mathcal{C}_i$ with a matching radius τ. The point matching loss for $\hat{\mathcal{C}}_i^*$ is computed as:

$$\mathcal{L}_{\text{nll},i} = -\sum_{(x,y)\in\mathcal{C}_i^*} \log \bar{z}_{x,y}^i - \sum_{x\in\mathcal{I}_i} \log \bar{z}_{x,m_i+1}^i - \sum_{y\in\mathcal{J}_i} \log \bar{z}_{n_i+1,y}^i \tag{13}$$

where $\mathcal{I}_i$ and $\mathcal{J}_i$ are the unmatched points in the two matched patches. The final loss is the average of the loss over all sampled superpoint matches: $\mathcal{L}_{\text{nll}} = \frac{1}{N_g}\sum_{i=1}^{N_g} \mathcal{L}_{p,i}$.

Mask Prediction Loss. Following [6], the mask prediction loss consists of the binary cross-entropy (BCE) loss and the dice loss with Laplace smoothing [6], which is defined as follows:

$$\mathcal{L}_{\text{mask},i} = \text{BCE}(m_i, m_i^{gt}) + 1 - 2\frac{m_i \cdot m_i^{gt} + 1}{|m_i| + |m_i^{gt}| + 1} \tag{14}$$

where m_i and m_i^{gt} are the predicted and the ground-truth instance masks, respectively. The final loss is the average loss over all superpoints: $\mathcal{L}_{\text{mask}} = \frac{1}{N_g}\sum_{i=1}^{N_g} \mathcal{L}_{\text{mask},i}$.

A.3. Training and Testing Settings

MIRETR is implemented in Pytorch [7] with an NVIDIA RTX 3090Ti. We train MIRETR using Adam optimizer [3] for $60/60/40$ epochs , with initial learning rate 10^{-4}, momentum 0.98, and weight decay 10^{-6} . The learning rate is exponentially decayed by 0.05 after each epoch.

We use the matching radius of $\tau = 5$cm for Scan2CAD, ShapeNet and $\tau = 0.3$cm for ROBI to determine overlapping during the generation of both superpoint-level and point-level ground-truth matches. The number of neighbors is set to 16 for Scan2CAD, ShapeNet, and 32 for ROBI. The confidence threshold τ of the mask score is set to 0.6. We use the same data augmentation as in [2, 8, 13]. During training, We sample $N_g = 128$ ground-truth superpoint matches. We generate the ground-truth masks for the instance-aware geometric transformer module and the Instance Candidate Generation module. We calculate the loss between ground-truth masks and the predicted masks in each iteration of the attention module. During testing, We sample $N_c = 128$ superpoint matches. During candidate selection and refinement, we set the threshold of similarity score as 0.7 for ROBI and 0.8 for Scan2CAD, ShapeNet. We calculate the number of max inliers within transformations and remove the transformation whose inliers is fewer than $\tau_3 \cdot max\_inlier$. We set τ_3 as 0.2 for ROBI and 0.8 for Scan2CAD, ShapeNet. The acceptance radius τ_2 is 5cm for Scan2CAD, ShapeNet and 0.3cm for ROBI.

B. Metrics

Following [10, 14], we evaluate our method with three registration metrics: (1) *Mean Recall*, (2) *Mean Precision* and (3) *Mean F_1 score*. We also report *Inlier Ratio* (IR) and *mean Intersect over Union* (mIoU).

Mean Recall (MR) is the ratio of registered instances over all ground-truth instances. For a pair of source point cloud and target point cloud, `recall` is computed as:

$$\texttt{recall} = \frac{1}{|I^{\text{gt}}|}\sum_{s=1}^{|I^{\text{gt}}|} I_s^{\text{gt}}, \tag{15}$$

where $I_s^{\text{gt}} = \{0, 1\}$ represents whether a ground-truth transformation is successfully registered. For non-symmetric instances, the registration is considered successful when the

RRE $\leqslant 15°$, RTE $\leqslant 0.1$m on Scan2CAD dataset, and RRE $\leqslant 15°$, RTE $\leqslant 0.006$m on ROBI dataset. For symmetric instances, the registration is considered successful when the ADD-S $\leqslant 0.1$ on both datasets. The mean recall (MR) is the average of all `recall` in test set.

Mean Precision (MP) is the ratio of registered instances overall predicted instances. For a pair of source point cloud and target point cloud, `precision` is computed as:

$$\texttt{precision} = \frac{1}{|I^{\text{pred}}|} \sum\nolimits_{s=1}^{|I^{\text{pred}}|} I_s^{\text{pred}}, \tag{16}$$

where $I_s^{\text{pred}} = \{0, 1\}$ represents whether a predicted transformation is successfully registered. The mean precision (MP) is the average of all `precision` in test set.

Mean F_1 score (MF) is the harmonic mean of MP and MR. The mean `F1` Score (MF) is computed as:

$$\texttt{MF} = \frac{2 \cdot \texttt{MR} \cdot \texttt{MP}}{\texttt{MR} + \texttt{MP}} \tag{17}$$

Inlier Ratio (IR) is the ratio of inlier correspondences among putative correspondences. Given point correspondences $\mathcal{C}$, IR is computed as:

$$\text{IR} = \frac{1}{|\mathcal{C}|} \sum_{(\mathbf{p},\mathbf{q}) \in \mathcal{C}} [\![\|\bar{\mathbf{T}}_k(\mathbf{p}) - \mathbf{q}\|_2 < \tau_1]\!], \tag{18}$$

where $[\![\cdot]\!]$ is the Iversion bracket, $\mathbf{T}_k$ is a ground-truth transformation. We set $\tau_1 = 0.05m$ on Scan2CAD, ShapeNet datasets and $\tau_1 = 0.005m$ on ROBI dataset.

Mean Intersect over Union (mIoU) is the ratio of the intersect of the predicted and the ground-truth instance masks over the union of them, which measures the quality of the predicted masks. Given the predicted and the ground-truth instance masks $\mathbf{M}^{\text{pred}}$ and $\mathbf{M}^{\text{gt}}$, the mIoU of $\mathbf{M}^{\text{pred}}$ is computed as:

$$\text{mIoU} = \frac{\text{Intersect}(\mathbf{M}^{\text{pred}}, \mathbf{M}^{\text{gt}})}{\text{Union}(\mathbf{M}^{\text{pred}}, \mathbf{M}^{\text{gt}})}. \tag{19}$$

C. Additional Experiments

C.1. Evaluation Results on ModelNet40

Dataset. ModelNet40 [12] consists of 12311 CAD models of man-made objects from 40 different categories. To verify the generalization ability of MIRETR, we train our method and competitors using 5112 point clouds from 20 categories and test on 1266 point clouds from the other 20 categories. For each meshed CAD model, we downsample 4096 points from it to form the source point cloud and generate 4-16 transformations to form the target point cloud.

Results. In Tab. 2,our method achieves a score of 99.94 in MF, demonstrating the capability of our model to estimate the poses of previously unseen objects. Additionally, MIRETR surpasses existing methods in MR, MP, and MF. The multi-model fitting methods that take our correspondences as input outperform those using CoFiNet and GeoTransformer, indicating a higher inlier ratio.

Model	IR(%)	MR(%)	MP(%)	MF(%)
CofiNet [13]+T-Linkage [1]	43.01	11.38	9.12	10.12
CofiNet [13]+RansaCov [1]		15.13	12.89	13.92
CofiNet [13]+PointCLM [14]		42.37	38.50	40.34
CofiNet [13]+ECC [10]		65.31	50.24	56.79
GeoTransformer [8]+T-Linkage [4]	58.63	33.03	31.01	31.98
GeoTransformer [8]+RansaCov [5]		40.21	48.92	44.13
GeoTransformer [8]+PointCLM [14]		86.14	85.37	85.75
GeoTransformer [8]+ECC [10]		82.49	79.26	80.84
MIRETR (*ours*) +T-Linkage [4]	63.04	40.03	43.39	41.64
MIRETR (*ours*) +RansaCov [5]		46.03	49.58	47.73
MIRETR (*ours*) +PointCLM [14]		99.48	98.08	98.77
MIRETR (*ours*) +ECC [10]		98.48	98.11	98.29
MIRETR (*ours*, full pipeline)		**99.95**	**99.93**	**99.94**

Table 2. Evaluation results on Modelnet40.

Model	IR(%)	MR(%)	MP(%)	MF(%)
FCGF [1]+T-Linkage [4]	9.63	17.93	5.61	8.54
FCGF [1]+RansaCov [5]		21.58	9.86	13.53
FCGF [1]+PointCLM [14]		36.28	18.81	24.77
FCGF [1]+ECC [10]		54.77	50.34	52.46
CofiNet [13]+T-Linkage [1]	26.73	44.31	12.07	18.97
CofiNet [13]+RansaCov [1]		55.73	27.33	36.67
CofiNet [13]+PointCLM [14]		44.41	50.85	47.41
CofiNet [13]+ECC [10]		74.58	26.28	38.86
GeoTransformer [8]+T-Linkage [4]	54.05	70.22	54.52	61.38
GeoTransformer [8]+RansaCov [5]		75.05	70.15	72.51
GeoTransformer [8]+PointCLM [14]		78.95	80.80	79.86
GeoTransformer [8]+ECC [10]		87.88	72.37	79.37
MIRETR (*ours*) +T-Linkage [4]	57.40	74.12	47.61	57.97
MIRETR (*ours*) +RansaCov [5]		79.34	73.07	76.07
MIRETR (*ours*) +PointCLM [14]		81.36	84.30	82.80
MIRETR (*ours*) +ECC [10]		**92.79**	75.68	83.36
MIRETR (*ours*, full pipeline)		88.06	**84.53**	**86.26**

Table 3. Evaluation results on single-frame Scan2CAD.

C.2. More Evaluation Results on Scan2CAD

Dataset. We further evaluate the performance of MIRETR in incomplete scenes by using scene point clouds reconstructed from single-frame RGBD data. Unlike [14], we do not replace the object instances with the CAD models but use the original incomplete point clouds. We select 6408 RGBD frames which contain multiple instances for evaluating the performance of the model under occlusion. These frames contain 807 objects where 564 are used for training, 80 for validation, and 163 for testing.

Results. As Tab. 3 shows, our method achieves the highest mean precision and mean F1 score. Due to the limited number of instances within the Scan2CAD dataset, the methods based on spatial consistency, such as ECC and PointCLM, can effectively filter noise and cluster instances. Therefore, the combination of our method with ECC achieved the highest average recall. However, the multi-model fitting meth-

Model	Time(s)			MF(%)
	Model	Pose	Total	
CofiNet [13]+T-Linkage [4]	0.13	3.19	3.32	0.93
CofiNet [13]+RansaCov [5]	0.13	0.16	0.29	1.31
CofiNet [13]+PointCLM [14]	0.13	0.70	0.83	1.50
CofiNet [13]+ECC [10]	0.13	0.15	0.28	4.66
GeoTransformer [8]+T-Linkage [4]	0.09	3.13	3.22	5.73
GeoTransformer [8]+RansaCov [5]	0.09	0.17	0.26	10.81
GeoTransformer [8]+PointCLM [14]	0.09	0.22	0.31	18.33
GeoTransformer [8]+ECC [10]	0.09	0.17	0.26	23.20
MIRETR (*ours*) +T-Linkage [4]	0.30	3.07	3.34	11.20
MIRETR (*ours*) +RansaCov [5]	0.30	0.17	0.47	18.38
MIRETR (*ours*) +PointCLM [14]	0.30	0.33	0.63	25.48
MIRETR (*ours*) +ECC [10]	0.30	0.21	0.51	28.91
MIRETR (*ours, full pipeline*)	0.30	0.10	0.40	39.80

Table 4. Times on ROBI. The *model time* is the time for correspondence extraction, while the *pose time* is for transformation estimation.

Model	MR(%)	MP(%)	MF(%)	mIOU (%)
(a) GME & GME	36.63	42.16	39.20	69.07
(b) None & GDE	36.41	42.30	39.13	67.38
(c) GME & GDE (*ours*)	38.51	41.19	39.80	69.26
(d) GME w/o mask	38.44	36.83	37.62	65.83
(e) GME w/ mask (*ours*)	38.51	41.19	39.80	69.26
(f) $[\ \mathbf{y}_{i_j}; \mathbf{y}_i;\ \mathbf{g}_{i,j}\ ]$	36.11	41.60	38.66	69.20
(g) $[\ \mathbf{y}_{i_j} - \mathbf{y}_i\ ]$	37.80	43.16	40.20	67.13
(h) $[\ \mathbf{y}_{i_j} - \mathbf{y}_i\ ;\ \mathbf{g}_{i,j}\ ]$ (*ours*)	38.51	41.19	39.80	69.26

Table 5. Ablation studies of the instance-aware geometric transformer on ROBI. **GME**: geometric embedding. **GDE**: geodesic embedding.

# Neighbors	# Inst	MR(%)	MP(%)	MF(%)
4	22.69	40.25	25.39	31.14
8	20.07	41.18	29.53	34.92
16	15.96	40.45	35.66	37.91
32 (*ours*)	13.70	38.51	41.19	39.80
48	13.01	37.59	43.83	40.47
64	12.60	37.13	45.09	40.75

Table 6. Ablation studies of the number of neighbors on ROBI.

ods integrated with our model achieve superior performance compared to those combined with other point cloud registration methods.

C.3. Time Evaluation

we study the time efficiency of MIRETR on ROBI in Tab. 4. The model time is the time for correspondence extraction, and the pose time is for transformation estimation. As shown in Tab. 4, while extracting correspondences, our method is slower than CofiNet and GeoTransformer. However, our method achieves faster pose estimation with 3 times acceleration over PointCLM and 2 times over ECC, which demonstrates the time efficiency of our method.

C.4. Additional Ablation Studies

Instance-aware geometric transformer. Tab. 5 demonstrates more ablation studies of the instance-aware geometric transformer.

We first study the influence of different embedding methods in the instance-aware geometric transformer module. We compare three methods: (a) the geometric embedding in both the geometric encoding block and the instance masking block, (b) no embedding in the geometric encoding block and the geodesic embedding in the instance masking block, and (c) the geometric embedding in both the geometric encoding block and the geodesic embedding in the instance masking block. achieves the best MF, which means the model (a) and (b) extract fewer instances than MIRETR.

We further study the effectiveness of the masking mechanism in the geometric encoding block. Ablating the masking mechanism (d) leads to a significant drop on MP as the superpoint features are polluted by the context outside the instances.

At last, we compare three methods for predicting confidence score of instance masks: (f) feature concatenation and geodesic embedding $[\ \mathbf{y}_{i_j}; \mathbf{y}_i;\ \mathbf{g}_{i,j}\ ]$, (g) feature residuals $[\ \mathbf{y}_{i_j} - \mathbf{y}_i\ ]$, and (h) features residuals and geodesic embedding $[\ \mathbf{y}_{i_j} - \mathbf{y}_i;\ \mathbf{g}_{i,j}\ ]$. It can be observed that the three methods performs comparably.

Neighbors. We study the influence of different numbers of neighbors in Tab. 6. Along with the decreasing number of neighbors, the performance of the model gradually decreases, especially for MP. When the number of neighbors is small, the Instance Candidate Generation module tends to extract more instances but obtain more wrong transformations, resulting in high recall, and low precision.

Candidate selection and refinement. We first replace the NMS-based filtering in the candidate selection and refinement with random sampling in Tab. 7, leading to a significant drop on MR. And note that the increase on MP is due to the duplicated registrations.

We conduct the sensitivity analysis on the similarity threshold in NMS in Tab. 7. τ_s controls which instance candidates should be merged. When τ_s goes from 0.9 to 0.5, the MP increases (34.77 to 46.18) but the MR drops (40.20 to 35.30), showing that a too small τ_s could not remove all duplicated instances.

C.5. More Qualitative Results

We provide more qualitative results in Fig. 1 for ROBI. In Fig. 1, our method detects more objects than GeoTransformer [8], especially in the low-overlap, clutter scenarios.

D. Limitations

In multi-instance scenarios, the amplitude of object rotation changes is greater than in traditional point cloud registra-

Model	MR(%)	MP(%)	MF(%)
MIRETR w/ random sampling	28.94	44.71	37.24
MIRETR w/ NMS (*ours*)	38.51	41.19	39.80
NMS 0.9	40.20	34.77	37.29
NMS 0.8	39.35	36.02	38.67
NMS 0.7 (*ours*)	38.51	41.19	39.80
NMS 0.6	36.83	43.55	39.91
NMS 0.5	35.30	46.18	40.20

Table 7. Ablation studies of the candidate selection and refinement module on ROBI.

tion. However, KpConv[11] struggles to obtain rotation-invariant features for point matching, which limits our performance.

In superpoint matching, MIRETR is hampered by the issue of uneven sampling of instances. In some scenarios, MIRETR may sample different points on the same instance.

MIRETR is hard to handle extreme clustering and severe occlusion data as shown in Fig. 2

References

[1] Christopher Choy, Jaesik Park, and Vladlen Koltun. Fully convolutional geometric features. In *CVPR*, pages 8958–8966, 2019. 3

[2] Shengyu Huang, Zan Gojcic, Mikhail Usvyatsov, Andreas Wieser, and Konrad Schindler. Predator: Registration of 3d point clouds with low overlap. In *CVPR*, pages 4267–4276, 2021. 2

[3] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. 2015. 2

[4] Luca Magri and Andrea Fusiello. T-linkage: A continuous relaxation of j-linkage for multi-model fitting. In *CVPR*, pages 3954–3961, 2014. 3, 4

[5] Luca Magri and Andrea Fusiello. Multiple model fitting as a set coverage problem. In *CVPR*, pages 3318–3326, 2016. 3, 4

[6] Fausto Milletari, Nassir Navab, and Seyed-Ahmad Ahmadi. V-net: Fully convolutional neural networks for volumetric medical image segmentation. In *3DV*, pages 565–571, 2016. 2

[7] 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*, 32:8026–8037, 2019. 2

[8] Zheng Qin, Hao Yu, Changjian Wang, Yulan Guo, Yuxing Peng, and Kai Xu. Geometric transformer for fast and robust point cloud registration. In *CVPR*, pages 11143–11152, 2022. 1, 2, 3, 4

[9] Paul-Edouard Sarlin, Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. Superglue: Learning feature matching with graph neural networks. In *CVPR*, pages 4938–4947, 2020. 2

[10] Weixuan Tang and Danping Zou. Multi-instance point cloud registration by efficient correspondence clustering. In *CVPR*, pages 6667–6676, 2022. 2, 3, 4

[11] 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*, pages 6411–6420, 2019. 1, 5

[12] Zhirong Wu, Shuran Song, Aditya Khosla, Fisher Yu, Linguang Zhang, Xiaoou Tang, and Jianxiong Xiao. 3d shapenets: A deep representation for volumetric shapes. In *CVPR*, pages 1912–1920, 2015. 3

[13] Hao Yu, Fu Li, Mahdi Saleh, Benjamin Busam, and Slobodan Ilic. Cofinet: Reliable coarse-to-fine correspondences for robust point cloud registration. *arXiv preprint arXiv:2110.14076*, 2021. 2, 3, 4

[14] Mingzhi Yuan, Zhihao Li, Qiuye Jin, Xinrong Chen, and Manning Wang. Pointclm: A contrastive learning-based framework for multi-instance point cloud registration. In *ECCV*, pages 595–611, 2022. 2, 3, 4

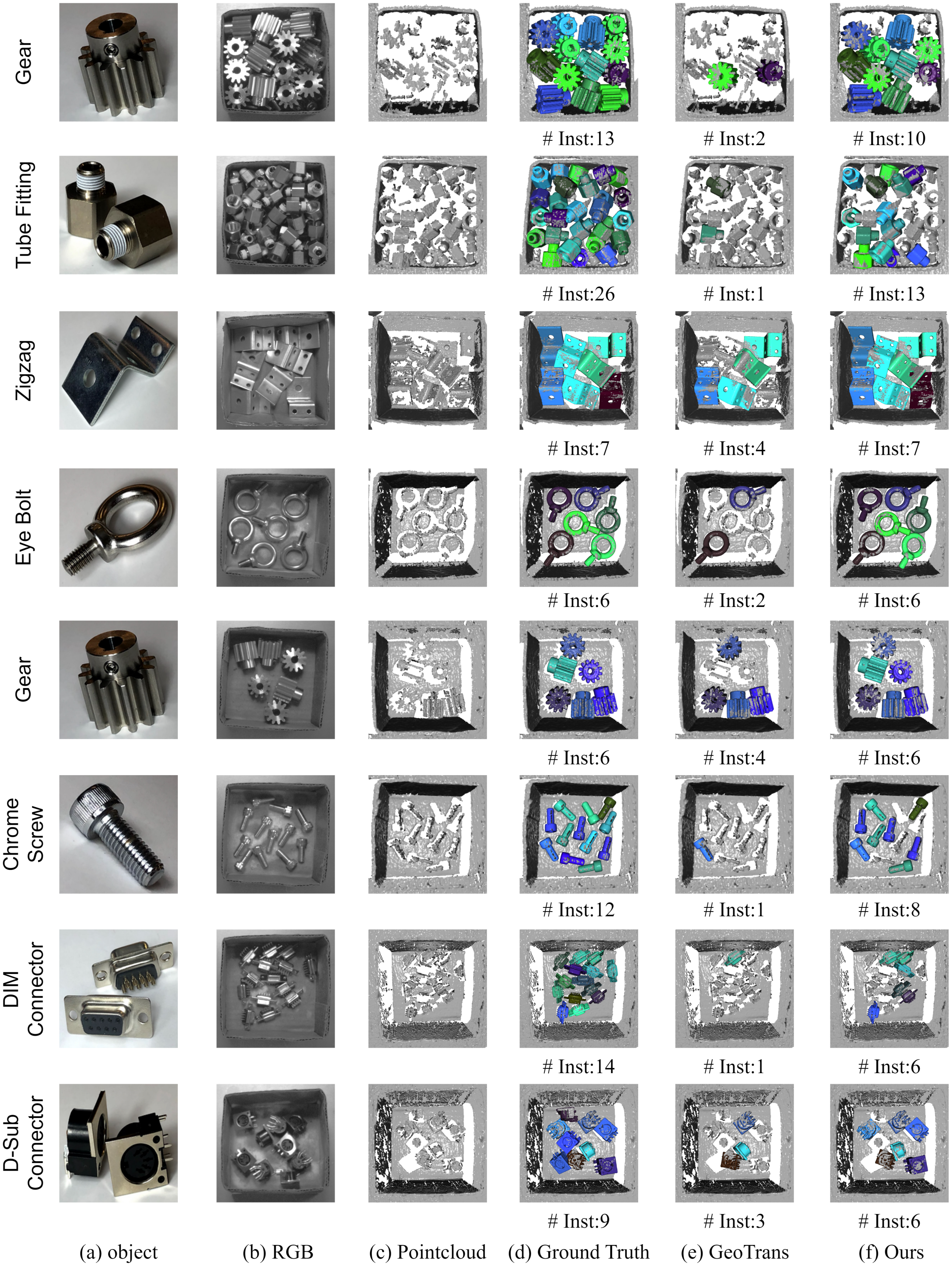

Figure 1. Results on the ROBI dataset. The gray point clouds represent the target point cloud and the point clouds in other colors represent the source point cloud with different transformations.

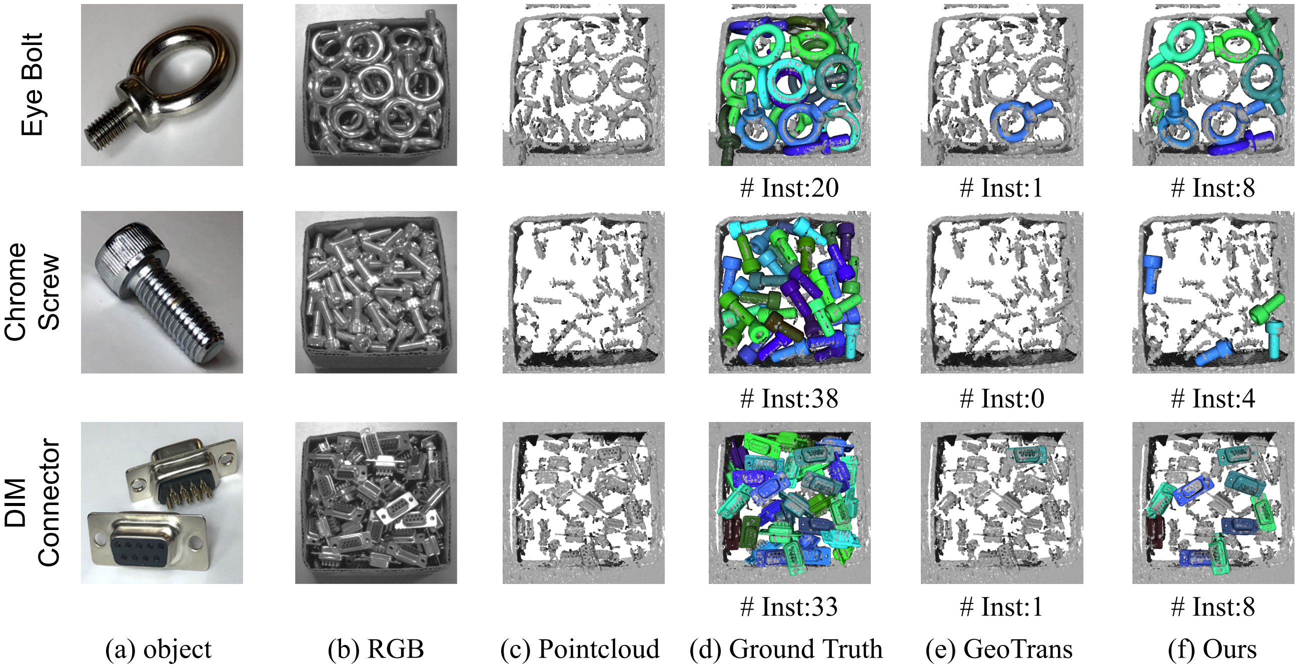

Figure 2. Fail cases of the ROBI dataset. The gray point clouds represent the target point cloud and the point clouds in other colors represent the source point cloud with different transformations.