3D Focusing-and-Matching Network for Multi-Instance Point Cloud Registration

Liyuan Zhang, Le Hui*, Qi Liu, Bo Li, Yuchao Dai*
School of Electronics and Information, Northwestern Polytechnical University
Shaanxi Key Laboratory of Information Acquisition and Processing
`zhangliyuannpu@mail.nwpu.edu.cn`, `{huile, liuqi, libo, daiyuchao}@nwpu.edu.cn`

Abstract

Multi-instance point cloud registration aims to estimate the pose of all instances of a model point cloud in the whole scene. Existing methods all adopt the strategy of first obtaining the global correspondence and then clustering to obtain the pose of each instance. However, due to the cluttered and occluded objects in the scene, it is difficult to obtain an accurate correspondence between the model point cloud and all instances in the scene. To this end, we propose a simple yet powerful 3D focusing-and-matching network for multi-instance point cloud registration by learning the multiple pair-wise point cloud registration. Specifically, we first present a 3D multi-object focusing module to locate the center of each object and generate object proposals. By using self-attention and cross-attention to associate the model point cloud with structurally similar objects, we can locate potential matching instances by regressing object centers. Then, we propose a 3D dual-masking instance matching module to estimate the pose between the model point cloud and each object proposal. It performs instance mask and overlap mask masks to accurately predict the pair-wise correspondence. Extensive experiments on two public benchmarks, Scan2CAD and ROBI, show that our method achieves a new state-of-the-art performance on the multi-instance point cloud registration task. The project page is at `https://npucvr.github.io/3DFMNet/`.

1 Introduction

Point cloud registration, a fundamental process in computer vision, involves aligning two point clouds through estimating a rigid transformation. In practical applications like robotic bin picking, multi-instance registration emerges as a critical need, demanding the alignment of a model's point cloud with multiple instances within the scene. This task presents heightened complexity compared to single-point cloud registration, primarily due to challenges such as the uncertain number of instances and inter-instance occlusions. These complexities are particularly pronounced in cluttered environments, where precise alignment becomes pivotal for effective robotic operations. Therefore, how to improve the accuracy of multi-instance point cloud registration is still a challenging issue.

There are a few efforts for tackling multi-instance point cloud registration. Existing pipelines can be roughly divided into two types: two-stage and one-stage. For the two-stage process, we first extract point correspondences between the model point cloud and scene point clouds, and then recover per-instance transformations through multi-model fitting [24, 36, 48]. Although two-stage methods are simple and feasible, the success of these methods largely depends on the quality of the

---

*Corresponding authors.
Liyuan Zhang, Le Hui, Qi Liu, Bo Li, and Yuchao Dai are with the Key Lab of Shaanxi Key Laboratory of Information Acquisition and Processing, School of Electronics And Information, Northwestern Polytechnical University, China.

38th Conference on Neural Information Processing Systems (NeurIPS 2024).
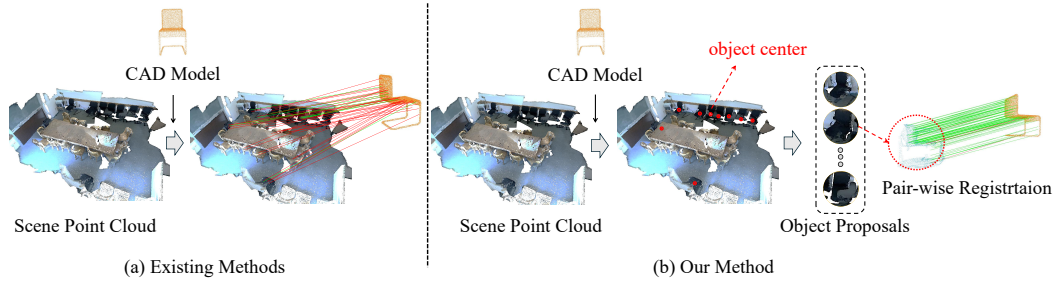
Figure 1: Comparison between our method and existing methods in multi-instance point cloud registration. Our method decomposes the multi-instance point cloud registration into multiple pairwise point cloud registration.

correspondence. Furthermore, due to cluttered and occluded objects, it is still difficult to accurately cluster the correspondences into individual instances for subsequent pair-wise registration. For the one-stage process, it takes the model point cloud and scene point cloud as inputs, and directly outputs pose. As a representative one-stage work, Yu *et al.*[46] proposed a coarse-to-fine framework, which learns to extract instance-aware correspondences for estimating transformations without multi-model fitting. Due to the consideration of instance-level information in correspondence, it can obtain fine-grained features, thereby boosting the performance. However, for the scene with multiple objects, obtaining accurate instance-level correspondence is very difficult, especially for the cluttered and occluded objects. Therefore, to alleviate the difficulty of learning correspondence between the model point cloud and multiple objects in the scene, as shown in Figure 1, we consider first focusing on the object centers, and then learning the matching between the object proposal and the model point cloud.

In this paper, we propose a simple yet powerful 3D focusing-and-matching network for multi-instance point cloud registration. The core idea of our method is to decompose the multi-instance point cloud registration into multiple pair-wise point cloud registrations. Specifically, we propose a 3D multi-object focusing module to localize the potential object centers and generate object proposals. To associate the object with the input CAD model, we use self-attention and cross-attention to learn the structurally similar features, thereby improving the accuracy of prediction for object centers. Based on the learned object center, we incorporate the radius of the CAD model to generate object proposals through ball query operation. After that, we propose a 3D dual-masking instance matching module to learn accurate pair-wise registration between the CAD model and object proposal. It adopts an instance mask to filter the background points in the object proposal and uses an overlap mask to improve the pair-wise partial registration of incomplete objects.

In summary, our contributions lie in three aspects:

1. Our primary contribution does not lie in the network architecture but rather in proposing a new pipeline to address the multi-instance point cloud registration problem. Existing methods (such as PointCLM [48] and MIRETR [46]) mainly learn correspondence between the one CAD model and multiple objects (one-to-many paradigm), while our method decompose the one-to-many paradigm into multiple pair-wise point cloud registration (multiple one-to-one paradigm) by first detecting the object centers and then learning the matching between the CAD model and each object proposal.

2. Our new pipeline is simple yet powerful, achieving the new state-of-the-art on both Scan2CAD [3] and ROBI [43] datasets. Especially on the challenging ROBI dataset, our method significantly outperforms the previous SOTA MIRETR by about 7% in terms of MR, MP, and MF.

3. The progressive decomposition approach of transforming multi-instance point cloud registration into multiple pair-wise registrations, as proposed in our paper, also holds significant insights for other tasks, such as multi-target tracking and map construction.

2

2 Related Work

Point Cloud Registration is a crucial task in fields such as robotics and autonomous driving, which usually involving three stages: point matching, outlier rejection, and pose estimation. Acquiring accurate point correspondences is critical for successful registration, making the first two stages particularly important. Accurate point matching deeply relies on features that are descriptive and rotation invariant. Many researchers have made efforts on this, including handcrafted descriptors[32, 6] and learning-based descriptors[1, 10, 49, 19, 20, 41, 2]. Some recent coarse-to-fine frameworks[18, 31] bypass keypoint detection and achieve accurate correspondences in large-scale scenes. To handle the problem of outliers, RANSAC[15] and its variants[7, 4]) follow the hypothesis-and-verification process to reject outliers. And some learning-based methods[11, 16, 42, 34] for eliminating outliers have also been proposed for robust pose estimation. On the other hand, several methods[39, 17, 40, 44, 38, 23, 45] directly estimate the transformation with a neural network in an end-to-end manner. However, these point cloud registration methods almost focus on the one-to-one problem which only need to solve the transformation between two point clouds. So they cannot directly work when faced with the challenge of a large number of instances in multi-instance registration tasks and heavy intra-instance occlusion.

Multi-Instance Point Cloud Registration, which aligns a source point cloud to its various instances within a target point cloud, has received relatively less attention. Unlike multi-way registration[35], which aims to create a globally consistent reconstruction from multiple fragments through pairwise registration[30], multi-instance registration involves not only rejecting outliers from noisy correspondences but also identifying the inlier set for each individual instance. This makes it even more challenging than the traditional registration problem. Early methods of multi-model fitting were used for this task. In the early stages of this task's development, various methods of multi-model fitting were employed. RANSAC-based approaches[24, 5, 13, 22] followed a hypothesis verification approach to fit multiple models, while another method, based on clustering[26, 27, 25, 36, 48, 8, 47], entailed sampling an extensive set of hypotheses and clustering the correspondences based on their residuals under these hypotheses. Recently, [46] proposed an instance-aware correspondence extraction method for end-to-end multi-instance pose estimation. However, existing methods are often affected by outliers from other instances and highly rely on scene-specific global feature extraction. In this approach, the multi-instance registration problem is addressed by detecting matching instance centers in the scene and subsequently splitting the instances, effectively transforming it into a pairwise point cloud registration problem.
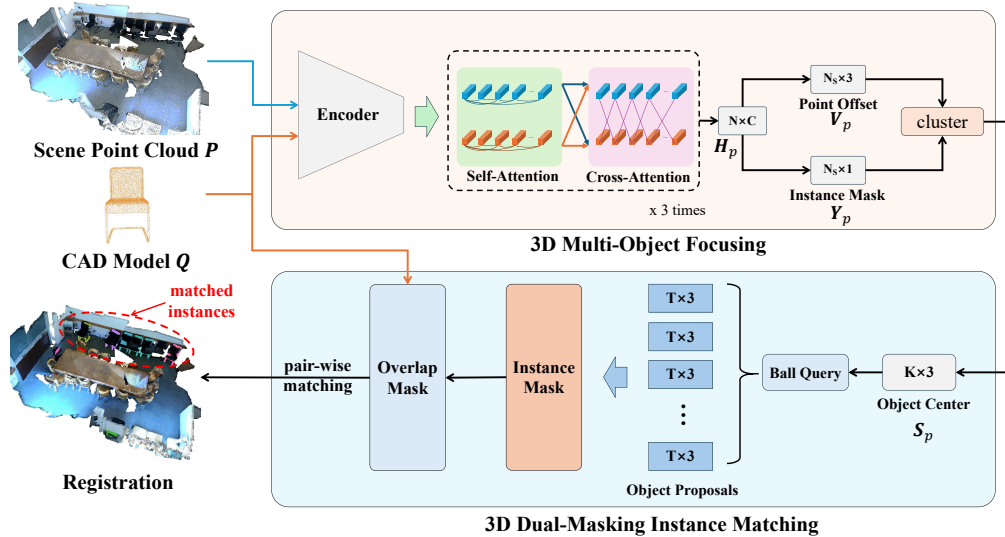
Figure 2: The framework of our 3D focusing-and-matching network for multi-instance pint cloud registration. Given the scene point cloud and the CAD model, we first present the 3D multi-object focusing module to localize the centers of the potential objects in the scene. Then, we design the 3D dual-masking instance matching module to learn pair-wise point cloud registration from the localized object proposals.

3

3 Method

The overall pipeline is illustrated in Figure 2. Our method is a two-stage framework, which first localizes the center of each object and then performs pair-wise correspondence. For the first stage, we present a 3D multi-object focusing module (Sec. 3.1) for detecting the potential instance centers by learning the correlation between the input model point cloud and the scene point cloud. For the second stage, we design a 3D dual-masking instance matching module (Sec. 3.2) for predicting pair-wise correspondence between the input model point cloud and the localized region of each object center. At last, we introduce the loss functions of our method (Sec. 3.3).

3.1 3D Multi-Object Focusing

As the first stage of our method, the 3D multi-object focusing module aims to regress the center of the potential objects for generating high-quality proposals for pair-wise correspondence. Compared to predicting the bounding box or mask of an instance, directly regressing the center of the object is much easier, especially in cluttered and occluded scenes. In order to accurately detect the object center, we first learn the correlation between the model point cloud and the scene point cloud. Then, we predict the object center by learning the offset of each point. Finally, we introduce how to construct 3D object proposals for subsequent pair-wise point cloud registration.

Feature correlation learning. We design a simple yet efficient feature extraction structure to learn the correlation between the scene point cloud $\boldsymbol{P} \in \mathbb{R}^{N \times 3}$ and the model point cloud $\boldsymbol{Q} \in \mathbb{R}^{M \times 3}$, where N and M are the numbers of their points, respectively. Note that for a fair comparison, we do not use RGB information of point cloud. Before learning correlation, we adopt the encoder of KPConv [37] to extract multi-scale point features of $\boldsymbol{P}$ and $\boldsymbol{Q}$, respectively. The output feature maps are denoted by $\boldsymbol{F}_p \in \mathbb{R}^{N_s \times C}$ and $\boldsymbol{F}_q \in \mathbb{R}^{M_s \times C}$, where N_s and M_s are the numbers of points after using grid subsampling [37]. After that, we simply use self-attention and cross-attention to build the correlation between the model point cloud and the scene point cloud, which is written as:

$$\boldsymbol{F}_p = \text{SelfAttn}(\boldsymbol{F}_p, \boldsymbol{F}_p, \boldsymbol{F}_p), \boldsymbol{F}_q = \text{SelfAttn}(\boldsymbol{F}_q, \boldsymbol{F}_q, \boldsymbol{F}_q),$$
$$\boldsymbol{H}_p = \text{CrossAttn}(\boldsymbol{F}_p, \boldsymbol{F}_q, \boldsymbol{F}_q) \tag{1}$$

where $\boldsymbol{H}_p \in \mathbb{R}^{N \times C}$ is the output feature map of the scene point cloud $\boldsymbol{P}$ that embeds the relationship of the model point cloud $\boldsymbol{Q}$. Note that in the experiments, we stack three cross-attention layers, in which the output of the previous layer will be sent to the next layer as the input. For simplicity, we use the same symbol $\boldsymbol{H}_p$ to represent the feature map of the final output. By learning feature correlations, it is desired that the potential instances will be enhanced in the background, making them easier to detect.

Object center prediction. After feature correlation learning, we regress the object center from the whole scene. Given the feature map $\boldsymbol{H}_p \in \mathbb{R}^{N_s \times C}$, we directly predict the offset vector and the instance mask for each point. The point offset vector means the displacement to its instance center, which is given by:

$$V_p = \text{MLP}(\boldsymbol{H}_p) \tag{2}$$

where $\boldsymbol{V}_p \in \mathbb{R}^{N_s \times 3}$ is the offset matrix for N_s points in the scene. To identify whether a point belongs to an instance rather than the background, we predict the point mask, which is written as:

$$\boldsymbol{Y}_p = \text{MLP}(\text{Concat}(\boldsymbol{H}_p, \boldsymbol{G}_p)) \tag{3}$$

where $\boldsymbol{Y}_p \in \mathbb{R}^{N_s \times 1}$ is the mask score. $\boldsymbol{G}_p$ is the geodesic distance embedding from [46]. If the mask score is larger than 0.5, this point is identified as belonging to the point on the object. To obtain an accurate object center, we first displace each point to its potential instance center by adding the original coordinates of each point to its learned point offset ($\boldsymbol{P} + \boldsymbol{V}_p$). Then, we use the learned mask $\boldsymbol{Y}_p$ to filter out background points and leave points on the instance. Subsequently, we employ DBSCAN [14] to group the offset points into K clusters. Finally, by averaging the points in each cluster, we can obtain the center of each instance, which is formulated by:

$$\boldsymbol{S}_p = \text{Avg}(\boldsymbol{Y}_p \cdot \boldsymbol{V}_p) \tag{4}$$

where $\boldsymbol{S}_p \in \mathbb{R}^{K \times 3}$ is the center of K objects.

Object proposal generation. Based on the obtained object centers, we construct the object proposals through ball query operation. Based on the object center, we use radius r to draw a three-dimensional

4

sphere and collect the points that fall within the sphere as the object proposal. Note that the radius parameter r is equal to the radius of the model point cloud. It is desired that the constructed spherical regions should include the entire object as much as possible. Compared with directly learning multi-instance point cloud registration, we can learn pair-wise point cloud registration between each object proposal and the model point cloud, thereby reducing the difficulty of registration.

3.2 3D Dual-Masking Instance Matching

Once we obtain object proposals, we employ a 3D dual-masking instance matching module to learn pair-wise point cloud registration. Specifically, we first learn the instance mask to segment the instance from the object proposal. Then, we learn the overlap mask to segment the common area between the instance and the model point cloud. Finally, based on the instance mask and overlap mask, we learn the pair-wise instance matching.

Instance mask. Since we cannot obtain the ideal object proposal, we need to filter out the background points from the object proposal to obtain the mask of the instance. Given the point cloud of object proposal $\boldsymbol{O} \in \mathbb{R}^{T \times 3}$ (T is the number of points in object proposal), we employ a small encoder structure of KPConv [37] to extract the feature of object proposal. Therefore, we can obtain the feature map $\boldsymbol{E}_o \in \mathbb{R}^{T_s \times C}$, where T_s is the number of points after grid subsampling. The instance mask $\boldsymbol{Y}_o$ is formulated by:

$$\boldsymbol{Y}_o = \mathrm{MLP}(\mathrm{Concat}(\boldsymbol{E}_o, \boldsymbol{G}_o)) \tag{5}$$

where $\boldsymbol{Y}_o \in \mathbb{R}^{T_s \times 1}$ is the mask score and $\boldsymbol{G}_o$ is the geodesic distance embedding from [46]. It is worth noting that in the first stage, we learn point masks for all instances from the entire scene, making it difficult to obtain accurate point masks for each instance. Here, we learn point masks from the object proposal, so we can obtain more accurate instance masks.

Overlap mask. Generally, due to object occlusion, there are a large number of incomplete objects. Therefore, we consider learning the overlap mask between the incomplete object and the complete model point cloud. We feed the model point cloud into the designed small KPConv encoder to obtain feature map $\boldsymbol{E}_q \in \mathbb{R}^{T_q \times C}$. Similarly, we use self-attention and cross-attention to learn the correlation between the object proposal and the model point cloud, which is formulated as:

$$\begin{aligned} \boldsymbol{E}_o &= \mathrm{SelfAttn}(\boldsymbol{E}_o, \boldsymbol{E}_o, \boldsymbol{E}_o), \boldsymbol{E}_q = \mathrm{SelfAttn}(\boldsymbol{E}_q, \boldsymbol{E}_q, \boldsymbol{E}_q), \\ \boldsymbol{Z}_o &= \mathrm{CrossAttn}(\boldsymbol{E}_o, \boldsymbol{E}_q, \boldsymbol{E}_q) \end{aligned} \tag{6}$$

where $\boldsymbol{Z}_o \in \mathbb{R}^{T_s \times C}$ is the enhanced feature map of the object proposal. After that, we use an MLP to predict the overlap mask, which is given by:

$$\boldsymbol{Y}_{op} = \mathrm{MLP}(\mathrm{Concat}(\boldsymbol{Z}_o, \boldsymbol{G}_o)) \tag{7}$$

where $\boldsymbol{Y}_{op} \in \mathbb{R}^{T_s \times 1}$ is the obtained overlap mask. To upsample the overlap mask to the original resolution, we use a small KPConv decoder to generate feature map $\hat{\boldsymbol{Y}}_{op} \in \mathbb{R}^{T \times 1}$.

For subsequent matching steps, we follow [31] to match the dense points within the local patches of two matched sampled points with an optimal transport layer[33]. However, the local correspondences extracted in this manner often cluster closely, which results in unstable pose estimation, as noted in[20]. Since the local area extracted from the focus network will contain some scene noise, and the intercepted instance point cloud will be incomplete, this problem will be more serious. To mitigate this problem, we suggest extracting a dense set of point correspondences within the instance boundaries by utilizing instance masks and overlap masks. For each points correspondences $\hat{\mathcal{C}}_k = (\hat{\mathbf{p}}_i, \hat{\mathbf{q}}_j)$, we collect their neighboring points $\mathcal{N}_i^P$ and $\mathcal{N}_j^Q$. The points out of instance is removed from $\mathcal{N}_i^P$ based on the instance mask. In order to solve the problem of incomplete point clouds, we further use overlap masks to eliminate non-overlapping parts within a patch. Then the clear pair-wise correspondences are extracted with an optimal transport layer and mutual top-k selection, and using the local-to-global registration followed by [31].

3.3 Loss Function

Loss in focusing. For the 3D multi-object focusing module, we need to learn better shape features for localization. Therefore, we follow [31] and use a circle loss L_{circle} to learn fine interactive features,

5

as:

$$L_{\text{circle}}^Q = \frac{1}{|\mathcal{A}|} \sum_{\mathcal{G}_i^Q \in \mathcal{A}} \log[1 + \sum_{\mathcal{G}_j^P \in \varepsilon_p^i} e^{\lambda_i^j \beta_p^{i,j}(d_i^j - \Delta_p)} \cdot \sum_{\mathcal{G}_k^P \in \varepsilon_n^i} e^{\beta_n^{i,k}(\Delta_n - d_i^k)}] \tag{8}$$

where P and Q are source and target points-set and $\mathcal{G}$ is the anchor patches of each set. $d_i^j = \|\hat{\mathbf{h}}_i^Q - \hat{\mathbf{h}}_j^P\|_2$ is the distance in feature space, $\lambda_i^j = (o_i^j)^{\frac{1}{2}}$ and o_i^j is the overlap ratio between $\mathcal{G}_i^P$ and $\mathcal{G}_j^Q$. 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 circle loss on $\mathcal{P}$ is calculated in the same way.

For each sampled points in the scene, we constrain their learned offsets $\mathbf{O} = \{o_1, ..., o_N\} \in \mathbb{R}^{N \times 3}$ from their nearest target instance center using an L1 regression loss as follows:

$$L_{reg} = \frac{1}{\sum_i p_i} \sum_i \|o_i - (\hat{c}_i - p_i)\| \tag{9}$$

where $\hat{c}_i$ is the centroid of the nearest instance that points i belongs to. Considering the varying object sizes across different categories, it is challenging for the network to accurately regress precise offsets, especially for boundary points of large objects, as these points are relatively far from the instance centroids. To tackle this problem, we introduce a direction loss to constrain the direction of the predicted offset vectors. We define this loss, following the method in[21], as a measure of the negative cosine similarities, $i.e.$,

$$L_{dir} = -\frac{1}{\sum_i p_i} \sum_i \frac{o_i}{\|o_i\|_2} \cdot \frac{\hat{c}_i - p_i}{\|\hat{c}_i - p_i\|_2} \tag{10}$$

The overall focusing loss is computed as: $L_{focusing} = L_{circle} + L_{reg} + L_{dir}$.

Loss in matching. For coarse points feature learning, we employ the circle loss, as mentioned earlier in the Focus Network. As for point matching, we following [31] use a negative log-likelihood loss on the assignment matrix $\bar{\mathbf{Z}}_i$ of each ground-truth points correspondence $\hat{c}_i^*$, just as following:

$$L_{\text{nll},i} = -\sum_{(x,y) \in \mathcal{C}_i^*} \log \bar{z}_{x,y}^i - \sum_{x \in \mathcal{I}_i} \log \bar{z}_{x,m_i+1}^i - \sum_{y \in \mathcal{J}_i} \log \bar{z}_{n_i+1,y}^i \tag{11}$$

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 points matches: $L_{\text{nll}} = \frac{1}{N_g} \sum_{i=1}^{N_g} L_{p,i}$.

Regarding the prediction of instance masks and overlap masks, we follow the methodology outlined in [28]. The mask prediction loss is composed of binary cross-entropy (BCE) loss and dice loss with Laplace smoothing, defined as follows:

$$L_{\text{mask},i} = \text{BCE}(m_i, m_i^{gt}) + 1 - 2\frac{m_i \cdot m_i^{gt} + 1}{|m_i| + |m_i^{gt}| + 1} \tag{12}$$

where m_i and m_i^{gt} are the predicted and the ground-truth instance masks, respectively. The final mask prediction loss is the average loss over all. The total matching loss function $L_{matching}$ is defined as:

$$L_{matching} = L_{circle} + L_{nll} + L_{overlapmask} + L_{instancemask}. \tag{13}$$

4 Experiments

4.1 Datasets and Evaluation Metrics

For multi-instance point cloud registration, We train and evaluate our method on two public benchmarks: Scan2CAD [3] and ROBI [43].

Scan2CAD. As a pioneering dataset in the realm of aligning scenes with CAD models, it leverages the resources of ScanNet [12] and ShapeNet [9] to form a multi-instance registration dataset. With a corpus comprising 1,506 scenes sourced from ScanNet, meticulously annotated with 14,225 CAD models from ShapeNet alongside their spatial orientations within the scenes, Scan2CAD sets a new standard in scene-CAD alignment. For the total of 2,184 pairs of point clouds, it allocates 70% of pairs for training, 10% for validation, and reserves 20% for testing.

6

Table 1: Results comparison of different methods on the test sets of both the Scan2CAD and ROBI datasets. The best results are highlighted in **bold**. Please note that "3DFMNet*" indicates the upper bound of our method.

Methods	Scan2CAD			ROBI		
	MR (%)	MP (%)	MF (%)	MR (%)	MP (%)	MF (%)
T-Linkage [26]	77.12	46.04	57.65	12.04	10.47	11.20
RansaCov [27]	84.78	71.34	77.48	14.14	26.29	18.38
PointCLM [48]	91.85	91.08	91.46	18.68	40.11	25.48
ECC [36]	**96.52**	89.03	92.62	24.65	34.85	28.91
MIRETR [46]	95.70	91.21	93.40	38.51	41.19	39.80
3DFMNet (ours)	95.44	**94.15**	**94.79**	**46.81**	**50.61**	**48.63**
3DFMNet* (ours)	97.68	94.63	96.14	52.59	63.13	57.38

ROBI. It is a dataset tailored specifically for industrial bin-picking applications. ROBI collects 7 reflective metallic industrial objects and 63 meticulously crafted bin-picking scenes. Each point cloud pair is meticulously crafted, with the scene point cloud generated through back projection from depth images, while the model point cloud is meticulously sampled from the CAD model corresponding to its industrial counterpart. There are a total of 4,880 pairs of ROBI, divided into 70% for training, 10% for validation, and 20% for testing.

Metrics. We adopt three registration metrics to evaluate the methods, including Mean Recall (MR), Mean Precision (MP), and Mean F1 score (MF). We refer to the settings used in MIRETR and previous work to determine whether an instance is recognized as correctly registered based on RTE and RRE [48, 36]. Specifically, we consider a match successful when $RTE \leq 4 \times voxelsize$ and $RRE \leq 15°$. Following existing methods, such as MIRETR[46], the voxel sizes of Scan2CAD and ROBI dataset are set to 0.025m and 0.0015m, respectively. MR quantifies the proportion of registered instances relative to the total number of ground-truth instances, while MP measures the ratio of registered instances against the entirety of predicted instances. MF score is the harmonic mean of both MP and MR. Additionally, we present the pair-wise inlier ratio (PIR), elucidating the proportion of inliers from a single instance amidst all extracted correspondences, considering one of the most important challenges of multi-instance registration is identifying the set of inliers for individual instances.

4.2 Implementation Details

Our method is trained on NVIDIA RTX 4090 GPUs and uses the Pytorch deep learning platform. We employ the Adam optimizer for 60 epochs. In initial learning rate and weight decay are set to 0.001 and 0.0001, respectively. We use a KPConv-FPN[37] backbone followed by [46] for feature extraction. We utilize a voxel subsampling approach to reduce the resolution of the point clouds, resulting in the creation of sampled points and dense points, which are then inputted into the network. The initial step involves downsampling the input point clouds using a voxel-grid filter with a size of 2.5cm for Scan2CAD and 0.15cm for ROBI. Subsequently, we employ a 4-stage backbone architecture in both the multi-object focusing and sub-matching network. Following each stage, the voxel size is increased twofold to further reduce the resolution of the point clouds. The initial and final (coarsest) levels of downsampled points correspond to the dense points and sampled points, respectively, which are used for follow-up process. In the multi-object focusing network, we employ a ball query approach akin to the one detailed in [29], enabling us to retrieve the local neighborhood surrounding the regression center. The search radius is set to 1.2 times the size of the CAD model. Specifically, we randomly sample 4096 points from the dense points obtained through voxel subsampling in both Scan2CAD and ROBI datasets. During training, we use the ground truth center as supervision to train the 3D multi-object focusing module and use the point cloud around the ground truth center as the training data to train the matching network. During testing, we use the center predicted by the 3D multi-object focusing module and its surrounding point cloud as the input to the 3D dual-masking instance matching module to regress the final pose.

7

4.3 Results

Quantitative results. For a comprehensive comparison, we compare our 3DFMNet with five recent advanced multi-instance point cloud registration approaches, including T-Linkage [26], RansaConv [27], PointCLM [48], ECC [36], and MIRETR [46]. As shown in Table 1, we report the results of different methods on Scan2CAD [3] and ROBI [43], respectively. Note that except for the results of our method, the results of all other methods are taken from the official paper of MIRETR, as their results are obtained using the same backbone network. From the table, it can be observed that the performance of our method is superior to all other methods on the ROBI datasets. The density of objects in ROBI is higher than that in Scan2CAD, making ROBI more challenging. It can be observed that our method improves performance by about 7% in terms of MR, MP, and MF on the ROBI dataset. Since we use the focus-and-matching strategy to learn multiple pair-wise registrations, our method can learn more accurate matches from object proposals compared to the whole scene.

Since our method first localizes the object and then executes pair-wise correspondence, the precision of object localization has a significant impact on subsequent matching. In Table 1, we provide the upper bound on the performance of our method, denoted by "3DFMNet*". Specifically, we utilize the ground truth object center to generate the object region, and use it to evaluate the performance upper bound of our method. For the Scan2CAD dataset, the upper bounds of MR and MP are about 97% and 94%, respectively. However, due to the cluttered and incomplete objects in the ROBI dataset, the upper bounds for MR and MP are only about 52% and 63%. Nonetheless, the performance upper bound of our method is still higher than previous methods. In a word, the theoretical upper bound indicates that the strategy of focusing first and then matching can ensure that our method can achieve excellent performance.

To evaluate the 3D multi-object focusing module, we computed the mean recall (MR), mean precision (MP), and root mean square error (RMSE) for detected object centers. We defined successful detection as cases where the predicted center lies within $distance \leq 0.1 \times r_{instance}$ (where $r_{instance}$ is the radius of instance) of the ground truth center. As shown in Table 2, the module achieves strong results across MR, MP, and RMSE, confirming its effectiveness in identifying object centers in the first stage. n the second stage, the 3D dual-masking instance matching module refines detections by applying instance and overlap mask learning to reduce false positives. Analyzing the Scan2CAD dataset, we found that 1.05% of objects were initially misdetected (25 instances), but 22 were successfully filtered out by the masking module. These misdetections have minimal impact, as few matching points are generated for falsely detected objects, preventing SVD from calculating a relative pose and limiting their influence on final results.

For unseen scene experiments, we follow MIRETR by using the ShapeNet dataset [9] (55 categories in total) to evaluate generalization to novel categories. Specifically, we train on CAD models from the first 30 categories and test on the remaining 25. To address class imbalance, we sample up to 500 models per category, as in MIRETR. Each point cloud pair includes a randomly selected CAD model and a scene model with 4–16 random poses applied, resulting in 8,634 pairs for training, 900 for validation, and 7,683 for testing. Table 3 shows that our method demonstrates strong generalizability on unseen scenes compared to MIRETR.

Table 2: Result related to the 3D multi-object focusing module.

Datasets	MR (%)	MP (%)	RMSE (m)
Scan2CAD [3]	98.14	98.85	0.0814
ROBI [43]	80.30	99.99	0.0065

Table 3: The generalizability of the 3D multi-object focusing module to unseen scenes.

Methods	MR (%)	MP (%)	MF (%)
MIRETR [46]	94.95	93.94	94.44
3DFMNet (ours)	**95.12**	**94.23**	**94.67**

Visualization. In Figure 3, we show the visualization of chairs of our method with the previous state-of-the-art MIRETR [46] on the test set of the Scan2CAD dataset. From the left to right, the figure shows the results of ground truth, MIRETR, and 3DFMNet (ours), respectively. Note that we transform the chair's CAD model to the corresponding scenes. By comparing the ground truth with our method, it can be observed that our method can successfully match the multiple chairs in the cluttered scene. In addition, we also show the pair-wise correspondences on the test set of the Scan2CAD dataset in Figure 5. In Figure 4, we also show the visualization of the incomplete objects

8

on the test set of the ROBI dataset. It can be observed that our method can effectively obtain the correspondences between the challenging incomplete objects and the model point cloud.

Time costs. Here we report the inference time of different methods on the ROBI dataset for a comprehensive comparison. Table 4 shows model inference time for feature extraction and pose estimation time for transformation. "Total" represents the sum of "Model" time and "Pose" time. While our two-stage method has a slightly higher total time than the one-stage MIRETR [46], it runs faster than the two-stage PointCLM [48]. Nonetheless, compared with Point-CLM and MIRETR, our 3DFMNet achieves higher performance.

Table 4: Per scene time on the ROBI dataset.

Methods	Model	Pose	Total
T-Linkage [26]	0.30	3.07	3.34
RansaCov [27]	0.30	0.17	0.47
PointCLM [48]	0.30	0.33	0.63
ECC [36]	0.30	0.21	0.51
MIRETR [46]	0.30	**0.10**	**0.40**
3DFMNet (ours)	**0.26**	0.28	0.54

Inst: 20   # Inst: 11   # Inst: 20

Inst: 26   # Inst: 12   # Inst: 25

(a) Ground Truth   (b) MIRETR   (c) 3DFMNet (ours)
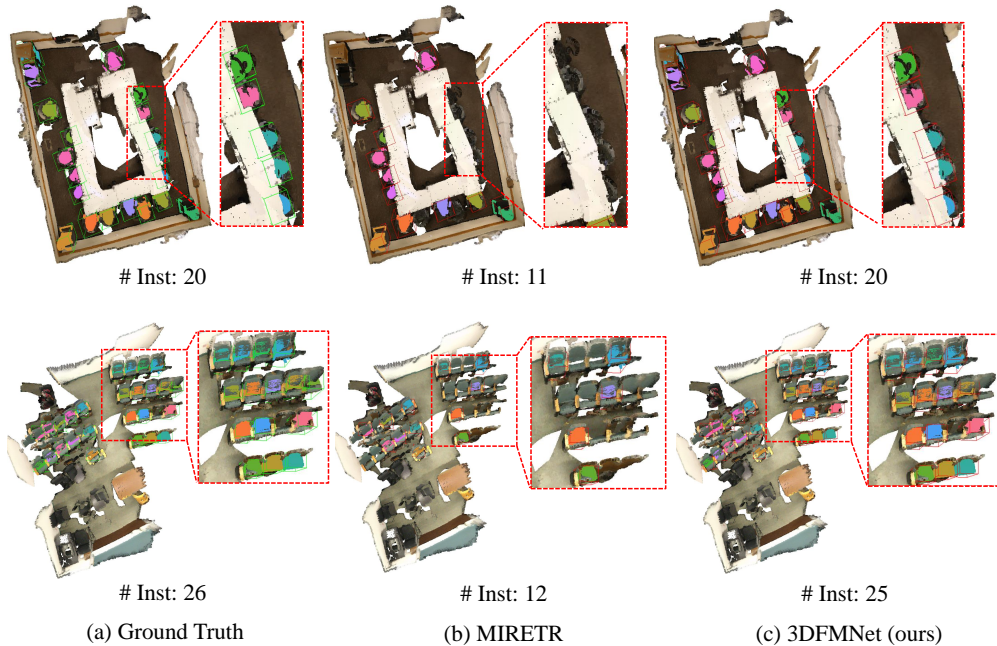
Figure 3: Registration results on the test set of the Sacn2CAD dataset. We visualize the successfully registered instances of MIRETR [46] in (b) and ours in (c). "# Inst" means the number of registered instances. Note that for a better view, we draw the green boxes for the ground truth and the red boxes for the predict correspondences.

4.4 Ablation Study

Impact of the number of sampled points. The grid sizes influence the number of sampled points. We conduct experiments on the Scan2CAD dataset to verify the impact of different numbers of points. The results of MR and MP are 83.76% and 92.50% (1024 points), **95.44**% and **94.15**% (default 4096 points), and 95.20% and 93.75% (8192 points), respectively. It is evident that too few sampling points hinder information gathering and reduce accuracy. Conversely, too many sampling points do not improve accuracy and instead decrease it due to redundancy.

Effective of dual-masking. We evaluate the necessity of our dual-masking structure under the fair setting of the Scan2CAD dataset. When taking turns removing components, the results of MR and MP are 94.76% and 93.30% (only removing the overlap mask), 91.82% and 93.53% (only removing the instance mask), 90.01% and 90.90% (removing both), and **95.44**% and **94.15**% (using both). The ablation study results can demonstrate the effectiveness of our dual-masking structure.

9

# Inst:14	# Inst:2	# Inst:6
# Inst:26	# Inst:10	# Inst:19
(a) Input	(b) Ground Truth	(c) MIRETR	(d) 3DFMNet (ours)
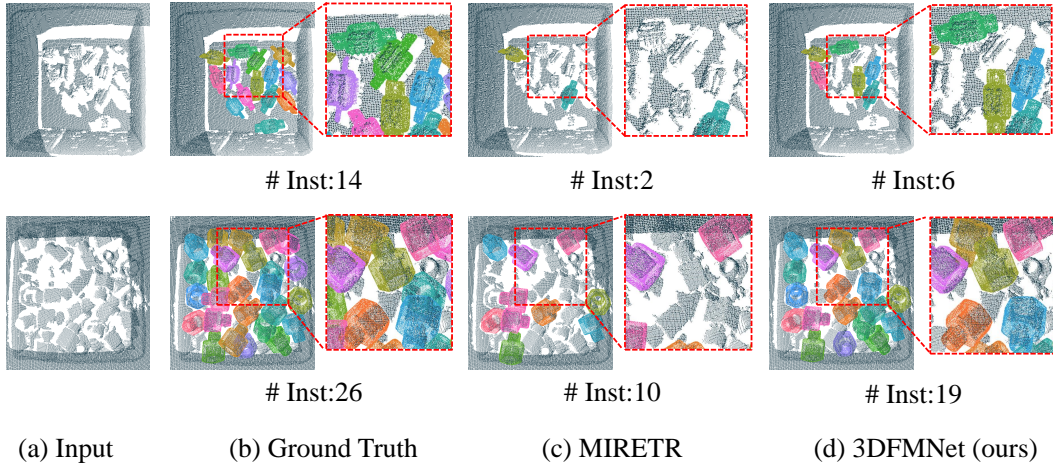
Figure 4: Registration results on the test set of the ROBI dataset. We visualize the successfully registered instances of MIRETR [46] in (c) and ours in (d).

(a) Ground Truth	(b) Input Point Cloud	(c) Instance Level Matching
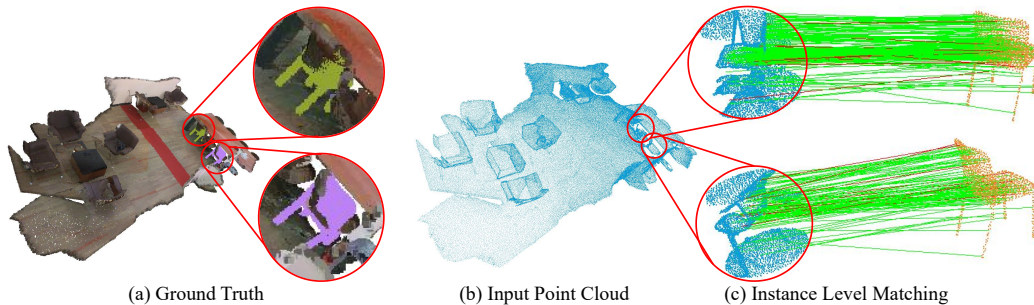
Figure 5: Visualization results of pair-wise correspondences on the test set of Scan2CAD dataset.

4.5 Limitation

Our 3D focusing-and-matching network is a two-stage framework for the multi-instance point cloud registration task. The localization accuracy of the first stage will affect the pair-wise correspondence in the second stage. We have analyzed the upper bound of our method in 4.3. In addition, due to the two-stage process, the inference time of our method is slightly lower than the previous MIRETR [46]. In future work, we will strive to improve the performance and speed of our method.

5 Conclusion

In this paper, we proposed a 3D focusing-and-matching network (3DFMNet) for multi-instance point cloud registration. Specifically, we first presented a 3D multi-object focusing module that learns to localize the center of the potential target in the scene by considering the correlation between the model point cloud and the scene point cloud. Then, we designed a 3D dual-masking instance matching module to learn the pair-wise correspondence between the model point cloud and the localized object. Extensive experiments on two public benchmarks, Scan2CAD and ROBI, show that our method achieves a new state-of-the-art performance on the multi-instance point cloud registration task.

Acknowledgments

The authors would like to thank reviewers for their detailed comments and instructive suggestions. This work was supported by the National Science Fund of China (Grant Nos. 62306238, 62271410, 62001394) and the Fundamental Research Funds for the Central Universities.

10

References

[1] Matthias Niessner Matthew Fisher Jianxiong Xiao Andy Zeng, Shuran Song and Thomas Funkhouser. 3DMatch: Learning local geometric descriptors from RGB-D reconstructions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[2] Sheng Ao, Qingyong Hu, Hanyun Wang, Kai Xu, and Yulan Guo. Buffer: Balancing accuracy, efficiency, and generalizability in point cloud registration. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023.

[3] Armen Avetisyan, Manuel Dahnert, Angela Dai, Manolis Savva, Angel X Chang, and Matthias Nießner. Scan2CAD: Learning CAD model alignment in RGB-D scans. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[4] Daniel Barath and Jiri Matas. Graph-cut RANSAC. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[5] Daniel Barath and Jiri Matas. Progressive-X: Efficient, anytime, multi-model fitting algorithm. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2019.

[6] Nassir Navab Bertram Drost, Markus Ulrich and Slobodan Ilic. Model gobally, match locally: Efficient and robust 3D object recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2010.

[7] Eric Brachmann and Carsten Rother. Neural-guided RANSAC: Learning where to sample model hypotheses. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2019.

[8] Xinyue Cao, Xiyu Zhang, Yuxin Cheng, Zhaoshuai Qi, Yanning Zhang, and Jiaqi Yang. Instance by instance: An iterative framework for multi-instance 3D registration. *arXiv preprint arXiv:2402.04195*, 2024.

[9] Angel X Chang, Thomas Funkhouser, Leonidas Guibas, Pat Hanrahan, Qixing Huang, Zimo Li, Silvio Savarese, Manolis Savva, Shuran Song, Hao Su, et al. ShapeNet: An information-rich 3D model repository. *arXiv preprint arXiv:1512.03012*, 2015.

[10] Jaesik Park Christopher Choy and Vladlen Koltun. Fully convolutional gometric features. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2019.

[11] Wei Dong Christopher Choy and Vladlen Koltun. Deep global registration. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[12] Angela Dai, Angel X. Chang, Manolis Savva, Maciej Halber, Thomas Funkhouser, and Matthias Nießner. ScanNet: Richly-annotated 3D reconstructions of indoor scenes. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[13] Ivan Eichhardt Levente Hajder Daniel Barath, Denys Rozumny and Jiri Matas. Progressive-X+: Clustering in the consensus space. In *arXiv preprint arXiv:2103.13875*, 2021.

[14] Martin Ester, Hans-Peter Kriegel, Jörg Sander, Xiaowei Xu, et al. A density-based algorithm for discovering clusters in large spatial databases with noise. In *kdd*, 1996.

[15] Martin A Fischler and Robert C Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 1981.

[16] Venu Madhav Govindu Jacinto C Nascimento Rama Chellappa G Dias Pais, Srikumar Ramalingam and Pedro Miraldo. 3DRegNet: A deep neural network for 3D point registration. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[17] Guangfu Wang Guanghui Liu Hao Xu, Shuaicheng Liu and Bing Zeng. Omnet: Learning overlapping mask for partial-to-partial point cloud registration. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2021.

[18] Mahdi Saleh Benjamin Busam Hao Yu, Fu Li and Slobodan Ilic. CofiNet: Reliable coarse-to-fine correspondences for robust point cloud registration. In *Proceedings of the Advances in Neural Information Processing Systems (NeurIPS)*, 2021.

[19] Tolga Birdal Haowen Deng and Slobodan Ilic. PPFNet: Global context aware local features for robust 3D point matching. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

11

[20] Shengyu Huang, Gojcic Zan, Usvyatsov Mikhail, Wieser Andreas, and Schindler Konrad. Predator: Registration of 3D point clouds with low overlap. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021.

[21] Li Jiang, Hengshuang Zhao, Shaoshuai Shi, Shu Liu, Chi-Wing Fu, and Jiaya Jia. Pointgroup: Dual-set point grouping for 3D instance segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[22] Yasushi Kanazawa and Hiroshi Kawakami. Detection of planar regions with uncalibrated stereo using distributions of feature points. In *Proceedings of British Machine Vision Conference(BMVC)*, 2004.

[23] Xiaoyuan Luo Kexue Fu, Shaolei Liu and Manning Wang. Robust point cloud registration framework based on deep graph matching. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021.

[24] Florian Kluger, Eric Brachmann, Hanno Ackermann, Carsten Rother, Michael Ying Yang, and Bodo Rosenhahn. Consac: Robust multi-model fitting by conditional sample consensus. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[25] et al Luca Magri, Fusiello Andrea. Robust multiple model fitting with preference analysis and low-rank approximation. In *Proceedings of British Machine Vision Conference(BMVC)*, 2015.

[26] Luca Magri and Andrea Fusiello. T-Linkage: A continuous relaxation of J-Linkage for multi-model fitting. In *CVPR*, 2014.

[27] Luca Magri and Andrea Fusiello. Multiple model fitting as a set coverage problem. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[28] Fausto Milletari, Nassir Navab, and Seyed-Ahmad Ahmadi. V-net: Fully convolutional neural networks for volumetric medical image segmentation. In *Proceedings of 2016 fourth international conference on 3D vision (3DV)*, 2016.

[29] Charles Ruizhongtai Qi, Li Yi, Hao Su, and Leonidas J Guibas. PointNet++: Deep hierarchical feature learning on point sets in a metric space. 2017.

[30] Jaesik Park Qian-Yi Zhou and Vladlen Koltun. Fast global registration. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2016.

[31] Zheng Qin, Hao Yu, Changjian Wang, Yulan Guo, Yuxing Peng, and Kai Xu. Geometric Transformer for fast and robust point cloud registration. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.

[32] Nico Blodow Radu Bogdan Rusu and Michael Beetz. Fast point feature histograms for 3D registration. 2009.

[33] Paul-Edouard Sarlin, Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. Superglue: Learning feature matching with graph neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[34] Yaqi Shen, Le Hui, Haobo Jiang, Jin Xie, and Jian Yang. Reliable inlier evaluation for unsupervised point cloud registration. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, 2022.

[35] Qian-Yi Zhou Sungjoon Choi and Vladlen Koltun. Robust reconstruction of indoor scenes. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015.

[36] Weixuan Tang and Danping Zou. Multi-instance point cloud registration by efficient correspondence clustering. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.

[37] Hugues Thomas, Charles R Qi, Jean-Emmanuel Deschaud, Beatriz Marcotegui, François Goulette, and Leonidas J Guibas. KPConv: Flexible and deformable convolution for point clouds. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2019.

[38] Yue Wang and Justin Solomon. PRNet: Self-supervised learning for partial-to-partial registration. In *Proceedings of the Advances in Neural Information Processing Systems (NeurIPS)*, 2019.

[39] Yue Wang and Justin M Solomon. Deep closest point: Learning representations for point cloud registration. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2019.

12

[40] Guofeng Mei Xiaoshui Huang and Jian Zhang. Feature-metric registration: A fast semi-supervised approach for robust point cloud registration without correspondences. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[41] Lei Zhou Hongbo Fu Long Quan Xuyang Bai, Zixin Luo and Chiew-Lan Tai. D3Feat: Joint learning of dense detection and description of 3D local features. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[42] Lei Zhou Hongkai Chen Lei Li Zeyu Hu Hongbo Fu Xuyang Bai, Zixin Luo and Chiew-Lan Tai. Robust point cloud registration using deep spatial consistency. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021.

[43] Jun Yang, Yizhou Gao, Dong Li, and Steven L Waslander. ROBI: A multi-view dataset for reflective objects in robotic bin-picking. In *Proceedings of the International Conference on Intelligent Robots and Systems (IROS)*, 2021.

[44] Rangaprasad Arun Srivatsan Yasuhiro Aoki, Hunter Goforth and Simon Lucey. PointnetLK: Robust and efficient point cloud registration using pointnet. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[45] Zi Jian Yew and Gim Hee Lee. RPM-Net: Robust point matching using learned features. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[46] Zhiyuan Yu, Zheng Qin, Lintao Zheng, and Kai Xu. Learning instance-Aware correspondences for robust multi-Instance point cloud registration in cluttered scenes. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024.

[47] Zhiyuan Yu, Qin Zheng, Chenyang Zhu, and Kai Xu. Efficient and accurate Multi-Instance point cloud registration with iterative main cluster detection. 2024.

[48] Mingzhi Yuan, Zhihao Li, Qiuye Jin, Xinrong Chen, and Manning Wang. PointCLM: A contrastive learning-based framework for multi-instance point cloud registration. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2022.

[49] Jan D. Wegner Zan Gojcic, Caifa Zhou and Andreas Wieser. The perfect match: 3D point cloud matching with smoothed densities. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

13

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: Please refer to the abstract and introduction.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: Please refer to the main paper.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory Assumptions and Proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [NA]

   Justification: Our Paper does not include theoretical results.

   Guidelines:

   - The answer NA means that the paper does not include theoretical results.
   - All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
   - All assumptions should be clearly stated or referenced in the statement of any theorems.

14

- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental Result Reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: Please refer to the experimental part.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: We will release the codes.

   Guidelines:

   - The answer NA means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.

15

- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental Setting/Details**

Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

Answer: [Yes]

Justification: Please refer to the experimental part.

Guidelines:
- The answer NA means that the paper does not include experiments.
- The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
- The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment Statistical Significance**

Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

Answer: [No]

Justification: We didn't report error bars.

Guidelines:
- The answer NA means that the paper does not include experiments.
- The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments Compute Resources**

Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

Answer: [Yes]

Justification: Please refer to the experimental part.

Guidelines:
- The answer NA means that the paper does not include experiments.
- The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
- The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.

16

- The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code Of Ethics**

Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

Answer: [Yes]

Justification: We have ensured anonymity.

Guidelines:

- The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
- If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
- The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader Impacts**

Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

Answer: [No]

Justification: Our work is only for academic research purpose.

Guidelines:

- The answer NA means that there is no societal impact of the work performed.
- If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

Answer: [No]

Justification: Our is only for academic research purpose.

Guidelines:

- The answer NA means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

17

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: Please refer to the reference part.

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
    - If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New Assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [No]

    Justification: We don't include any new assets.

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and Research with Human Subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

    Justification: We didn't use crowdsourcing or conduct research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional Review Board (IRB) Approvals or Equivalent for Research with Human Subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

    Justification: We didn't use crowdsourcing or conduct research with human subjects.

    Guidelines:

18

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

19