FINet: Dual Branches Feature Interaction for Partial-to-Partial Point Cloud Registration

Hao Xu[1,2], **Nianjin Ye**[2], **Guanghui Liu**[1*], **Bing Zeng**[1], **Shuaicheng Liu**[1,2*]
[1]University of Electronic Science and Technology of China
[2]Megvii Technology
{xuhao02@std., guanghuiliu@, eezeng@, liushuaicheng@}uestc.edu.cn
{xuhao02, yenianjin, liushuaicheng}@megvii.com

Abstract

Data association is important in the point cloud registration. In this work, we propose to solve the partial-to-partial registration from a new perspective, by introducing multi-level feature interactions between the source and the reference clouds at the feature extraction stage, such that the registration can be realized without the attentions or explicit mask estimation for the overlapping detection as adopted previously. Specifically, we present FINet, a feature interaction-based structure with the capability to enable and strengthen the information associating between the inputs at multiple stages. To achieve this, we first split the features into two components, one for rotation and one for translation, based on the fact that they belong to different solution spaces, yielding a dual branches structure. Second, we insert several interaction modules at the feature extractor for the data association. Third, we propose a transformation sensitivity loss to obtain rotation-attentive and translation-attentive features. Experiments demonstrate that our method performs higher precision and robustness compared to the state-of-the-art traditional and learning-based methods. Code is available at https://github.com/megvii-research/FINet.

Introduction

Point cloud registration is a longstanding research problem in the areas of computer vision and computer graphics, including augmented reality (Azuma 1997; Billinghurst, Clark, and Lee 2014), object pose estimation (Fang et al. 2018; Wang et al. 2019a) and 3D reconstruction (Lin, Kong, and Lucey 2018). It aims to predict a rigid 3D transformation, aligning the source point cloud to the reference. Data association is critical for aligning two point clouds, especially for the practical partial-to-partial scenes where inputs are obscured by partiality and contaminated by noise. Algorithms for this task have been improved steadily, which can be divided into two categories: correspondence matching-based and global feature-based methods.

Iterative Closest Point (ICP) (Besl and McKay 1992) is the most classical algorithm among the correspondence matching-based methods, where the correspondences are obtained by the nearest neighbor search. Subsequently, several methods (Segal, Haehnel, and Thrun 2009; Pomerleau, Colas, and Siegwart 2015; Yang, Li, and Jia 2013) are proposed to either improve the matching quality or search a larger transformation space. Recently, some learning-based approaches implement descriptor with neural network to improve the performance. Specifically, (Wang and Solomon 2019b) adopts Graph Neural Network (GNN) (Wang et al. 2019b) and attention mechanism (Vaswani et al. 2017), which enables the information exchange between the inputs. However, most of them rely on the deep features that extracted from the local geometric structures. As a result, they can hardly utilize the knowledge of the entire point clouds. Without a global picture, the data association is inefficient.

In contrast, global feature-based methods can overcome the above-mentioned issues by aggregating global information without correspondences, e.g., PointNetLK (Aoki et al. 2019) and Feature-metric Registration (FMR) (Huang, Mei, and Zhang 2020). Although they can learn features from the entire point cloud, they perform poorly in partial-to-partial registration due to the lack of data association. Recently, OMNet (Xu et al. 2021) predicts overlapping masks, con-

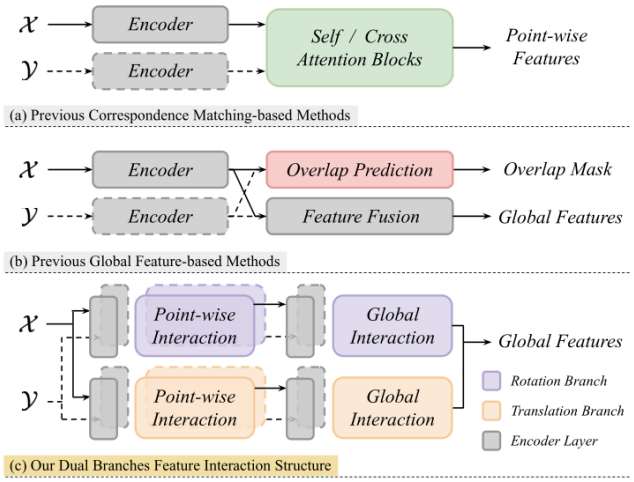

Figure 1: The structure comparison of partiality handling. The specific designs are shown in different colors. Our method use the multi-level feature interaction to deal with the partiality without using any attention mechanism in (a) or explicit overlapping region prediction in (b).

*Corresponding author

verting the partial-to-partial registration to the registration of the same shape. However, the mask is predicted based on the features that extracted without early information exchange, making it harder to be estimated accurately.

In this paper, we solve the partial-to-partial registration from a new perspective, by introducing multi-level feature interactions between the input point clouds. We show that abundant information exchange between the inputs at the feature extraction stage can naturally equip the network with the partiality perceptual capability. As such, the global features from two inputs can focus on the same parts of an object. Interestingly, as shown in Fig. 1, this can be achieved implicitly without the need of attention modules or explicit mask estimations, as long as feature interactions are enabled.

To this end, we propose FINet: a feature interaction-based structure with the ability to enable and strengthen the data association between the inputs at multiple stages. To promote the information associating, several interaction modules are inserted to the feature extractor. Besides, the 3D rigid transformation consists of 3D translation and 3D rotation, which reside in different solution spaces. Previously, they are regressed from the same deep feature. In this work, we implement a dual branches structure to process them separately so as to enhance the feature interactions. Moreover, based on the dual branches structure, we design a transformation sensitivity loss, which encourages the network to extract the rotation-attentive and translation-attentive features, improving the quality of regression from their own solution space. Furthermore, to avoid concentrating on local geometry, we propose a point-wise feature dropout loss to encourage aggregating features from scattered locations. We summarize our key contributions as follows:

- We propose a multi-level feature interaction module for the point cloud registration, which promotes the data association between the inputs and shows state-of-the-art performance on several partially visible datasets.
- We propose a dual branches structure to mitigate the effect of solution space disparity between rotation and translation, so as to further enhance feature interactions.
- We design a transformation sensitivity loss, which supervises two branches of the feature extractor to learn rotation-attentive and translation-attentive features. Besides, we propose a point-wise feature dropout loss to facilitate the learning of global information.

Related Work

Correspondence Matching-based Methods. ICP (Besl and McKay 1992) and its variants (Rusinkiewicz and Levoy 2001; Segal, Haehnel, and Thrun 2009) are the earlier correspondence matching-based methods, which calculate the nearest neighbors as correspondences. However, they are often strapped into local minima due to the non-convexity. To this end, Go-ICP (Yang, Li, and Jia 2013) utilizes a branch-and-bound strategy to find a good optimum at the expense of speed. Symmetric ICP (Rusinkiewicz 2019) improves the point-to-plane objective function. Furthermore, Fast Global Registration (FGR) (Zhou, Park, and Koltun 2016) uses FPFH (Rusu, Blodow, and Beetz 2009) features and an alternating optimization technique to improve efficiency. Recent learning-based methods use Multi-Layer Perceptron (MLP) (Qi et al. 2017a,b) or GNN (Wang et al. 2019b) to replace the handcrafted descriptors. Specifically, DCP (Wang and Solomon 2019a) calculates feature-to-feature correspondences. DeepGMR (Yuan et al. 2020) integrates Gaussian Mixture Model (GMM) to learn point-to-GMM correspondences. Generally, the accurate matching heavily relies on the distinctive geometric structures, and an extra time-consuming RANSAC (Fischler and Bolles 1981) may be needed. In contrast, we aggregate feature from the entire point clouds and achieve an end-to-end registration.

Global Feature-based Methods. PointNetLK (Aoki et al. 2019) pioneers the global feature-based methods, which utilizes the Lucas & Kanade (LK) algorithm (Lucas and Kanade 1981) to solve the rigid transformation. PCR-Net (Sarode et al. 2019) improves the robustness to noise by replacing the LK algorithm with a MLP. Subsequently, FMR (Huang, Mei, and Zhang 2020) constrains the global feature distance of the inputs with an extra decoder. However, all of them ignore the effect of partiality. Our method is aware of the partiality and robust to different scenes.

Partial-to-partial Registration. As a more realistic problem, partial-to-partial registration is studied by several works. Particularly, PRNet (Wang and Solomon 2019b) uses the Gumble-Softmax (Jang, Gu, and Poole 2016) to improve the feature matching. RPMNet (Yew and Lee 2020) applies the Sinkhorn normalization (Sinkhorn 1964) to encourage the bijectivity of the matching. RGM (Fu et al. 2021) further utilizes deep graph matching to reject outliers. Meanwhile, some methods are designed for the more challenging indoor scenes, e.g., DGR (Choy, Dong, and Koltun 2020), FCGF (Choy, Park, and Koltun 2019) and D3Feat (Bai et al. 2020). Recently proposed PREDATOR (Huang et al. 2020a) predicts the overlapping and matchability scores to tackle the partiality. Similarly, OMNet (Xu et al. 2021) estimates the overlapping masks. Nevertheless, most of these methods lack early information exchange between the inputs, resulting in features that are imperceptible of partiality.

Feature Interaction. Some previous works implement the data association during the feature extraction. Particularly, DCP and PRNet introduce the attention module (Vaswani et al. 2017) to enable information exchanging between the inputs. Nevertheless, the 3D local features lack the global information, further reducing the effectiveness of the feature interaction. Although PREDATOR applies cross-attention mechanism to capture the global picture, it has significant computational and memory requirements. Besides, the data associations in OMNet is located after the feature extractor, which is too late to catch enough partiality knowledge. However, our method possesses feature interactions at multiple levels, which enables the partiality perception without any attention mechanism or overlapping prediction.

Method

Fig. 2 shows an illustration of our method. We estimate the 3D rigid transformation iteratively, which is represented in the form of quaternion $\mathbf{q}$ and translation $\mathbf{t}$. The entire

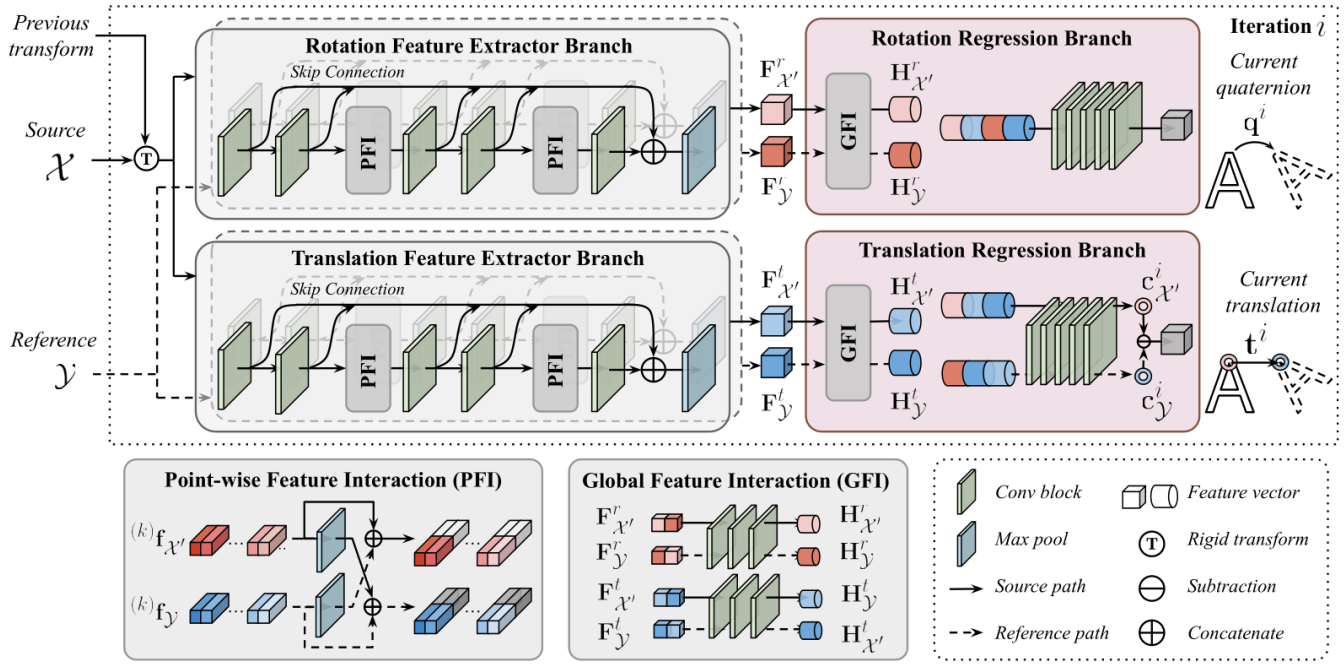

Figure 2: Network architecture for FINet and the multi-levels feature interaction modules.

structure can be divided into two parts: a dual branches encoder and a dual branches transformation estimator, where the multi-level features of the source and the reference point clouds are interactive with each other. Finally, the loss functions are detailed explained.

Notation

Here we introduce some notation that will be used throughout the paper. The registration problem is that for a given source point cloud $\mathcal{X} \in \mathbb{R}^{3\times N_\mathcal{X}}$ and a reference point cloud $\mathcal{Y} \in \mathbb{R}^{3\times N_\mathcal{Y}}$ of $N_\mathcal{X}$ and $N_\mathcal{Y}$ points, we aim to find the 3D rigid transformation $\{\mathbf{R}, \mathbf{t}\}$ that aligns $\mathcal{X}$ to $\mathcal{Y}$.

Dual Branches Encoder

Since the translation belongs to Euclidean space, which is little correlated to the quaternion space. It is not appropriate to obtain the rotation-attentive and translation-attentive features with shared weights. Meanwhile, the features that only contain the rotation or translation information are more beneficial to the data association. To this end, we use a dual branches encoder to extract features for them separately. Refer to (Huang et al. 2020b), point-wise features that output from each convolution block in the MLP are extracted to combine the multi-level features by max-pooling. To promote the low-level information propagation between the source and the reference point clouds, two point-wise feature interaction modules are injected into the encoder. At each iteration, the source point cloud $\mathcal{X}$ is first transformed by the previous transformation into the transformed point cloud $\mathcal{X}'$. The encoder takes $\mathcal{X}'$ and the reference point cloud $\mathcal{Y}$ as inputs, generating the global features as follow:

$$\mathbf{F}^n_m = \max(\mathrm{cat}[\{{}^{(k)}\mathbf{f}^n_m | k = 1..K\}]). \tag{1}$$

Here, $m \in \{\mathcal{X}', \mathcal{Y}\}$, $n \in \{r, t\}$. The superscript r and t denote the global features $\mathbf{F}$ that belong to the rotation and translation encoders respectively. The superscript k represents the point-wise features $\mathbf{f}$ that output from the k-th convolution block, and there are K blocks in total. $\max(\cdot)$ denotes channel-wise max-pooling, and $\mathrm{cat}[\cdot,\cdot]$ means concatenation. The encoder is shared weights for the inputs.

Dual Branches Regression

Given the distinctive features $\mathbf{F}^r$ and $\mathbf{F}^t$ of the inputs, we use a global feature interaction module to produce the hybrid features $\mathbf{H}^r$ and $\mathbf{H}^t$ for the rotation and translation respectively. Consistent with the encoder, a dual branches regression network is applied to regress the parameters for the rotation and the translation separately. Specifically, the rotation regression branch takes all the global features as inputs and produces a 4D vector, which represents the 3D rotation $\mathbf{R}$ in the form of quaternion (Shoemake 1985) $\mathbf{q} \in \mathbb{R}^4, \mathbf{q}^T\mathbf{q} = 1$. Meanwhile, rather than regressing the translation vector $\mathbf{t} \in \mathbb{R}^3$ directly, the translation regression branch produces two 3D vectors, which represent two saliency points of the source and the reference point clouds respectively, then calculating the difference between them as $\mathbf{t}$. At each iteration, the transformation $\{\mathbf{q}, \mathbf{t}\}$ is obtained as

$$\begin{aligned} \mathbf{q} = f^r_\theta(\mathrm{cat}[\mathbf{H}^r_{\mathcal{X}'}, \mathbf{H}^t_{\mathcal{X}'}, \mathbf{H}^r_\mathcal{Y}, \mathbf{H}^t_\mathcal{Y}]), \quad \mathbf{t} = \mathbf{c}_\mathcal{Y} - \mathbf{c}_{\mathcal{X}'}. \\ \mathbf{c}_{\mathcal{X}'} = f^t_\theta(\mathrm{cat}[\mathbf{H}^r_{\mathcal{X}'}, \mathbf{H}^t_{\mathcal{X}'}, \mathbf{H}^t_\mathcal{Y}]), \\ \mathbf{c}_\mathcal{Y} = f^t_\theta(\mathrm{cat}[\mathbf{H}^r_\mathcal{Y}, \mathbf{H}^t_\mathcal{Y}, \mathbf{H}^t_{\mathcal{X}'}]). \end{aligned} \tag{2}$$

Here, the functions $f^r_\theta(\cdot)$ and $f^t_\theta(\cdot)$ denote the rotation and translation regression networks. The vectors $\mathbf{c}_{\mathcal{X}'} \in \mathbb{R}^3$ and $\mathbf{c}_\mathcal{Y} \in \mathbb{R}^3$ mean the coordinates of the saliency points.

	Method	RMSE(**R**)		MAE(**R**)		RMSE(**t**)		MAE(**t**)		Error(**R**)		Error(**t**)	
		OS	*TS*	*OS*	*TS*	*OS*	*TS*	*OS*	*TS*	*OS*	*TS*	*OS*	*TS*
(a) Unseen Shapes	ICP	20.036	22.840	10.912	12.147	0.1893	0.1931	0.1191	0.1217	22.232	24.654	0.2597	0.2612
	Symmetric ICP	10.419	11.295	8.992	9.592	0.1367	0.1394	0.1082	0.1124	17.954	19.571	0.2367	0.2414
	FGR	48.533	46.766	29.661	29.635	0.2920	0.3041	0.1965	0.2078	55.855	57.685	0.4068	0.4263
	PointNetLK	23.866	27.482	15.070	18.627	0.2368	0.2532	0.1623	0.1778	29.374	36.947	0.3454	0.3691
	DCP	12.217	11.109	9.054	8.454	0.0695	0.0851	0.0524	0.0599	7.835	9.216	0.1049	0.1259
	RPMNet	1.347	2.162	0.759	1.135	0.0228	0.0267	0.0089	0.0141	1.446	2.280	0.0193	0.0302
	FMR	7.642	8.033	4.823	4.999	0.1208	0.1187	0.0723	0.0726	9.210	9.741	0.1634	0.1617
	RGM	3.470	4.912	1.251	1.786	0.0391	0.0428	0.0135	0.0183	2.441	3.506	0.0289	0.0393
	OMNet	0.771	1.384	0.277	0.542	0.0154	0.0226	0.0056	0.0093	0.561	1.118	0.0122	0.0198
	Ours	**0.694**	**1.267**	**0.198**	**0.269**	**0.0076**	**0.0168**	**0.0029**	**0.0048**	**0.383**	**0.591**	**0.0066**	**0.0110**
(b) Unseen Categories	ICP	20.387	22.906	12.651	13.599	0.1887	0.1994	0.1241	0.1286	25.085	26.819	0.2626	0.2700
	Symmetric ICP	12.291	12.333	10.841	10.746	0.1488	0.1456	0.1212	0.1186	21.399	21.437	0.2577	0.2521
	FGR	46.161	41.644	27.475	26.193	0.2763	0.2872	0.1818	0.1951	49.749	51.463	0.3745	0.4003
	PointNetLK	27.903	42.777	18.661	28.969	0.2525	0.3210	0.1752	0.2258	36.741	53.307	0.3671	0.4613
	DCP	13.387	12.507	9.971	9.414	0.0762	0.1020	0.0570	0.0730	11.128	12.102	0.1143	0.1493
	RPMNet	3.934	7.491	1.385	2.403	0.0441	0.0575	0.0150	0.0258	2.606	4.635	0.0318	0.0556
	FMR	10.365	11.548	6.465	7.109	0.1301	0.1330	0.0816	0.0837	12.159	13.827	0.1773	0.1817
	RGM	4.983	7.298	1.669	2.259	0.0402	0.0624	0.0164	0.0234	3.254	4.474	0.0348	0.0511
	OMNet	3.719	4.014	1.314	1.619	0.0392	0.0406	0.0151	0.0179	2.657	3.206	0.0321	0.0383
	Ours	**3.583**	**3.918**	**1.109**	**1.286**	**0.0324**	**0.0404**	**0.0115**	**0.0142**	**2.207**	**2.572**	**0.0245**	**0.0311**
(c) Gaussian Noise	ICP	20.566	21.893	12.786	13.402	0.1917	0.1963	0.1265	0.1278	25.417	26.632	0.2667	0.2679
	Symmetric ICP	12.183	12.576	10.723	10.987	0.1487	0.1478	0.1210	0.1203	21.169	21.807	0.2576	0.2560
	FGR	49.133	46.213	31.347	30.116	0.3002	0.3034	0.2068	0.2141	56.652	58.968	0.4230	0.4364
	PointNetLK	26.476	29.733	19.258	21.154	0.2542	0.2670	0.1853	0.1937	37.688	42.027	0.3831	0.3964
	DCP	13.117	12.730	9.741	9.556	0.0779	0.1072	0.0591	0.0774	11.350	12.173	0.1187	0.1586
	RPMNet	4.118	6.160	1.589	2.467	0.0467	0.0618	0.0175	0.0274	2.983	4.913	0.0378	0.0589
	FMR	10.604	11.674	6.725	7.400	0.1300	0.1364	0.0827	0.0867	12.627	14.121	0.1788	0.1870
	RGM	5.968	6.893	2.479	3.068	0.0583	0.0650	0.0247	0.0311	4.766	6.243	0.0515	0.0662
	OMNet	**3.572**	4.356	1.570	1.924	0.0391	0.0486	0.0172	0.0223	3.073	3.834	0.0359	0.0476
	Ours	3.676	**3.841**	**1.363**	**1.532**	**0.0327**	**0.0379**	**0.0130**	**0.0158**	**2.673**	**2.984**	**0.0273**	**0.0336**

Table 1: Results on ModelNet40 (using the partial manner of RPMNet). For each metric, *OS* and *TS* denote the results on the once-sampled and twice-sampled data. The best results are marked in **bold** for better comparison.

Multi-level Feature Interaction

For partial-to-partial registration, the network has to possess partiality perception ability, which means the feature extraction of the inputs need the information from each other.

Point-wise Feature Interaction. In the dual branches encoder, we insert the Point-wise Feature Interaction (PFI) after multiple convolution blocks. The point-wise features of one point cloud are first aggregated by channel-wise max-pooling, then broadcasted and concatenated with the point-wise features of the other point cloud at the same level. The encoder features are then updated as

$$ {}^{(k)}\mathbf{f}_{\mathcal{X}'} = {}^{(k)}g_\theta({\rm cat}[{}^{(k-1)}\mathbf{f}_{\mathcal{X}'},\, {\rm repeat}({\rm max}({}^{(k-1)}\mathbf{f}_{\mathcal{Y}}), N_{\mathcal{X}'})]), \tag{3} $$

where the function ${}^{(k)}g_\theta(\cdot)$ denotes the k-th convolution block, and ${\rm repeat}(\mathbf{z}, N)$ denotes repeating N times for the vector $\mathbf{z}$ at the element-wise dimension. The positions of inputs are inverted when extracting features for $\mathcal{Y}$.

Global Feature Interaction. Before regressing the transformation parameters, we further strengthen the information exchanging between the inputs by the Global Feature Interaction (GFI). The rotation and translation features are concatenated separately and sent into MLPs to generate the hybrid global features. The entire process is defined as

$$ \mathbf{H}^r_{\mathcal{X}'} = h^r_\theta({\rm cat}[\mathbf{F}^{\rm r}_{\mathcal{X}'}, \mathbf{F}^{\rm r}_{\mathcal{Y}}]), \quad \mathbf{H}^t_{\mathcal{X}'} = h^t_\theta({\rm cat}[\mathbf{F}^{\rm t}_{\mathcal{X}'}, \mathbf{F}^{\rm t}_{\mathcal{Y}}]), \tag{4} $$

where $h^r_\theta(\cdot)$ and $h^t_\theta(\cdot)$ denote the GFI functions for the rotation and translation . The positions of inputs are inverted when producing features for $\mathcal{Y}$.

Loss Functions

Transformation Sensitivity Loss. To modify the sensitivity of the encoder to the 3D transformation, we design the transformation sensitivity loss, which is a variant of the triplet loss (Dong and Shen 2018). It follows a simple intuition: the rotation branch should be more attentive to rotation and less attentive to translation, and vice versa for the translation branch. Hence, we cast the features of transformed source point cloud $\mathbf{F}^r_{\mathcal{X}'}$ and $\mathbf{F}^t_{\mathcal{X}'}$ as anchor features. For the rotation, $\mathcal{X}'$ is first rotated by the previous predicted $\mathbf{q}$ to form $\mathcal{X}'_r$, then sent into the dual branches encoder to extract features $\mathbf{F}^r_{\mathcal{X}'_r}$ and $\mathbf{F}^t_{\mathcal{X}'_r}$. Similarly, using $\mathbf{t}$ to alternate $\mathbf{q}$, we can obtain the translated point cloud $\mathcal{X}'_t$ and its features $\mathbf{F}^r_{\mathcal{X}'_t}$ and

$\mathbf{F}^t_{\mathcal{X}'_t}$. The loss function is $\mathcal{L}_s = \mathcal{L}^r_s + \mathcal{L}^t_s$, where

$$\mathcal{L}^r_s = \max(\,\|\mathbf{F}^r_{\mathcal{X}'} - \mathbf{F}^r_{\mathcal{X}'_t}\|_2 - \|\mathbf{F}^r_{\mathcal{X}'} - \mathbf{F}^r_{\mathcal{X}'_r}\|_2 + \delta,\ \|\mathbf{F}^r_{\mathcal{X}'} - \mathbf{F}^r_{\mathcal{X}'_t}\|_2). \tag{5}$$

Here, $\delta = 0.01$ denotes the margin. $\mathcal{L}^t_s$ is calculated with the features that output from the translation branch and exchanging the positive and negative pairs.

Point-wise Feature Dropout Loss. To avoid the case that the feature extractor only concentrates on the local geometry of the inputs, we design a point-wise feature dropout loss, which encourages the network to learn the global features from more dispersed regions of the inputs. Concretely, inspired by (Baldi and Sadowski 2013), we randomly set some of the point-wise features to zero before the max-pool layers in the encoder, and the distances between the global features that calculated before and after the dropout operation are constrained by the loss $\mathcal{L}_d = \mathcal{L}^{\mathcal{X}'}_d + \mathcal{L}^{\mathcal{Y}}_d$, where

$$\mathcal{L}^{\mathcal{X}'}_d = \|\mathbf{F}^r_{\mathcal{X}'} - \mathbf{F}^r_{\mathcal{X}'_d}\|_2 + \|\mathbf{F}^t_{\mathcal{X}'} - \mathbf{F}^t_{\mathcal{X}'_d}\|_2. \tag{6}$$

Here, the subscript d denotes the features are processed by dropout, and the superscript $\mathcal{X}'$ and $\mathcal{Y}$ denote the features belong to $\mathcal{X}'$ and $\mathcal{Y}$. The dropout ratio is set to 0.3.

Parameter Regression Loss. Following (Xu et al. 2021), we directly measure the deviation of $\{\mathbf{q}, \mathbf{t}\}$ from the ground truth. The 3D transformation parameter regression loss is

$$\mathcal{L}_p = |\mathbf{q} - \mathbf{q}_{gt}| + \lambda\,\|\mathbf{t} - \mathbf{t}_{gt}\|_2, \tag{7}$$

where subscript gt denotes the ground-truth. We notice that using the combination of ℓ^1 and ℓ^2 distance can marginally improve the performance. Besides, the factor λ is empirically set to 4.0 in all our experiments.

Combining the terms above after N iterations, we have the weighted sum loss

$$\mathcal{L}_{total} = \frac{1}{N}\sum_{i=1}^{N}(\mathcal{L}^i_p + \beta\,\mathcal{L}^i_s + \gamma\,\mathcal{L}^i_d), \tag{8}$$

where the factors β and γ are set to 10^{-3}.

Experiments

Dataset and Implementation Details

ModelNet40. (Wu et al. 2015) includes CAD models from 40 object categories. We use the data from OMNet (Xu et al. 2021), where 8 axisymmetrical categories are removed to avoid the ill-posed problem. We denote the data that sampled once from the CAD model as *OS*, while sampled twice separately as *TS*. We use the official train/test splits, resulting in 4,196 training, 1,002 validation, and 1,146 test objects. Two manners proposed by PRNet and RPMNet are applied to generate partially visible data.

7Scenes. (Shotton et al. 2013) is a widely used benchmark where data is captured by a Kinect camera in 7 indoor scenes. We use the code from (Zeng et al. 2017) to generate scan pairs with >30% overlap, and adopt the official train/test splits. We random sample 2,048 points to train our model, and set the voxel sample grid to 0.03 to train others.

Implementation Details. We run 4 iterations of alignment. Adam optimizer (Kingma and Ba 2015) is used with $l_r = 10^{-4}$. The batch size is 64, and training for 260k steps.

Method	RMSE(**R**)	RMSE(**t**)	Error(**R**)	Error(**t**)
ICP	18.588	0.0920	18.720	0.1026
Go-ICP	15.214	0.0566	9.002	0.0445
Symmetric ICP	7.096	6.280	0.0617	0.1191
FGR	33.723	0.1593	35.971	0.1828
PointNetLK	29.747	0.1841	32.760	0.1959
DCP	7.300	0.0389	8.853	0.0539
PRNet	5.883	0.0380	5.974	0.0472
FMR	5.304	0.0323	5.392	0.0342
DeepGMR	24.908	0.1057	25.830	0.1371
IDAM	8.008	0.0484	8.774	0.0578
OMNet	2.563	0.0183	2.360	0.0196
Ours	**2.378**	**0.0152**	**2.018**	**0.0138**

Table 2: Results on the TS unseen categories with Gaussian noise (using the partial manner of PRNet).

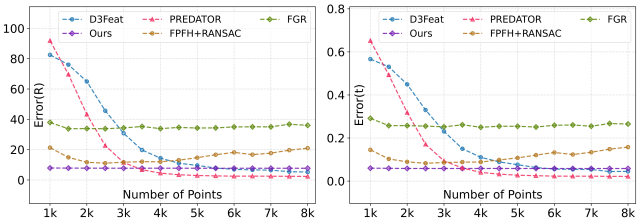

Figure 3: Errors on 7Scenes with different number of points.

Baseline Algorithms

We compare our method to the traditional methods: ICP (point-to-point), Go-ICP, Symmetric ICP and FGR, and the learning-based works: PointNetLK, DCP, RPMNet, FMR (supervised version), PRNet, IDAM, DeepGMR, RGM, OMNet, D3Feat and PREDATOR. We use the implementations of ICP and FGR in Intel Open3D (Zhou, Park, and Koltun 2018), Symmetric ICP in PCL (Rusu and Cousins 2011) and others released by their authors.

We measure root mean squared error (RMSE), mean absolute error (MAE), and the isotropic error (Error). Angular measurements are in units of degrees.

Evaluation on ModelNet40

Unseen Objects. We first evaluate the models on the same categories. Note that the input point clouds of the *TS* data have no exact correspondences. Table 1(a) shows the results, where our method ranks first in all measures. Go-ICP and DeepGMR fail to obtain reasonable results on the partially visible data of RPMNet due to the partial manner, so that we do not report their results. A qualitative comparison of the registration results can be found in Fig. 4(a).

Unseen Categories. To evaluate the generalization ability, we train the models on the first 14 categories and test on the remaining unseen categories. The results are summarized in Table 1(b). All the learning-based methods consistently perform worse compared with Table 1(a). However, the traditional methods are not affected so much because their handcrafted features are not sensitive to the shape variance. Our

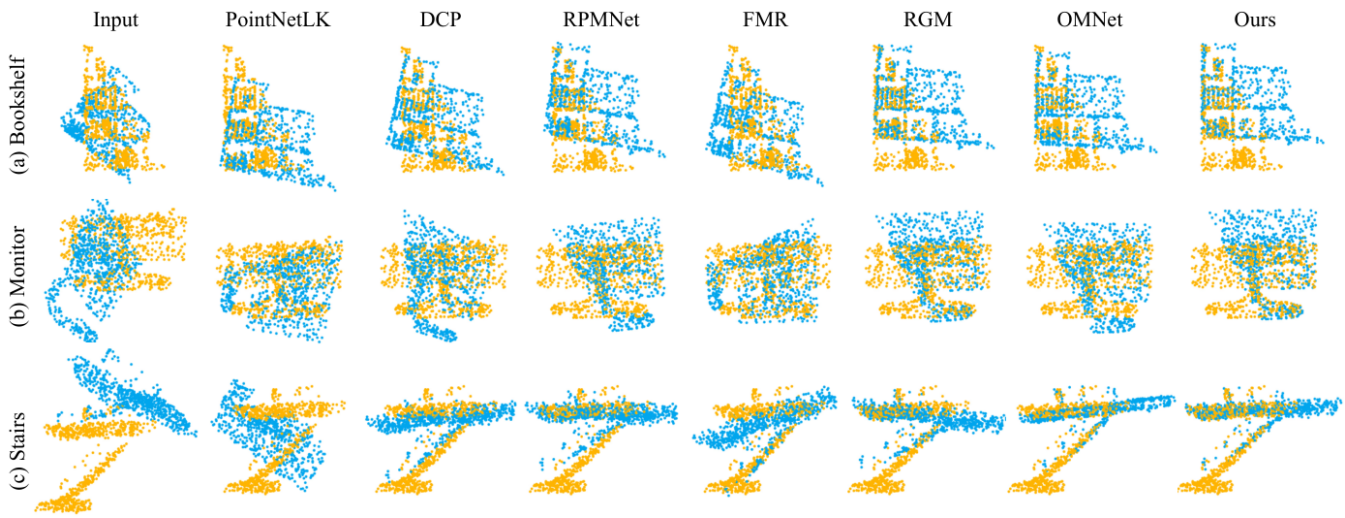

Figure 4: Qualitative examples on (a) Unseen objects, (b) Unseen categories, and (c) Gaussian noise.

		PFI	GFI	SP	PFDL	TSL	Error(**R**)	Error(**t**)
Single Branch	1)						4.850	0.0562
	2)	✓					3.982	0.0465
	3)		✓				4.575	0.0532
	4)	✓	✓				3.846	0.0448
	5)	✓	✓	✓			3.981	0.0489
	6)	✓	✓		✓		3.804	0.0410
Dual Branches	7)						4.624	0.0557
	8)	✓					3.799	0.0429
	9)		✓				3.666	0.0419
	10)	✓	✓				3.294	0.0387
	11)	✓	✓	✓			3.183	0.0370
	12)	✓	✓	✓	✓		3.082	0.0368
	13)	✓	✓	✓	✓	✓	**2.984**	**0.0336**

Table 3: Ablation studies.

method outperforms its competitors in all metrics. A qualitative example can be found in Fig. 4(b).

Gaussian Noise. In this experiment, we evaluate the robustness to noise. We test on the unseen categories and add noise that independently sampled from $\mathcal{N}(0, 0.01^2)$ and clipped to $[-0.05, 0.05]$ for each point. As shown in Table 1(c), all learning-based methods get worse with noises injected. Our method exhibits the best robustness. An example result is shown in Fig. 4(c).

Different Partial Manner. To valid the effectiveness to different partiality, we further test all the algorithms on the unseen categories used the partial manner in PRNet (Wang and Solomon 2019b). We retrain the leaning-based methods and the results are shown in Table 2. Since RPMNet and RGM use the different partially visible data from PRNet and IDAM, we evaluate them separately. Our method shows stronger robustness, and achieves the best performance.

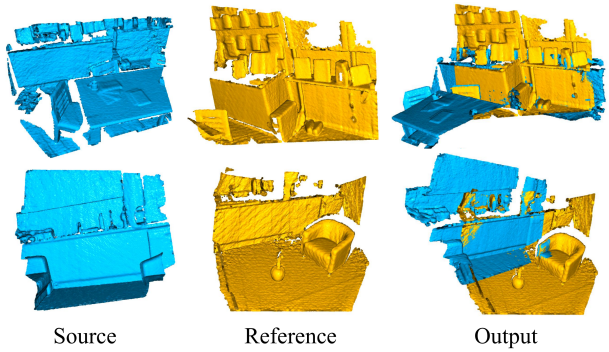

Figure 5: Qualitative results on 7Scenes (30∼50% overlap).

Evaluation on 7Scenes

We conduct experiments on the 7Scenes dataset. All scans are randomly sampled to the point clouds with different density, results on which are shown in Fig. 3. Our method is more robust than D3Feat and PREDATOR with respect to the density variation. Although they achieve higher precise with large inputs, our method is ∼10 times faster than them. Fig. 5 shows some qualitative results.

Ablation Studies

Dual Branches Structure. The dual branches structure (DB) is an important feature of our network. We replace it with a single branch structure (SB), where two branches are shared weights. For fairness, we double the output channel of each layer in SB to obtain a comparable number of parameters. Comparing Row 1 with 7 in Table 3, simply applying the DB structure only brings a slight improvement. However, only the DB structure can enable the ability of learning attentive features and enhance feature interactions.

Multi-level Feature Interaction. The multi-level feature interaction is another important aspect. We compare the performances in the case of different combinations of the PFI

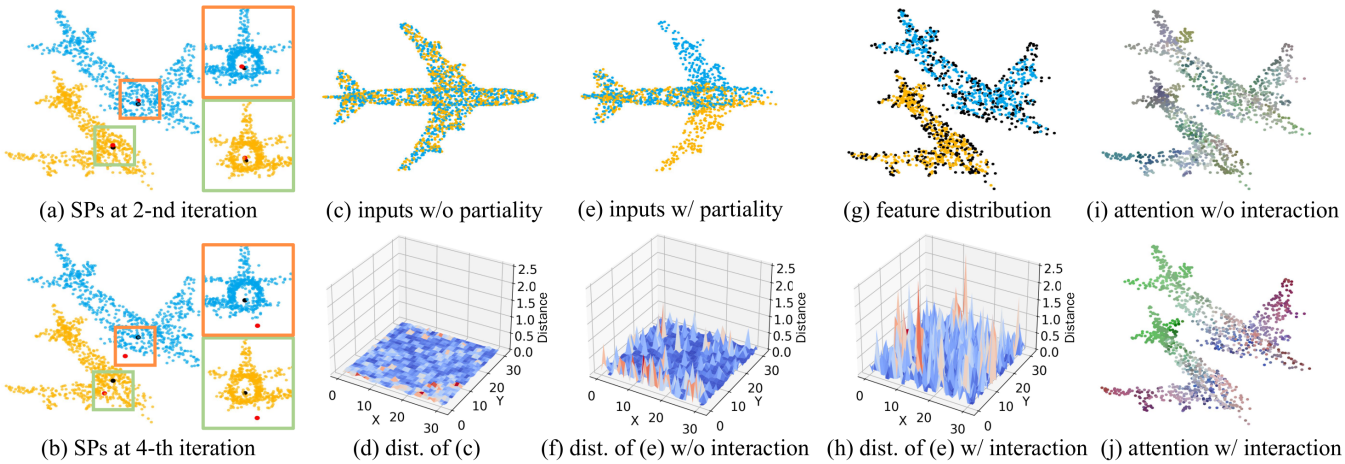

Figure 6: Visualization results in § and §. SP denotes saliency point and dist. means distance.

and GFI. Comparing Row 1∼4 in Table 3, we can see that only the PFI improves the performance with a large margin. Since the same encoder is applied for the rotation and translation in SB, it may confuse the GFI with their information blended. However, comparing Row 7∼10, both the PFI and GFI improve the performance in DB.

Loss Functions. We verify the effectiveness of our Pointwise Feature Dropout Loss (PFDL) and Transformation Sensitivity Loss (TSL). Comparing Row 4 with 6, and Row 11 with 12 in Table 3, PFDL improves the performance when applying on both SB and DB. Note that TSL can only be applied to DB. Comparing Row 12 with 13, it improves the performance with the TSL supervising, which supports the intuition that registration becomes more precise with the rotation and translation features more attentive to themselves, and it exists space difference between them.

Discussion

Translation Regression: Distance or SP?

We first try to answer why estimating the saliency points (SP) is better. Intuitively, directly regressing the translation $\mathbf{t}$ implements two steps implicitly: (1) estimate the SP for the inputs; (2) compute the difference between them as $\mathbf{t}$, which can be described as $\mathbf{R}\mathbf{c}_{\mathcal{X}} + \mathbf{t} = \mathbf{c}_{\mathcal{Y}}$,

$$\begin{cases} \mathbf{R} \not\approx \mathbf{E} & \text{s.t. } \mathbf{R}\mathbf{c}_{\mathcal{X}} = \mathbf{c}_{\mathcal{X}},\ \mathbf{c}_{\mathcal{X}} \leftrightarrow \mathbf{c}_{\mathcal{Y}} \\ \mathbf{R} \approx \mathbf{E} & \text{s.t. } \mathbf{c}_{\mathcal{X}} \leftrightarrow \mathbf{c}_{\mathcal{Y}} \end{cases} \Rightarrow \mathbf{t} = \mathbf{c}_{\mathcal{Y}} - \mathbf{c}_{\mathcal{X}}. \quad (9)$$

Here, $\leftrightarrow$ denotes the corresponding relationship, and $\mathbf{E}$ means the identity matrix. It requires that SP are the rotation centers when $\mathbf{R} \not\approx \mathbf{E}$, otherwise $\mathbf{c}_{\mathcal{X}}$ only needs to correspond to $\mathbf{c}_{\mathcal{Y}}$. Considering it exists a large rotation gap between the inputs, it is difficult to regress $\mathbf{t}$ directly. Comparing Row 4 with 5 in Table 3, the hybrid features of the rotation and the translation interfere with the regression of SP in SB. However, comparing Row 10 with 11, regressing SP in DB improves the performance. Therefore, as an explicit goal, regressing SP helps to reduce learning difficulty. We visualize the rotation centers and the predicted SP in black and red respectively, as shown in Fig. 6(a-b). The predicted SP close to the rotation center when it exists a rotation gap, otherwise merely form the correspondences.

Partiality Perception Ability

To explore the partiality perceptual ability, we calculate the distances between the global features of inputs. As shown in Fig. 6(f), it tries to extract the same features without the feature interaction. In contrast to concentrating on the overlapping regions, the network seems to learn how to utilize the partiality information to better register, resulting in the larger distance between features, as shown in Fig. 6(h).

Moreover, we visualize where the values in global feature come from, and the source points are shown in Fig. 6(g). Some of the non-overlapping points also have contributions because the spatial position of a particular part relative to the whole object can be used as prior information to help with registration. Finally, we show the attentive areas of the global features with and without the feature interaction, where the colors denote different iterations, as shown in Fig. 6(i-j). With the data association, the network tends to focus on distinct areas at each iteration. In contrast, the areas are almost the same without the feature interaction, which reflects the lack of partiality perception.

Conclusion

We have presented the FINet, an end-to-end global feature-based algorithm for adapting data association for the partial-to-partial point cloud registration. Our method possesses the multi-level feature interaction based on a dual branches structure, which enables effective early information exchange between the inputs. In addition, we design two loss functions to learn attentive features for the rotation and the translation respectively. Experimental results show the state-of-the-art performance and robustness of our method.

Acknowledgments

This work was supported by the National Natural Science Foundation of China (NSFC) under Grants No.62071097, No.61872067, and No.61720106004.

References

Aoki, Y.; Goforth, H.; Srivatsan, R. A.; and Lucey, S. 2019. PointNetLK: Robust & Efficient Point Cloud Registration using PointNet. In *Proc. CVPR*, 7163–7172.

Azuma, R. T. 1997. A Survey of Augmented Reality. *Presence: Teleoperators & Virtual Environments*, 6(4): 355–385.

Bai, X.; Luo, Z.; Zhou, L.; Fu, H.; Quan, L.; and Tai, C.-L. 2020. D3feat: Joint learning of dense detection and description of 3d local features. In *Proc. CVPR*, 6359–6367.

Baldi, P.; and Sadowski, P. J. 2013. Understanding dropout. In *Proc. NeurIPS*, 2814–2822.

Besl, P. J.; and McKay, N. D. 1992. A Method for Registration of 3D Shapes. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 14(2): 239–256.

Billinghurst, M.; Clark, A.; and Lee, G. 2014. A Survey of Augmented Reality. *Interaction*, 8(2-3): 73–272.

Choy, C.; Dong, W.; and Koltun, V. 2020. Deep global registration. In *Proc. CVPR*, 2514–2523.

Choy, C.; Park, J.; and Koltun, V. 2019. Fully convolutional geometric features. In *Proc. ICCV*, 8958–8966.

Dong, X.; and Shen, J. 2018. Triplet loss in siamese network for object tracking. In *Proc. ECCV*, 459–474.

Fang, H.-S.; Xu, Y.; Wang, W.; Liu, X.; and Zhu, S.-C. 2018. Learning pose grammar to encode human body configuration for 3d pose estimation. In *Proc. AAAI*.

Fischler, M. A.; and Bolles, R. C. 1981. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6): 381–395.

Fu, K.; Liu, S.; Luo, X.; and Wang, M. 2021. Robust Point Cloud Registration Framework Based on Deep Graph Matching. In *Proc. CVPR*, 8893–8902.

Huang, S.; Gojcic, Z.; Usvyatsov, M.; Wieser, A.; and Schindler, K. 2020a. PREDATOR: Registration of 3D Point Clouds with Low Overlap. *arXiv:2011.13005*.

Huang, X.; Mei, G.; and Zhang, J. 2020. Feature-Metric Registration: A Fast Semi-Supervised Approach for Robust Point Cloud Registration Without Correspondences. In *Proc. CVPR*, 11366–11374.

Huang, Z.; Yu, Y.; Xu, J.; Ni, F.; and Le, X. 2020b. PF-Net: Point fractal network for 3D point cloud completion. In *Proc. CVPR*, 7662–7670.

Jang, E.; Gu, S.; and Poole, B. 2016. Categorical Reparameterization with Gumbel-Softmax. *arXiv:1611.01144*.

Kingma, P. D.; and Ba, L. J. 2015. Adam: A Method for Stochastic Optimization. In *Proc. ICLR*.

Lin, C.-H.; Kong, C.; and Lucey, S. 2018. Learning efficient point cloud generation for dense 3d object reconstruction. In *Proc. AAAI*.

Lucas, B. D.; and Kanade, T. 1981. An Iterative Image Registration Technique with an Application to Stereo Vision. In *Proc. IJCAI*, 674–679.

Pomerleau, F.; Colas, F.; and Siegwart, R. 2015. A Review of Point Cloud Registration Algorithms for Mobile Robotics. *Foundations and Trends in Robotics*, 4(1): 1–104.

Qi, C. R.; Su, H.; Mo, K.; and Guibas, L. J. 2017a. PointNet: Deep Learning on Point Sets for 3D Classification and Segmentation. In *Proc. CVPR*, 652–660.

Qi, C. R.; Yi, L.; Su, H.; and Guibas, L. J. 2017b. Pointnet++: Deep Hierarchical Feature Learning on Point Sets in a Metric Space. In *Proc. NeurIPS*, 5099–5108.

Rusinkiewicz, S. 2019. A symmetric objective function for ICP. *ACM Trans. Graphics*, 38(4): 1–7.

Rusinkiewicz, S.; and Levoy, M. 2001. Efficient variants of the ICP algorithm. In *International Conference on 3-D Digital Imaging and Modeling (3DIM)*, 145–152.

Rusu, R. B.; Blodow, N.; and Beetz, M. 2009. Fast Point Feature Histograms (FPFH) for 3D registration. In *Proc. ICRA*, 3212–3217.

Rusu, R. B.; and Cousins, S. 2011. 3D is here: Point Cloud Library (PCL). In *Proc. ICRA*, 1–4.

Sarode, V.; Li, X.; Goforth, H.; Aoki, Y.; Srivatsan, R. A.; Lucey, S.; and Choset, H. 2019. PCRNet: Point Cloud Registration Network using PointNet Encoding. *arXiv:1908.07906*.

Segal, A.; Haehnel, D.; and Thrun, S. 2009. Generalized-ICP. In *Robotics: Science and Systems*, volume 2, 435.

Shoemake, K. 1985. Animating Rotation with Quaternion Curves. In *Annual Conference on Computer Graphics and Interactive Techniques*, 245–254.

Shotton, J.; Glocker, B.; Zach, C.; Izadi, S.; Criminisi, A.; and Fitzgibbon, A. 2013. Scene coordinate regression forests for camera relocalization in RGB-D images. In *Proc. CVPR*, 2930–2937.

Sinkhorn, R. 1964. A Relationship between Arbitrary Positive Matrices and Doubly Stochastic Matrices. *The Annals of Mathematical Statistics*, 35(2): 876–879.

Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, L. u.; and Polosukhin, I. 2017. Attention is All you Need. In *Proc. NeurIPS*, 5998–6008.

Wang, C.; Xu, D.; Zhu, Y.; Martín-Martín, R.; Lu, C.; Fei-Fei, L.; and Savarese, S. 2019a. Densefusion: 6d object pose estimation by iterative dense fusion. In *Proc. CVPR*, 3343–3352.

Wang, Y.; and Solomon, J. M. 2019a. Deep Closest Point: Learning Representations for Point Cloud Registration. In *Proc. ICCV*, 3523–3532.

Wang, Y.; and Solomon, J. M. 2019b. PRNet: Self-Supervised Learning for Partial-to-Partial Registration. In *Proc. NeurIPS*, 8814–8826.

Wang, Y.; Sun, Y.; Liu, Z.; Sarma, S. E.; Bronstein, M. M.; and Solomon, J. M. 2019b. Dynamic Graph CNN for Learning on Point Clouds. *ACM Trans. Graphics*, 38(5): 1–12.

Wu, Z.; Song, S.; Khosla, A.; Yu, F.; Zhang, L.; Tang, X.; and Xiao, J. 2015. 3D ShapeNets: A Deep Representation for Volumetric Shapes. In *Proc. CVPR*, 1912–1920.

Xu, H.; Liu, S.; Wang, G.; Liu, G.; and Zeng, B. 2021. OMNet: Learning Overlapping Mask for Partial-to-Partial Point Cloud Registration. In *Proc. ICCV*.

Yang, J.; Li, H.; and Jia, Y. 2013. Go-ICP: Solving 3D Registration Efficiently and Globally Optimally. In *Proc. CVPR*, 1457–1464.

Yew, Z. J.; and Lee, G. H. 2020. RPM-Net: Robust Point Matching using Learned Features. In *Proc. CVPR*, 11824–11833.

Yuan, W.; Eckart, B.; Kim, K.; Jampani, V.; Fox, D.; and Kautz, J. 2020. DeepGMR: Learning Latent Gaussian Mixture Models for Registration. In *Proc. ECCV*, 733–750.

Zeng, A.; Song, S.; Nießner, M.; Fisher, M.; Xiao, J.; and Funkhouser, T. 2017. 3dmatch: Learning local geometric descriptors from rgb-d reconstructions. In *Proc. CVPR*, 1802–1811.

Zhou, Q.-Y.; Park, J.; and Koltun, V. 2016. Fast Global Registration. In *Proc. ECCV*, 766–782.

Zhou, Q.-Y.; Park, J.; and Koltun, V. 2018. Open3D: A Modern Library for 3D Data Processing. *arXiv:1801.09847*.