Feature Interactive Representation for Point Cloud Registration

Bingli Wu, Jie Ma [*], Gaojie Chen, Pei An
National Key Laboratory of Science and Technology on Multi-spectral Information Processing,
School of Artificial Intelligence and Automation,
Huazhong University of Science and Technology, P.R.China
{bingliwu, majie, chengaojie, anpei}@hust.edu.cn

Abstract

Point cloud registration is the process of using the common structures in two point clouds to splice them together. To find out these common structures and make these structures match more accurately, interacting information of the source and target point clouds is essential. However, limited attention has been paid to explicitly model such feature interaction. To this end, we propose a Feature Interactive Representation learning Network (FIRE-Net), which can explore feature interaction among the source and target point clouds from different levels. Specifically, we first introduce a Combined Feature Encoder (CFE) based on feature interaction intra point cloud. The CFE extracts interactive features intra each point cloud and combines them to enhance the ability of the network to describe the local geometric structure. Then, we propose a feature interaction mechanism inter point clouds which includes a Local Interaction Unit (LIU) and a Global Interaction Unit (GIU). The former is used to interact information between point pairs across two point clouds, thus the point features in one point cloud and its similar point features in another point cloud can be aware of each other. The latter is applied to change the per-point features depending on the global cross information of two point clouds, thus one point cloud has the global perception of another. Extensive experiments on partially overlapping point cloud registration show that our method achieves state-of-the-art performance.

1. Introduction

Geometric registration is a key task in 3D data analysis, which aligns one point cloud (source) to another (target) by estimating the transformation between them. It has a variety of applications in many computational fields, including medical imaging, robotics, and autonomous driving. Iterative Closest Point (ICP) [3] is the most popular and widely-used registration algorithm, however, it often stalls in local minima and is only suitable for small transformations. Other methods [12, 30, 49, 55], which can register two point clouds with large rotation and translation, are typically slower than the original ICP.

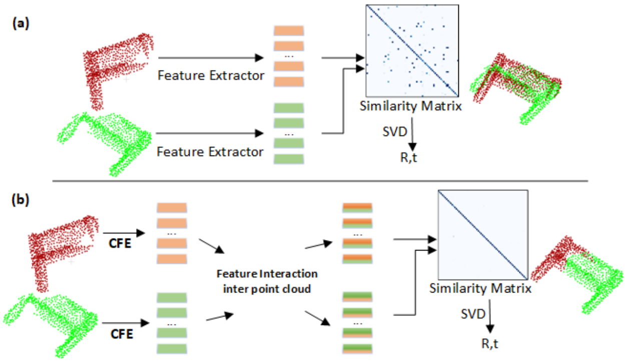

Figure 1: **Comparison of prior work and the proposed FIRE-Net.** (a) The pipeline without feature interaction [23], which use a feature extractor to extract features for source and target point clouds independently. (b) In our work, the multi-level feature interaction mechanism greatly improves the discriminative power of features, which enables a more discriminative similarity matrix and a more accurate registration result.

The past few years have seen a breakthrough in deep learning, leading to remarkable advancements in most 3D computer vision tasks, such as classification [28, 29, 40, 43, 47], segmentation [17, 44, 52, 53], and detection [5, 18, 22, 35]. Recently, some learning-based methods [19, 23, 41, 42, 50] for point cloud registration are proposed. They solve the registration problem in three steps: (1) extract features for the source and target point clouds using networks such as PointNet [28]; (2) design a module, such as pointer network [41], to find the correspondence; (3) apply a differentiable singular value decomposition (SVD) layer to find the least-squares rigid transformation. These methods show excellent performance with faster speed than traditional algorithms and have the ability to handle large rotations, but there are still problems. Many methods only focus on the matching stage and neglect the important cor-

[*] Corresponding author: *Jie Ma*.

nerstone of registration–feature extraction. Those methods simply use the existing feature extractors, such as PointNet [28] and DGCNN [43], which can't provide the discriminative feature for the subsequent matching process to resolve the ambiguity. We argue that by interacting the information between source and target point clouds and making the features of point clouds depend on each other, the features can be more discriminative and task-relevant. In this work, we formulate a novel feature interaction model, named FIRE-Net, to fully leverage interaction among the source and target point clouds from different levels. Specifically, it sequentially explores feature interaction intra each point cloud and feature interaction inter point clouds.

We first introduce a Combined Feature Encoder (CFE) to obtain interactive features intra each point cloud by using the graph neural networks (GNN) [15]. For better description of the local geometry structure, the CFE combines features from different propagation layers to capture comprehensive semantic and geometric information.

In addition to modeling feature interaction intra each point cloud, we observe that GNN can be naturally extended to feature interaction inter point clouds from the local perspective, namely Local Interaction Unit (LIU). Different from the CFE, we construct a hybrid graph with both source and target features and then update the node feature by aggregating information from each other. The key to LIU is making each point feature be adapted with respect to the point features in another point cloud.

Furthermore, we propose a Global Interaction Unit (GIU) to exploit feature interaction inter point clouds from the global perspective. The GIU crosses the information between source and target global features and automatically control the cross information transfer for both point clouds. With this process, all features of the source and target can complement each other, assisting registration in avoiding matching confusion and improving robustness.

We observe that, through comprehensive information interaction from different levels, the similarity matrix is more discriminative than the one obtained without feature interaction, as shown in Fig. 1. Meanwhile, our experiment results verify that FIRE-Net not only achieves state-of-the-art performance when the overlap rate of point clouds is high but also offers the largest improvements in the case of low overlap rates.

In summary, our main contributions are:

- We present a Combined Feature Encoder to extract interactive features in the local region and combine the features of different layers, enhancing the ability of the network to extract local geometric and semantic information.
- We design a novel feature interaction mechanism inter point clouds which enables each point cloud to have contextual awareness of another point cloud, thus providing more discriminative features for subsequent modules.
- Our end-to-end FIRE-Net model achieves state-of-the-art performance on the ModelNet40 benchmark in several settings, thus demonstrating its effectiveness and generalization ability.

2. Related Work

2.1. Learning on graphs and point clouds

Graph neural network (GNN) is a useful tool on non-Euclidean structures, which is first proposed in [15]. The target of GNN [10, 38, 46, 54] is to learn a state embedding that contains the information of neighborhood for each node. This process is similar to acquiring local features for each point in the point cloud. PointNet [28] is the pioneer of applying deep learning on the raw point cloud. It can be seen as applying GCN (Graph Convolutional Network) on graphs without edges, mapping points into high-dimensional space. As an alternative, DGCNN [43] can be regarded as a graph neural network applied to point clouds with dynamic edges. DensePoint [25] tries to use densely-connected blocks to encourage feature reuse and enhance feature propagation in 3D point clouds.

2.2. Traditional registration methods

Iterative Closest Point (ICP) [3] is a well-known rigid registration algorithm, which estimates point correspondence and performs a least-squares optimization iteratively. For each point in the source, it finds the closest neighbor in the target as the correspondence. Afterward, many variants [4, 34, 49] have been proposed based on the basic concept of ICP. Nevertheless, they may be stuck in multiple local minimums triggered by the improper initialization and generate the wrong final transformation [27, 30]. Go-ICP [49] uses a branch-and-bound approach to search for the globally optimal registration at the cost of longer computing time. Typical global registration methods are generally based on local features and RANSAC [11] algorithm. The representative handcraft descriptors include PFH [32], FPFH [31], and SHOT [33], etc [16, 48]. Recent studies use deep learning to learn these descriptors [1, 6, 8, 13, 20, 24, 36], such as 3DMatch [51] and PPFNet [9]. However, these methods are not end-to-end registration pipelines.

2.3. Learned registration methods

In recent years, many end-to-end learning-based point cloud registration frameworks are proposed. The pioneer is PointNetLK [19], which unrolls PointNet and the LK (Lucas & Kanade) algorithm [2, 26] into a single trainable recurrent deep neural network. Deep Closest Point (DCP) [41] incorporates DGCNN [43] and an attention module to extract features, following a pointer network [37, 39] predicts soft matching between the point clouds. PRNet [42] incorporates keypoint detection to handle partial visibility and leverages self-supervised learning to learn geometric priors directly from data. RPM-Net [50] extracts PPF [9]

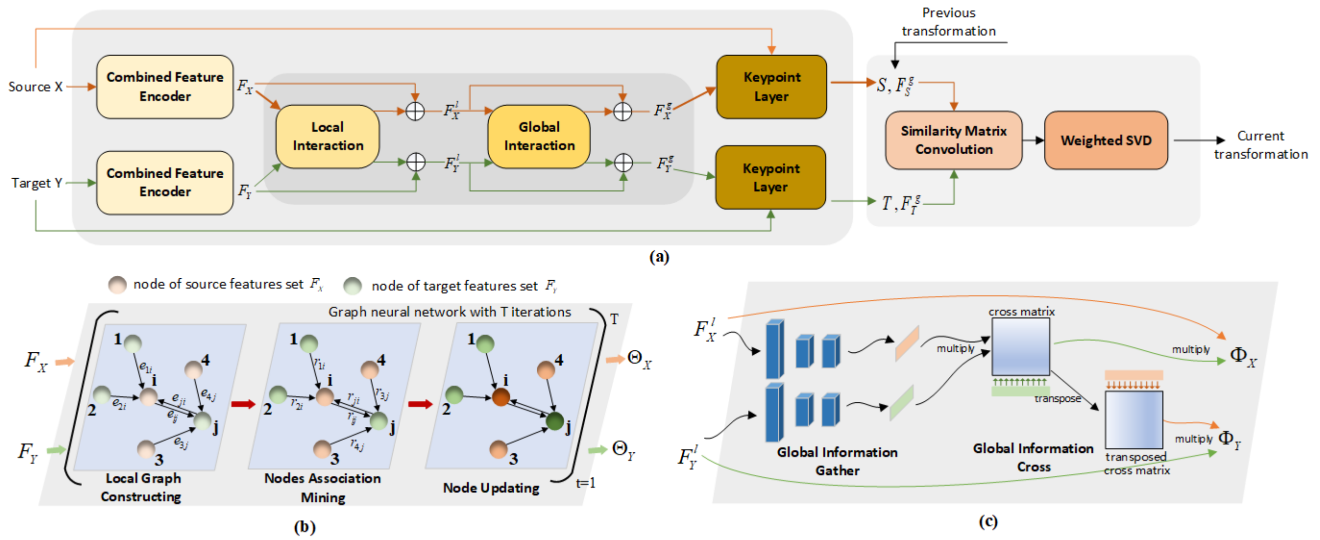

Figure 2: (a) Overview of our FIRE-Net, (b) Local Interaction Unit, and (c) Global Interaction Unit. Our LIU and GIU are designed as residual terms.

features as the initial input of the feature extraction network and implement RPM [14] algorithm using deep learning. IDAM [23] uses a distance-aware similarity matrix convolution module to incorporate information from both the feature and Euclidean space into the pairwise point matching process and includes a two-stage learnable point elimination technique to reduce computational complexity. Previous methods focus more on the matching stage, while our work focuses on the feature interaction and making two input point clouds be aware of each other.

3. Method

Given point clouds $X = \{x_i \in \mathbb{R}^3 | i = 1, \ldots, M\}$ and $Y = \{y_i \in \mathbb{R}^3 | i = 1, \ldots, N\}$, our objective is to find the rigid transformation $\{\boldsymbol{R}, \boldsymbol{t}\}$ which can align two input point clouds. $\boldsymbol{R} \in \mathrm{SO}(3)$ is a rotation matrix and $\boldsymbol{t} \in \mathbb{R}^3$ is a translation vector. The two point clouds can have a different number of points, i.e. $M \neq N$, or cover different extents.

Figure 2 shows an illustration of our FIRE-Net. Briefly, the input of our network includes source and target raw point clouds. The Combined Feature Encoder (CFE) (§3.1) first extracts initial features for the source and target, embedding all points into a common feature space. Then, a local interaction unit (LIU) and a global interaction unit (GIU) (§3.2) are designed serially for modeling the feature interaction inter point clouds. The final interacted features are fed to the keypoint layer to select the common structures in two point clouds (§3.3). Finally, we use a transformation computation module [23] to obtain reliable correspondence and compute the rigid transformation (§3.4) in an iterative way. The details of each module will be explained in the following sections.

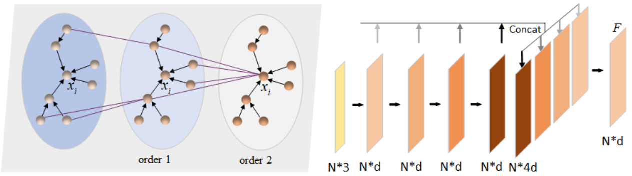

Figure 3: Combined Feature Encoder.

3.1. Combined Feature Encoder

The main objective of the CFE is to describe the local structures of the source and target point clouds. Meanwhile, as the first level of our feature interaction model, the CFE implements feature interaction intra point cloud. Altogether, our CFE is a GNN-based network that can hierarchically extract multi-order interactive features for points by constructing a fixed graph in 3D coordinate space and applying several propagation layers connected serially, as is illustrated in Fig. 3.

Local Graph in Coordinate Space. To represent the local geometry structure of the given point clouds $X = \{x_i \in \mathbb{R}^3 | i = 1, \ldots, M\}$ and $Y = \{y_i \in \mathbb{R}^3 | i = 1, \ldots, N\}$, we treat each point as a central node and construct a kNN graph respectively.

Propagation Layer. Assume x_i is a point in point cloud X, $N(x_i)$ is the set of k nearest neighbors in its local graph. Let $f_{x_i}^{(\ell)}$ be the feature for point x_i at the $(\ell)_{th}$ propagation layer. Then the output of the $(\ell+1)_{th}$ propagation layer can be expressed as Eq. 1. In particular, $f_{x_i}^{(0)}$ is the coordinate of x_i.

$$f_{x_i}^{(\ell+1)} = \sigma(\underset{x_j \in N(x_i)}{A}(g_1(f_{x_i}^{(\ell)} - f_{x_j}^{(\ell)}))), \qquad (1)$$

where g_1 is a shared MLP applied on the relative features of the center and its neighbors. A is max pooling. σ is a linear layer.

Initial Embedding To augment the module's ability to capture geometric and semantic information, we merge the low-order and high-order interactive features by concatenating the output vector of each propagation layer , as shown on the right of Fig. 3. Finally, we apply a shared MLP on the concatenated vectors to get initial features F_X and F_Y. This process can be formulated as:

$$\begin{aligned} F_X &= g_2(concat(F_X^1, F_X^2, ..., F_X^L)) \in \mathbb{R}^{M \times d}, \\ F_Y &= g_2(concat(F_Y^1, F_Y^2, ..., F_Y^L)) \in \mathbb{R}^{N \times d}, \end{aligned} \tag{2}$$

where g_2 is a shared MLP. F_Y^l denotes the feature in l_{th} layer and L denotes the number of propagation layers.

3.2. Feature Interaction inter point clouds

This module aims to model the interaction inter point clouds, which consists of two serial units: The LIU learns a local interaction function θ to obtain Local Interacted Embedding Θ: $\theta(F_X, F_Y) \rightarrow \Theta_X, \Theta_Y$. Then, the GIU learns a global interaction function ϕ to obtain Global Interacted Embedding Φ: $\phi(F_X^l, F_Y^l) \rightarrow \Phi_X, \Phi_Y$. Note that we treat Θ and Φ as residual terms , providing an additive change to the original features with learnable scale parameter α and β, that is,

$$F_X^l = F_X + \alpha \times \Theta_X \quad F_Y^l = F_Y + \alpha \times \Theta_Y, \tag{3}$$

$$F_X^g = F_X^l + \beta \times \Phi_X \quad F_Y^g = F_Y^l + \beta \times \Phi_Y, \tag{4}$$

where F_X^l and F_Y^l is the updated features after LIU, F_X^g and F_Y^g is the updated features after GIU.

3.2.1 Local Interaction Unit

With the premise that the CFE in §3.1 embeds the input point clouds into a common feature space, local interaction inter point clouds can be implemented by applying a well-designed GNN on feature space. See Fig. 2(b), we construct a hybrid graph with both source and target features and then update the node feature by aggregating information from neighboring node.

Local Graph in Feature Space. In the common feature space, we have M features for source point cloud (F_X) and N features for target point cloud (F_Y). We first construct a mixed feature set $F = \{F_X, F_Y\}$, then treat each feature f_i in F as a central node and apply kNN to construct a local graph G_i, thus obtaining a hybrid graph $G = \{G_i \in \mathbb{R}^{k \times d}, i = 1, 2, ..., (M + N)$. There are two strategies to construct the local graph G_i: (1) As shown on the left of Fig. 4, we connect all edges between the central node and its neighbors whether the neighbor feature belongs to F_X or F_Y. (2) As shown on the right of Fig. 4, the edge is only constructed between the nodes that belong to different point clouds. As a result, we have a sparse edge connection across source and target point clouds.

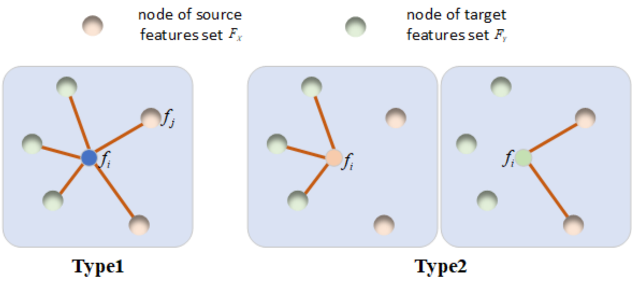
node of source features set F_x
node of target features set F_y
f_j
f_i
f_i
f_i
Type1
Type2

Figure 4: Local graph in feature space.

In the local graph, we treat the edges as the relation between two nodes, such as $e_{ji} = (f_j, f_i)$ indicates the relation from f_j to f_i. Thus in Fig. 4, type 1 models relation in feature space more comprehensively, yet type 2 is more clear than type 1 for passing the message from another point cloud to central node f_i. We will evaluate two graph construction types in the experiment(§4.2).

Nodes Association Mining. With the local graph defined above, the nodes association can be mined through the edge of the center node and its neighbors in node embedding space, e.g., use e_{ji} to obtain association vector r_{ji}. To find a more effective association function, we study three different forms of association mining and analyze how they affect the performance of registration.

$$\begin{aligned} &(Form \quad 1) \qquad r_{ji} = \sigma_1(f_j - f_i), \\ &(Form \quad 2) \qquad r_{ji} = \sigma_1(concat(f_i, f_j - f_i)), \\ &(Form \quad 3) \qquad r_{ji} = \sigma_1(f_i) + \sigma_2(f_j - f_i), \end{aligned} \tag{5}$$

where f_i denote the central feature and f_j is one of its neighbor feature. σ_1, σ_2 are linear layers.

We update the center's feature through aggregating association vector, that is:

$$f_i' = \underset{f_j \in G_(i)}{A} (r_{ji}), \tag{6}$$

where A is an aggregation function, such as max pooling.

Combined with the node updating formula, we can see that in form 1 of Eq. 5, the new node feature is entirely obtained from the difference of the features between node j and node i, which is limited because the feature of the node i itself is ignored. In form 2, we compensate for this limitation by concatenating the feature of the node itself with the difference of the features. While in form 3, we use two different linear layers for the feature of the node itself and the difference of the features. We will evaluate three nodes association mining forms in the experiment(§4.2).

Local Interacted Embedding. More generally, we repeat the above operations several times and get a high-level association representation $R \in \mathbb{R}^{(M+N) \times k \times d}$ and its corresponding aggregated output feature $F' \in \mathbb{R}^{(M+N) \times d}$. Note

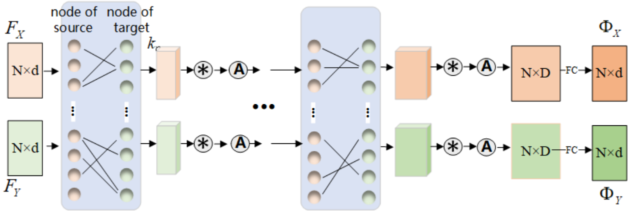

Figure 5: Detailed illustration of Local Interaction Unit.

that the local graph will be recomputed dynamically as the node feature updates, as shown in Fig. 5. This is a crucial distinction from CFE (§3.1) which works on a fixed input graph. Finally, we apply a linear layer σ_3 on the aggregated output feature F' to get the refined feature Θ, called Local Interacted Embedding. This process can be described as:

$$\Theta = \sigma_3(F') \in \mathbb{R}^{(M+N)\times d}. \tag{7}$$

Through the LIU, the feature of each point is informed with the feature of its local neighborhood. As a result, the point feature can lean towards the point with a similar feature (i.e., the matching point) and deviate from points with dissimilar features (i.e., the mismatched points). After the LIU, features of the source and the target are updated to F_X^l and F_Y^l as shown in Eq. 3, which are then bridged by the GIU.

3.2.2 Global Interaction Unit

To automatically share the global information and learn interactions between the source and the target comprehensively, we design a GIU as shown in Fig. 2(c). The proposed GIU obtains cross matrix of the source and target global features, and then updates point features by projecting the cross matrix into their latent representation spaces.

Global Information Gather. We first apply gather operation on F_X^l and F_Y^l to get the global features $f_{x_{global}}$ and $f_{y_{global}}$. The gather operation concatenates the pooled features and then refine them using a shared MLP, which can be expressed as follows:

$$\begin{aligned} f_{x_{global}} &= g_3(concat(maxpool(F_X^l), avgpool(F_X^l))) \\ f_{y_{global}} &= g_3(concat(maxpool(F_Y^l), avgpool(F_Y^l))), \end{aligned} \tag{8}$$

where g_3 is a shared MLP.

Global Information Cross. For source global feature $f_{x_{global}} \in \mathbb{R}^{1\times d}$ and target global feature $f_{x_{global}} \in \mathbb{R}^{1\times d}$, we then construct $d \times d$ pairwise interaction:

$$\begin{aligned} C &= f_{x_{global}}^T f_{y_{global}} \\ &= \begin{bmatrix} f_{x_{global}}^{(1)} f_{y_{global}}^{(1)} & \dots & f_{x_{global}}^{(1)} f_{y_{global}}^{(d)} \\ \dots & & \dots \\ f_{x_{global}}^{(d)} f_{y_{global}}^{(1)} & \dots & f_{x_{global}}^{(d)} f_{y_{global}}^{(d)} \end{bmatrix} \end{aligned} \tag{9}$$

where $C \in \mathbb{R}^{d\times d}$ is the cross feature matrix and d is the dimension of embedding features. Through the cross operation, each possible feature interaction $f_{x_{global}}^{(i)} f_{y_{global}}^{(j)}, \forall(i,j) \in \{1,\dots,d\}^2$ between source global feature $f_{x_{global}}$ and target global feature $f_{y_{global}}$ is modeled explicitly in the cross feature matrix.

Global Interacted Embedding. To project the information contained in the cross matrix into the per-point feature, we multiply the source feature $F_X^l \in \mathbb{R}^{M\times d}$ by the cross matrix $C \in \mathbb{R}^{d\times d}$, while multiply the target feature $F_Y^l \in \mathbb{R}^{N\times d}$ by the transposed cross matrix $C^T \in \mathbb{R}^{d\times d}$. This process can be expressed as Eq. 10, which provides the source Global Interacted Embedding Φ_X and target Global Interacted Embedding Φ_Y.

$$\begin{aligned} \Phi_X &= F_X^l \times C \in \mathbb{R}^{M\times d}, \\ \Phi_Y &= F_Y^l \times C^T \in \mathbb{R}^{N\times d}. \end{aligned} \tag{10}$$

Through the GIU, source and target point clouds can exchange global contextual information with each other. Simultaneously, the features in the source not only integrate the source global information but also the target global information and vice versa. After the GIU, features of the source and the target are updated to F_X^g and F_Y^g as shown in Eq. 4,

3.3. Keypoint Layer

Given the final interacted feature F_X^g and F_Y^g, we design a simple and efficient Keypoint Layer to select interest points that are shared in the partial common region of the source and target point clouds (Verified in Experiment §4.4). Our Keypoint Layer can be expressed as follows:

$$\begin{aligned} \mathcal{S} = X(topK(h(F_X^g))) \quad & F_\mathcal{S}^g = F_X^g(topK(h(F_X^g))), \\ \mathcal{T} = Y(topK(h(F_Y^g))) \quad & F_\mathcal{T}^g = F_Y^g(topK(h(F_Y^g))), \end{aligned} \tag{11}$$

where h is a shared MLP, and $h(F^g)$ output the significant score for each point. Here, $topK()$ extracts the indices of the K largest elements of the given input. $\mathcal{S}$, $\mathcal{T}$ are keypoint sets selected from the original point cloud. $F_\mathcal{S}^g$ and $F_\mathcal{T}^g$ denote the corresponding keypoint-feature sets.

3.4. Transformation Computation

Given the coordinates and features of point pairs, the transformation computation module employs a similarity matrix convolution (SMC) from IDAM [23] for regressing the similarity score of each point pair. The difference is that IDAM concatenates the features of point pairs as a part of the input to SMC, while our model uses the feature disparities of point pairs. The correspondences obtained by SMC are passed to the SVD layer to compute the transformation. Overall, the transformation computation process iterates T times.

4. Experiments

Dataset. We evaluate our model on the ModelNet40 dataset [45] which is composed of 12,311 CAD models from 40 object categories. In keeping with previous work, we use the pre-processed data from [28]. Only the (x, y, z) coordinates of the sampled points are used. Following [42], we randomly sample 1,024 points from each model's outer surface and sample rotations by sampling three Euler angle rotations in the range $[0, 45°]$ and translations in the range $[-0.5, 0.5]$ on each axis during training and testing. We transform the source point cloud X using the sampled rigid transform and the task is to register it to the unperturbed reference point cloud Y.

For partial-to-partial registration, we follow the same method as PRNet [42], which fixes a random point far away from the two point clouds X and Y, and preserve 768 points closest to the far point for each point cloud. It gives point clouds with a mean overlap rate (MOR) of 96%.

For getting point cloud pairs with lower mean overlap rate, we place the far point for X and Y independently, which provides point clouds with MOR of 82%.

Evaluation Metric. We measure root mean squared error (RMSE), and mean absolute error (MAE) between ground truth values and predicted values. Ideally, all of these error metrics should be zero if the rigid alignment is perfect. All angular measurements in our results are in units of degrees.

Network. We use 4 propagation layers in the CFE and the local graph is constructed using $k = 12$ nearest neighbors. In the LIU, we iterate the interaction process for 2 times and the dimensions of node embedding in each iteration are [128, 256], the kNN graph of type 1 uses $k = 10$ and the graph of type 2 uses $k = 5$. In the Global Information Gather process of the GIU, we use a 3-layer MLP with output dimensions [128, 64, 64]. Both CFE, LIU, and GIU output features of dimension $d = 64$. In the Keypoint Layer, we use a 3-layer MLP with output dimensions [64, 32, 1] and the number of keypoints preserved is 128. The transformation computation process iterates with $T = 3$.

Training Protocol. FIRE-Net is trained on one GTX 2080 Ti GPU with the batch size 16. Adam [21] is used to optimize the network parameters, with an initial learning rate of 0.0001. We divide the learning rate by 10 at epoch 30, training for a total of 40 epochs.

4.1. Partial Registration

We compare our method to ICP, Go-ICP [49], FGR [55], FPFH+RANSAC, PointNetLK [19], DCP [41], PRNet [42] and IDAM [23]. All the data-driven methods are trained on the same training set. We use the metrics mentioned above to evaluate all these methods. For demonstrating the robustness to the lower overlap rate of our model, we also experiment on inputs with a lower mean overlap rate. Note that for lower overlap rate registration experiments, we only focus on comparing learning-based algorithms.

Model	MOR	RMSE($\boldsymbol{R}$)	MAE($\boldsymbol{R}$)	RMSE($\boldsymbol{t}$)	MAE($\boldsymbol{t}$)
ICP	96%	33.68	25.05	0.293	0.250
Go-ICP [49]	96%	14.00	3.17	0.033	0.012
FGR [55]	96%	11.24	2.83	0.030	0.008
FPFH+RANSAC	96%	2.33	1.96	0.015	0.008
PointNetLK [19]	96%	16.74	7.55	0.045	0.025
DCP-v2 [41]	96%	6.71	4.45	0.027	0.020
PRNet [42]	96%	3.20	1.45	0.016	0.003
IDAM [23]	96%	2.95	0.76	0.021	0.005
FIRE-Net (ours)	96%	**0.95**	**0.27**	**0.006**	**0.001**
PointNetLK [19]	82%	27.62	14.37	0.186	0.101
DCP-v2 [41]	82%	8.87	6.44	0.085	0.061
IDAM [23]	82%	14.28	4.90	0.031	0.013
FIRE-Net (ours)	82%	**2.21**	**0.74**	**0.007**	**0.002**

Table 1: Test on unseen objects in ModelNet40.

Model	MOR	RMSE($\boldsymbol{R}$)	MAE($\boldsymbol{R}$)	RMSE($\boldsymbol{t}$)	MAE($\boldsymbol{t}$)
ICP	96%	33.68	25.56	0.293	0.250
Go-ICP [49]	96%	12.53	2.94	0.031	0.010
FGR [55]	96%	9.93	1.95	0.038	0.007
FPFH+RANSAC	96%	2.11	1.82	0.015	0.013
PointNetLK [19]	96%	22.94	9.66	0.061	0.033
DCP-v2	96%	9.77	6.95	0.034	0.025
PRNet	96%	4.99	2.33	0.021	0.015
IDAM	96%	3.42	0.93	0.022	0.005
FIRE-Net (ours)	96%	**0.82**	**0.24**	**0.006**	**0.001**
PointNetLK [19]	82%	30.60	15.93	0.191	0.105
DCP-v2 [41]	82%	10.83	8.23	0.117	0.090
IDAM [23]	82%	15.87	9.57	0.133	0.085
FIRE-Net (ours)	82%	**2.12**	**0.69**	**0.009**	**0.002**

Table 2: Test on unseen categories in ModelNet40.

Unseen Objects. We first experiment on the ModelNet40 train/test split, which has 9843 training objects and 2468 testing objects from all the 40 categories. Table 1 shows that our method outperformed all the counterparts. When MOR=82%, our network performs better consistently and leads by a large margin, e.g. FIRE-Net is 6× better than the second place.

Unseen Categories. We then evaluate the generalization ability to unseen categories. Data-driven methods are trained on the first 20 categories and then all algorithms are tested on the held-out categories. This experiment tests the capability to generalize to point clouds of unseen categories. Table 2 shows FIRE-Net behaves more robust than other models in both MOR=96% and MOR=82% cases, as all others tend to perform much worse than experiment on unseen objects.

Unseen Objects with Gaussian Noise. Further, we evaluate the performance in the presence of noise, which is present in real-world point clouds. The same preprocessing is done as in the first experiment (Unseen Objects), except that we randomly jitter the points in both point clouds by gaussian noises sampled from N(0, 0.01) and clipped to [-

Model	MOR	RMSE($\boldsymbol{R}$)	MAE($\boldsymbol{R}$)	RMSE($\boldsymbol{t}$)	MAE($\boldsymbol{t}$)
ICP	96%	35.07	25.56	0.294	0.250
Go-ICP [49]	96%	12.26	2.85	0.028	0.029
FGR [55]	96%	27.67	13.79	0.070	0.039
FPFH+RANSAC	96%	5.06	4.19	0.021	0.018
PointNetLK [19]	96%	19.94	9.08	0.057	0.032
DCP-v2	96%	6.88	4.53	0.028	0.021
PRNet	96%	4.32	2.05	0.017	0.012
IDAM	96%	**3.72**	1.85	**0.023**	0.011
FIRE-Net (ours)	96%	4.11	**1.71**	0.024	**0.009**
PointNetLK [19]	82%	38.99	21.26	0.245	0.142
DCP-v2 [41]	82%	9.28	6.56	0.092	0.068
IDAM [23]	82%	9.60	5.29	0.100	0.054
FIRE-Net (ours)	82%	**5.76**	**2.87**	**0.026**	**0.013**

Table 3: Test on unseen objects with Gaussian noise in ModelNet40.

0.05, 0.05]. The results of this experiment are summarized in Table 3. We can see that FIRE-Net generally performs better than traditional methods and learning-based methods. When MOR=96%, our model achieves comparable results with the IDAM [23], but our model outperforms it in the case of MOR=82%.

Performance under different Overlap Rates Lastly, we explore the performance of the registration model under different overlap rates. We divide the test set into different intervals according to the overlap rate and then calculate the success rate of registration in each interval. As mentioned in the introduction, IDAM is sensitive to overlap rate, and our FIRE-Net is more robust with the use of feature interaction which is confirmed in Fig. 6. Note that the intervals may be of unequal length as we try to make every interval have a similar number of samples.

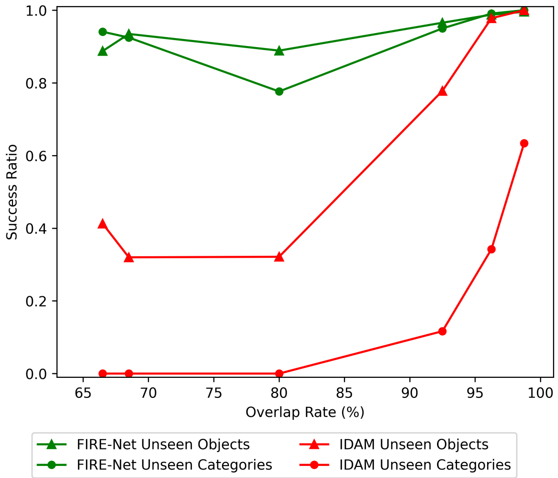

Figure 6: **Performance under different Overlap Rate.** A registration is counted as successful if the final alignment rotation error is less than 5 degrees and translation error is less than 0.01. Comparing with IDAM, FIRE-Net can still achieve a high success ratio when the overlap rate is reduced. Moreover, the figure also shows that FIRE-Net has stronger generalization ability to unseen categories.

4.2. Ablation Study

In this section, we conduct several ablation experiments to investigate the effect of each essential component of FIRE-Net. All experiments are done under the settings of “unseen objects with Gaussian noise” with MOR=82% as described in §4.1.

Effectiveness of our FIRE-Net. We examine three key components of our FIRE-Net: Combined Feature Encoder (CFE), Local Interaction Unit (LIU), and Global Interaction Unit (GIU). We use BS to denote the model that does not contain any of the three components. As shown in Table 4, all components indeed bring significant performance improvements, especially the combination of LIU and GIU.

Local Graph Constructing Strategy in the LIU. In Table 5, we show that using type 2 is better than type 1 for local graph construction as type 2 is more clear for interaction between source and target nodes.

Model	RMSE($\boldsymbol{R}$)	MAE($\boldsymbol{R}$)	RMSE($\boldsymbol{t}$)	MAE($\boldsymbol{t}$)
BS	14.30	8.14	0.118	0.073
BS + CFE	12.67	7.24	0.118	0.071
BS + CFE + LIU	11.48	4.13	0.032	0.016
BS + CFE + GIU	11.74	5.26	0.064	0.032
BS + CFE + LIU + GIU	**5.76**	**2.87**	**0.026**	**0.013**

Table 4: Effectiveness of our FIRE-Net (§4.2)

Component	method	RMSE($\boldsymbol{R}$)	MAE($\boldsymbol{R}$)	RMSE($\boldsymbol{t}$)	MAE($\boldsymbol{t}$)
Local Graph in the LIU	type 1	10.03	4.29	0.057	0.027
	type 2	**5.76**	**2.87**	**0.026**	**0.013**
Association Mining in the LIU	form 1	7.30	3.71	0.030	0.016
	form 2	6.57	3.37	0.028	0.014
	form 3	**5.76**	**2.87**	**0.026**	**0.013**
Global Feature Obtaining in the GIU	maxpool	6.69	3.32	0.027	0.014
	meanpool	6.04	3.01	0.028	0.014
	{maxpool,meanpool}	**5.76**	**2.87**	**0.026**	**0.013**

Table 5: Comparison of different choices of each component)

Nodes Association Mining Forms in the LIU. As shown in the middle column of Table. 5 , form 1 gets the worst result as the feature of the center node itself is ignored after aggregation. Form 3 is better than form 2, because the former treats the feature of the node itself and the difference of the features respectively, avoiding the feature of the central node dominates the association learning.

Global Feature Obtaining Method in the GIU. We observe that combining maxpooled feature and averagepooled feature together has the best performance because it helps our model capture important information and keep complete information in global features.

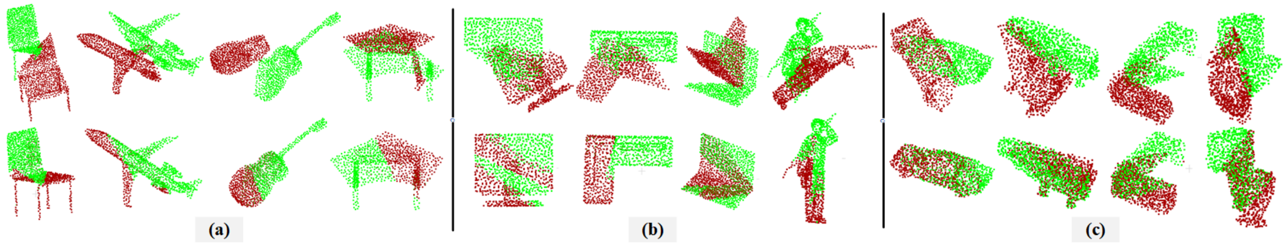

Figure 7: Qualitative results on modelnet40. (a) unseen objects. (b) unseen categories. (c) unseen objects with gaussian noise. (top: initial positions, bottom: registration results)

Model	size(MB)	Infer(s)	MAE($\boldsymbol{R}$)	MAE($\boldsymbol{t}$)
PointNetLK	0.594	0.079	15.93	0.105
DCP-v2	21.40	0.021	8.23	0.090
IDAM	**0.38**	0.017	9.57	0.085
FIRE-Net(ours)	1.02	**0.014**	**0.69**	**0.002**

Table 6: Space and time complexity results on ModelNet40.

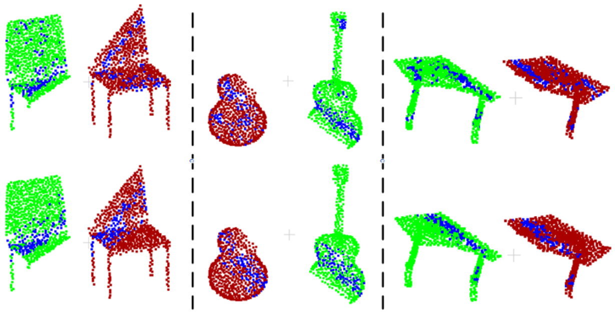

Figure 8: **Visualization of the keypoints** In each cell separated by the vertical line, the top row shows the keypoints detected by the model without LIU and GIU, and the bottom row shows the keypoints of our FIRE-Net.(red: source point cloud, green: target point cloud, blue: keypoints)

4.3. Space and Time Complexity Analysis

We test the space and time complexity of our model on point clouds with 1024 points and compare to PointNetLK [19], DCP [41], IDAM [23]. We use the official implementation of PointNetLK, DCP, and IDAM released by the authors.The experiments are done on a machine with an Intel Core i9-9900K CPU, a single Nvidia GeForce RTX 2080 Ti GPU, and 64G memory. We use a batch size of 1 for all the models. The speed is measured in seconds per frame and is computed by averaging 1266 results (the test set in the experiment of Unseen Categories). Table 6 summarizes the results (we relist the registration results for easy comparison). Obviously, our method achieves state-of-the-art performance with minuscule model size and faster inference speed. Note that our model performs 10× better than IDAM while space and time complexity is at the same level.

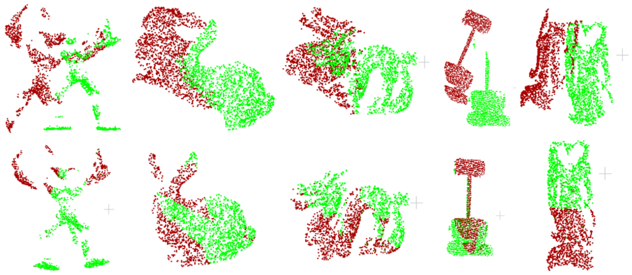

Figure 9: Qualitative results on Stanford 3D Scan dataset. (top: initial positions, bottom: registration results)

4.4. Visualization

In this section, we show some visualization results registering two partially overlapping point clouds in ModelNet40. This corresponds to the “Partial Registration” experiment (§4.1)) in the paper, as shown in Fig. 7. Moreover, we visualize keypoints on several objects in Fig. 8. With the interaction-based feature extraction module, most of the keypoints detected by FIRE-Net are located in the overlapped area of the source and target point clouds, yet the keypoints detected by the model without LIU and GIU are very scattered. We also test the same model (§4.1 Unseen Objects) on the Stanford 3D Scan dataset [7]. We use the same method described in the paper to generate partially overlapping point clouds with a lower overlap rate. Note that we only train on the ModelNet40 dataset and no fine-tuning. The results in Fig. 9 show the generalization ability of our model.

5. Conclusion

In this paper, we propose a point cloud registration method named FIRE-Net for partially overlapping point cloud registration. FIRE-Net dynamically enhances the feature’s discriminative power by exploiting the interaction of different levels. Extensive experiments on partially overlapping point cloud registration demonstrate the effectiveness of each component in our proposed network and show our method yields state-of-the-art 3D registration performance.

References

[1] X. Bai, Z. Luo, L. Zhou, H. Fu, L. Quan, and C. L. Tai. D3feat: Joint learning of dense detection and description of 3d local features. *IEEE*, 2020. 2

[2] Simon Baker and Iain Matthews. Lucas-kanade 20 years on: A unifying framework. *International journal of computer vision*, 56(3):221–255, 2004. 2

[3] Paul J. Besl and Neil D. McKay. A method for registration of 3-d shapes. *IEEE Trans. Pattern Anal. Mach. Intell.*, 14(2):239–256, Feb. 1992. 1, 2

[4] Sofien Bouaziz, Andrea Tagliasacchi, and Mark Pauly. Sparse iterative closest point. *Computer Graphics Forum*, 32(5), 2013. 2

[5] Xiaozhi Chen, Huimin Ma, Ji Wan, Bo Li, and Tian Xia. Multi-view 3d object detection network for autonomous driving. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. 1

[6] Christopher Choy, Jaesik Park, and Vladlen Koltun. Fully convolutional geometric features. In *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, 2019. 2

[7] Brian Curless and Marc Levoy. A volumetric method for building complex models from range images. *ACM*, pages 303–312, 1996. 8

[8] Haowen Deng, Tolga Birdal, and Slobodan Ilic. Ppf-foldnet: Unsupervised learning of rotation invariant 3d local descriptors. In *European Conference on Computer Vision*, 2018. 2

[9] Haowen Deng, Tolga Birdal, and Slobodan Ilic. Ppfnet: Global context aware local features for robust 3d point matching. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2018. 2

[10] David K Duvenaud, Dougal Maclaurin, Jorge Iparraguirre, Rafael Bombarell, Timothy Hirzel, Alan Aspuru-Guzik, and Ryan P Adams. Convolutional networks on graphs for learning molecular fingerprints. In C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett, editors, *Advances in Neural Information Processing Systems 28*, pages 2224–2232. Curran Associates, Inc., 2015. 2

[11] Martin A. Fischler and Robert C. Bolles. Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Commun. ACM*, 24(6):381–395, June 1981. 2

[12] A. W. Fitzgibbon. Robust registration of 2D and 3D point sets. In *British Machine Vision Conference*, pages 662–670, 2001. 1

[13] Z. Gojcic, C. Zhou, Jan D Wegner, and A. Wieser. The perfect match: 3d point cloud matching with smoothed densities. 2018. 2

[14] Steven Gold, Anand Rangarajan, Chien-Ping Lu, Suguna Pappu, and Eric Mjolsness. New algorithms for 2d and 3d point matching: pose estimation and correspondence. *Pattern recognition*, 31(8):1019–1031, 1998. 3

[15] Marco Gori, Gabriele Monfardini, and Franco Scarselli. A new model for learning in graph domains. In *IJCNN*, 2005. 2

[16] Yulan Guo, Ferdous A. Sohel, Mohammed Bennamoun, Jianwei Wan, and Min Lu. Rops: A local feature descriptor for 3d rigid objects based on rotational projection statistics. In *International Conference on Communications*, 2013. 2

[17] Wenkai Han, Chenglu Wen, Cheng Wang, Xin Li, and Qing Li. Point2node: Correlation learning of dynamic-node for point cloud feature modeling. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(7):10925–10932, 2020. 1

[18] Chenhang He, Hui Zeng, Jianqiang Huang, Xian Sheng Hua, and Lei Zhang. Structure aware single-stage 3d object detection from point cloud. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. 1

[19] R. Arun Srivatsan Hunter Goforth, Yasuhiro Aoki and Simon Lucey. Pointnetlk: Robust & efficient point cloud 7 registration using pointnet. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, LA, USA, June 2019. 1, 2, 6, 7, 8

[20] Marc Khoury, Qian-Yi Zhou, and Vladlen Koltun. Learning compact geometric features. In *Proceedings of the IEEE international conference on computer vision*, pages 153–161, 2017. 2

[21] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980, 2015. 6

[22] Jason Ku, Melissa Mozifian, Jungwook Lee, Ali Harakeh, and Steven L. Waslander. Joint 3d proposal generation and object detection from view aggregation. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2019. 1

[23] Jiahao Li, Changhao Zhang, Ziyao Xu, Hangning Zhou, and Chi Zhang. Iterative distance-aware similarity matrix convolution with mutual-supervised point elimination for efficient point cloud registration. In *Computer Vision – ECCV 2020*, pages 378–394, Cham, 2020. Springer International Publishing. 1, 3, 5, 6, 7, 8

[24] L. Li, S. Zhu, H. Fu, P. Tan, and C. L. Tai. End-to-end learning local multi-view descriptors for 3d point clouds. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. 2

[25] Yongcheng Liu, Bin Fan, Gaofeng Meng, Jiwen Lu, Shiming Xiang, and Chunhong Pan. Densepoint: Learning densely contextual representation for efficient point cloud processing. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5239–5248, 2019. 2

[26] B. D Lucas. An iterative image registration technique with an application to stereo vision (darpa). *Proc Ijcai*, 81(3):674–679, 1981. 2

[27] Fran?Ois Pomerleau, Francis Colas, and Roland Siegwart. A review of point cloud registration algorithms for mobile robotics. *Foundations & Trends in Robotics*, 4(1):1–104, 2015. 2

[28] Charles R Qi, Hao Su, Kaichun Mo, and Leonidas J Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. 1, 2, 6

[29] Charles R. Qi, Li Yi, Hao Su, and Leonidas J. Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. In *Advances in Neural Information Processing Systems*, page 5099–5108, 2017. 1

[30] Szymon Rusinkiewicz and Marc Levoy. Efficient variants of the ICP algorithm. In *Third International Conference on 3D Digital Imaging and Modeling (3DIM)*, June 2001. 1, 2

[31] Radu Bogdan Rusu, Nico Blodow, and Michael Beetz. Fast point feature histograms (fpfh) for 3d registration. In *IEEE*

International Conference on Robotics & Automation, 2009. 2

[32] Radu Bogdan Rusu, Nico Blodow, Zoltan Csaba Marton, and Michael Beetz. Aligning point cloud views using persistent feature histograms. In *2008 IEEE/RSJ International Conference on Intelligent Robots and Systems, September 22-26, 2008, Acropolis Convention Center, Nice, France*, 2008. 2

[33] Samuele Salti, Federico Tombari, and Luigi Di Stefano. Shot: Unique signatures of histograms for surface and texture description. *Computer Vision & Image Understanding*, 125(AUG.):251–264, 2014. 2

[34] Aleksandr Segal, Dirk Hhnel, and Sebastian Thrun. Generalized-icp. In *Robotics: Science and Systems V, University of Washington, Seattle, USA, June 28 - July 1, 2009*, 2009. 2

[35] Weijing Shi, Ragunathan, and Rajkumar. Point-gnn: Graph neural network for 3d object detection in a point cloud. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. 1

[36] R. Spezialetti, S. Salti, and L. D. Stefano. Learning an effective equivariant 3d descriptor without supervision. In *International Conference on Computer Vision*. 2

[37] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. Attention is all you need. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 5998–6008. Curran Associates, Inc., 2017. 2

[38] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. Graph Attention Networks. *International Conference on Learning Representations*, 2018. 2

[39] Oriol Vinyals, Meire Fortunato, and Navdeep Jaitly. Pointer networks. In C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett, editors, *Advances in Neural Information Processing Systems 28*, pages 2692–2700. Curran Associates, Inc., 2015. 2

[40] Peng Shuai Wang, Yang Liu, Yu Xiao Guo, Chun Yu Sun, and Xin Tong. O-cnn: Octree-based convolutional neural networks for 3d shape analysis. *Acm Transactions on Graphics*, 36(4):72, 2017. 1

[41] Yue Wang and Justin M. Solomon. Deep closest point: Learning representations for point cloud registration. In *International Conference on Computer Vision (ICCV)*, 2019. 1, 2, 6, 7, 8

[42] Yue Wang and Justin M. Solomon. Prnet: Self-supervised learning for partial-to-partial registration. In *33rd Conference on Neural Information Processing Systems (To appear)*, 2019. 1, 2, 6

[43] Yue Wang, Yongbin Sun, Ziwei Liu, Sanjay E Sarma, Michael M Bronstein, and Justin M Solomon. Dynamic graph cnn for learning on point clouds. *ACM Transactions on Graphics*, 38(5), 2018. 1, 2

[44] Jiacheng Wei, Guosheng Lin, Kim Hui Yap, Tzu Yi Hung, and Lihua Xie. Multi-path region mining for weakly supervised 3d semantic segmentation on point clouds. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. 1

[45] Null Zhirong Wu, Shuran Song, Aditya Khosla, Null Fisher Yu, Null Linguang Zhang, Null Xiaoou Tang, and Jianxiong Xiao. 3d shapenets: A deep representation for volumetric shapes. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015. 6

[46] Zonghan Wu, Shirui Pan, Fengwen Chen, Guodong Long, Chengqi Zhang, and Philip S. Yu. A comprehensive survey on graph neural networks. *CoRR*, abs/1901.00596, 2019. 2

[47] Zhirong Wu, Shuran Song, Aditya Khosla, Fisher Yu, Linguang Zhang, Xiaoou Tang, and Jianxiong Xiao. 3d shapenets: A deep representation for volumetric shapes. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015. 1

[48] Jiaqi Yang, Zhiguo Cao, and Qian Zhang. A fast and robust local descriptor for 3d point cloud registration. *Information Sciences*, pages 163–179, 2016. 2

[49] Jiaolong Yang, Hongdong Li, Dylan Campbell, and Yunde Jia. Go-icp: A globally optimal solution to 3d icp point-set registration. *IEEE Trans. Pattern Anal. Mach. Intell.*, 38(11):2241–2254, Nov. 2016. 1, 2, 6, 7

[50] Zi J. Yew and Gim H. Lee. Rpm-net: Robust point matching using learned features. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. 1, 2

[51] Andy Zeng, Shuran Song, Matthias Nießner, Matthew Fisher, Jianxiong Xiao, and Thomas Funkhouser. 3dmatch: Learning local geometric descriptors from rgb-d reconstructions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1802–1811, 2017. 2

[52] Jiazhao Zhang, Chenyang Zhu, Lintao Zheng, and Kai Xu. Fusion-aware point convolution for online semantic 3d scene segmentation. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. 1

[53] Yang Zhang, Zixiang Zhou, Philip David, Xiangyu Yue, and Hassan Foroosh. Polarnet: An improved grid representation for online lidar point clouds semantic segmentation. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. 1

[54] Jie Zhou, Ganqu Cui, Zhengyan Zhang, Cheng Yang, Zhiyuan Liu, and Maosong Sun. Graph neural networks: A review of methods and applications. *CoRR*, abs/1812.08434, 2018. 2

[55] Qian-Yi Zhou, Jaesik Park, and Vladlen Koltun. Fast global registration. In *Computer Vision - ECCV 2016 - 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part II*, pages 766–782, 2016. 1, 6, 7