STORM: Structure-based Overlap Matching for Partial Point Cloud Registration

Yujie Wang, Chenggang Yan, Yutong Feng, Shaoyi Du, *Member, IEEE,* Qionghai Dai, and Yue Gao, *Senior Member, IEEE*

Abstract—Partial point cloud registration aims to transform partial scans into a common coordinate system. It is an important preprocessing step to generate complete 3D shapes. Although previous registration methods have made great progress in recent decades, traditional registration methods, such as Iterative Closest Point (ICP) and its variants, all these methods highly depend on the sufficient overlaps between two point clouds, because they cannot distinguish outlier correspondences. Note that the overlap between point clouds could always be small, which limits the application of these methods. To tackle this problem, we present a StrucTure-based OveRlap Matching (STORM) method for partial point cloud registration. In our method, an overlap prediction module with differentiable sampling is designed to detect points in overlap utilizing structure information, and facilitates exact partial correspondence generation, which is based on discriminative pointwise feature similarity. The pointwise features which contain effective structural information are extracted by graph-based methods. Experimental results and comparison with state-of-the-art methods demonstrate that STORM can achieve better performance. Moreover, most registration methods perform worse when the overlap ratio decreases, while STORM can still achieve satisfactory performance when the overlap ratio is small.

Index Terms—Point Cloud Registration, Partial Registration, Overlap Matching, Point Cloud Sampling

1 Introduction

With the development of 3D scanner devices, point cloud has become an important domain in 3D computer vision. Point cloud registration [1] predicts a rigid motion to transform two or more point clouds into a common coordinate system, and is a key technology in 3D computer vision. It has been applied in various areas, such as Simultaneous Localization and Mapping (SLAM), autonomous driving, *etc.* In recent decades, much research efforts [1], [2], [3], [4] have been dedicated to point cloud registration. Iterative Closest Point (ICP) [1] is a well-known point cloud registration algorithm, which provides a basic registration procedure that alternately finds point-to-point correspondences, and computes a rigid motion based on the correspondences by singular value decomposition (SVD) [5]. Given correct correspondences, the optimal rigid motion can be exactly obtained by SVD. Therefore, the key to point cloud registration is correspondence estimation. For correspondence generation, ICP simply establishes correspondences for all points based on spatial distances, so it cannot always obtain a global optimal solution and resolve the partial point cloud problem.

- *Yujie Wang and Chenggang Yan are with the School of Mechanical, Electrical and Information Engineering, Shandong University, Weihai 264209, China. Chenggang Yan is also with the School of Automation, Hangzhou Dianzi University, Hangzhou 310018, China. Yutong Feng, Qionghai Dai and Yue Gao are with BNRist, THUIBCS, BLBCI, Tsinghua Universitty, Beijing 100084, China. Yutong Feng and Yue Gao are also with KLISS, School of Software, Tsinghua University, Beijing 100084, China. Shaoyi Du is with Institute of Artificial Intelligence and Robotics, College of Artificial Intelligence, Xi'an Jiaotong University, Xi'an 710049, China.*
  Yue Gao is the corresponding author.

A group of methods [6], [7], [8], [9], [10], [11] introduce local or global feature descriptors to generate exact corresponding point pairs. Most of these methods rely on traditional outlier rejection methods, e.g., RANSAC [12], to remove outliers and predict precise rigid motions. Due to the limitation of RANSAC, how to effectively conduct registration is still an open problem. In addition, 3DRegNet [13] and Deep Global Registration [14] propose correspondence prune methods and perform transformation estimation. Both of them require appropriate feature extraction methods to generate putative correspondences. Recently, much attention has been drawn to use an end-to-end neural network for point cloud registration. PointNetLK [15] applies a modified Lucas & Kanade (LK) algorithm to PointNet [16]. It learns network representations to directly predict a rigid motion without seeking correspondences. Deep Closest Point (DCP) [17] employs graph-based methods to learn structural pointwise features, and then finds corresponding points for all points with a pointer generation layer. DCP and PointNetLK perform well when the input point clouds are complete. However, point clouds are usually partial and contain outliers. Both DCP and PointNetLK cannot perform outlier rejection, and fail to handle partial-to-partial registration.

To tackle the partial registration problem, MaskNet [18] presents a fully-convolutional neural network to perform overlap prediction, employed as a preprocessing step for partial point cloud registration, removing points in non-overlap to improve registration performance. However, MaskNet is limited to the input of a partial point cloud and a complete point cloud. Yew *et al.* [19] proposed a Robust Point Matching Network (RPM-Net) to predict correspondences for partial point clouds through a Sinkhorn [20] layer. It relies on point cloud normal estimation methods

due to its requirement of additional geometric information for feature computation. Partial Registration Network (PRNet) [21], an iterative framework for partial-to-partial registration, performs keypoint detection to find the common points of the input point clouds based on the L2-norm of features. However, it only performs well when the two input point clouds have a large overlap. The poor performance of PRNet on point clouds with small overlap indicates that the L2-norm of features cannot exactly reflect the common structure of the input point clouds.

It is noted that the overlap between two point clouds cannot be always large. Actually, it could be small due to the occlusion situation or inaccurate scans. We empirically find that the points in non-overlap have a large negative impact on the performance of partial registration. To tackle this challenge, we propose a StrucTure-based OveRlap Matching method, called *STORM*, for partial point cloud registration. To exactly detect the overlap in point clouds, we perform overlap prediction with differentiable sampling based on contextual structural information of the input point clouds. More specifically, the overlap prediction module draws samples from the probability distribution. This distribution incorporates the structural dependencies of point clouds in different poses, indicating the latent common structure of the input point clouds to avoid the negative impact of points in non-overlap on registration performance. Besides, to satisfy the permutation-invariance of point cloud in sampling, we independently sample each point from the predicted distribution.

The feature extraction in STORM does not require additional information. Given plain point clouds, our method employs densely-connected EdgeConv [22] layers and two Transformer [17], [21], [23] layers to extract discriminative pointwise features of structural information. The densely-connected layers are used to map the point clouds to a high-dimensional feature space, modeling the structural information of 3D shapes, and the Transformer layers are employed to further capture long-term dependencies of fine-grained structural information between the input point clouds. To detect the point cloud overlap, an overlap prediction module with differentiable sampling is introduced to detect the common points of point clouds based on the pointwise features and facilitate partial correspondence generation. Equipped with the above modules, our method iteratively predicts a more precise rigid motion in a coarse-to-fine way. To evaluate the performance of our proposed STORM method, we have conducted experiments on the ModelNet40 [24] dataset and the outdoor KITTI [25] dataset. The experimental results demonstrate that the proposed STORM method can achieve comparable performance compared to the state-of-the-art registration methods. Moreover, we have further investigated the influence of different overlap rates on the performance of partial point cloud registration. We can observe that most existing methods perform worse when the overlap ratio decreases, or even do not work at all, while the proposed STORM method can still achieve satisfactory partial registration performance when the overlap ratio is small, demonstrating the effectiveness and robustness of the proposed method.

Our contributions can be summarized as follows.

- We propose a structure-based overlap matching method with differentiable sampling to perform overlap prediction and partial correspondence generation for partial point cloud registration.
- Given plain point clouds, the discriminative pointwise features are extracted by graph-based methods which utilize structural information. Our method can predict a more precise rigid motion in a coarse-to-fine way.
- We have evaluated the influence of different overlap ratios on the registration performance. From the results, we can observe that most existing methods fail when the overlap ratio is small, while the proposed method can still work well.

2 Related Work

2.1 Point Cloud Registration

Point cloud registration aims to yield a transformation for aligning the input point clouds together. Iterative Closest Point (ICP) [1] is widely employed to solve the rigid registration problem. It alternately generates point pairs based on the spatial distance of points and recovers the rigid transformation from the generated point pairs using least-squares [5]. However, ICP is easily prone to local optima and cannot distinguish inlier or outlier correspondences. Therefore, some following methods [2], [3], [26] are designed to improve registration performance based on the ICP algorithm. Generalized-ICP [2] incorporates a probabilistic module into an ICP-like framework to increase the robustness of algorithm. Go-ICP [3] combines the ICP algorithm with the Branch-and-Bound (BnB) procedure. It searches for an optimal solution in the entire 3D motion space $SE(3)$. To tackle outliers and noises in point clouds, the maximum correntropy criterion (MCC) [27] is integrated to the ICP pipeline. Besides, Trimmed ICP (TrICP) [26] introduces the Least Trimmed Squares (LTS) into each part of the ICP pipeline to tackle the partial overlap problem. Based on TrICP, Wang $et\,al.$ [28] proposed a parallel point feature histogram (PPFH) descriptor and a parallel TrICP to perform partial registration in a coarse-to-fine way, improving the accuracy and efficiency of registration. All these registration algorithms aim to solve the difficulties in the ICP pipeline, but they cannot provide an ideal solution without a good initial estimation.

Recently, there is an increasing interest in developing an end-to-end neural network for point cloud registration. PointNetLK [15] incorporates a modified Lucas Kanade (LK) algorithm into the PointNet [16] framework to iteratively align the input point clouds. Deep Closest Point (DCP) [17] establishes correspondences for all points based on learned pointwise features in a single shot. The satisfactory performance of DCP on the ModelNet40 [24] dataset shows that learned features of structural information can facilitate point cloud registration. To further tackle the partial overlap problem, PRNet [21] performs keypoint detection to select the points that the input point clouds have in common. OPRNet [29] resorts to the Sinkhorn algorithm [30] tailored to partial-to-partial registration. Li $et\,al.$ [31] presented a two-stage point elimination technique to help generate partial correspondences, but this registration method still has a poor performance when the point cloud overlap is low. OMNet [32] proposes a global feature based iterative network to

predict rigid transformations, and learns masks to reject the points in non-overlap for partial-to-partial registration. The mask prediction method alleviates the negative impact of points in non-overlap regions on registration performance. PointGMM [33] employs a hierarchical Gaussian mixture model (hGMM) which provides additional geometric and structural information. The proposed novel framework for rigid registration based on hGMM learns to disentangle orientation from the input point clouds. However, it is limited to taking the partial point clouds rotating in a specific direction as input. RPM-Net [19] is proposed to be less sensitive to outliers and noise. This registration method is based on 4D point pair features (PPF) [12], [34] which require exact point normals. In this paper, we show that learned features which do not require additional geometric information, are enough to yield exact partial correspondences for point clouds with low overlaps and large initial transformations.

2.2 Graph-based Feature Learning

Employing graph-based methods to process point clouds has gained much attention [22], [35], [36], [37], [38], [39], [40], [41]. These approaches regard the point cloud as a graph to model the structural information of 3D shapes. Graph convolutions are first applied to point cloud classification in [39]. Moreover, KCNet [35] extracts the local structural features using kernel correlation layers. To capture semantically similar structures, Wang $et\,al.$ [37] proposed an EdgeConv operation to extract the local structural information of points and updated the graph dynamically using the k-Nearest Neighbors (k-NN) algorithm after each EdgeConv. Based on dynamic graphs, LDGCNN [22] links the hierarchical features from different layers to avoid the vanishing gradient problem. However, the above graph-based methods define standard convolution and neglect the differences between neighbor points. Therefore, GAC [38] introduces an attention mechanism into graph convolution networks to utilize fine-grained structural information for semantic segmentation of point clouds. The attention mechanism is widely used in both computer vision and neural language process (NLP). Transformer [23] is based on stacked multi-head attention modules and achieves great performance in NLP. In this paper, we utilize multiple EdgeConv [37] layers and Transformer layers to perform feature extraction and refinement. The generated pointwise features contain contextual structural information and facilitate the generation of exact partial correspondence.

2.3 Partial Correspondences

The rigid motion obtained from corresponding point pairs is much more precise than that directly predicted by stacked Multilayer Perception (MLP) layers [17], [42]. Therefore, the key to the partial point cloud registration problem is to find partial correspondences between two point sets. Traditional feature-based registration methods [12], [43] employ handcrafted feature descriptors to detect keypoints and generates keypoint-to-keypoint correspondences. With the development of deep learning technologies, 3D feature descriptors can also be learned by neural networks [6], [7], [8], [9], [34], [44], [45]. However, they rely on traditional outlier rejection methods, e.g., RANSAC [12], to remove outlier correspondences, and thus are time-consumed. KeyPointNet [46] shows that 3D keypoints can be learned in a self-supervised way. Inspired by KeyPointNet, to tailor to partial-to-partial point cloud registration, PRNet [21] introduces a self-supervised keypoint detection based on L2-norm of pointwise features to find the common points of input point clouds. To tackle the problem of small overlap in point clouds, we follow this self-supervised idea but select the common points by sampling from the distribution predicted by our feature extraction module.

2.4 Point Sampling

Point sampling often aims to select a representative subset of point cloud [40], [47], [48], [49], [50]. PointNet++ [47] employs farthest point sampling (FPS), a traditional sampling technology to enlarge points' receptive fields, while SampleNet [49], a differentiable sampling network, is applied in point cloud classification and reconstruction tasks. Grid-GCN [40] and RandLA-Net [50] employ point sampling techniques to tackle large-scale point cloud scenes. To yield a more representative subset of point clouds with low computation cost, Gumbel Subset Sampling (GSS) [48] is designed to sample virtual points based on Gumbel-Softmax [51] for point cloud classification and semantic segmentation. However, from the above, few sampling techniques are employed in point cloud registration. In this paper, we propose an overlap prediction module with differentiable sampling for partial point cloud registration.

3 METHOD

In this section, we introduce our proposed STORM method for partial point cloud registration. First, we briefly introduce the overview of STORM. Then, we introduce our feature extraction amd refinement method which utilizes the structural information to yield pointwise features. Third, we describe the details of our proposed overlap prediction module for partial-to-partial registration. Fourth, we introduce the method of pose estimation. Fifth, the loss functions of our framework are given. Finally, we provide the implementation details of our method.

3.1 Overview

For partial point cloud registration, our proposed STORM method iteratively matches the points based on pointwise feature similarity. The whole framework is illustrated in Figure 1. Given the input point clouds $\mathcal{X}$ and $\mathcal{Y}$, we perform feature extraction and refinement to generate pointwise features. Specifically, multiple densely-connected EdgeConv layers [22] and two Transformer [17], [21], [23] layers with an encoder-decoder architecture are used to extract the contextual structural information of point clouds. Between the Transformer layers, our proposed overlap prediction module is employed to sample the points which represent the point cloud overlap from a predicted distribution. After that, we perform pose estimation based on the generated pointwise features. The affinity matrix is computed based on feature similarity. To tackle the noise in real scans, the softmax function is used to express the correspondences in probability space to yield virtual partial point-to-point

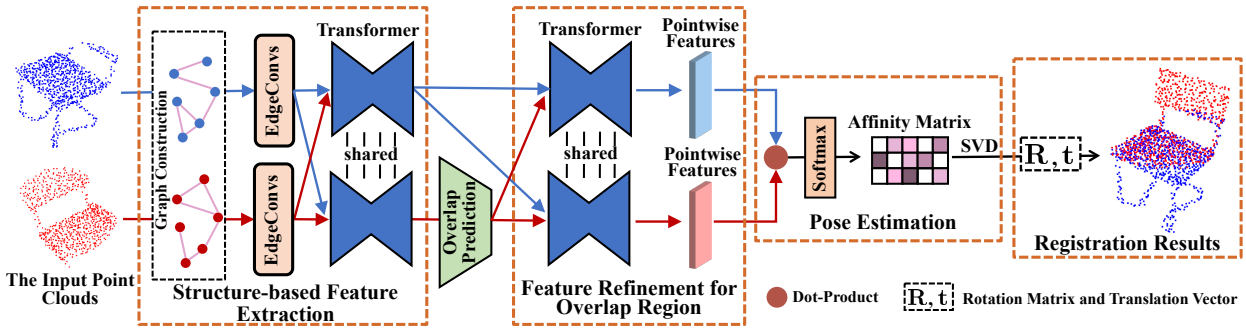


Fig. 1: The overall pipeline of STORM. Given the input point clouds, STORM first performs structure-based feature extraction. The multiple densely-connected EdgeConv layers and a Transformer layer are employed to map the input points to a feature space. Then, we perform overlap prediction to detects the points in common between the input point clouds. After that, a Transformer layer is employed to perform feature refinement for overlap region and generate pointwise features containing structural information. Finally, we perform pose estimation based on the generated pointwise features. Specifically, we generate virtual partial correspondences based on feature similarity. The correspondences are formulated as an affinity matrix, and $\boldsymbol{R}$ and $\mathbf{t}$ can be obtained by SVD. Utilizing $\boldsymbol{R}$ and $\mathbf{t}$ aligns the input point clouds together.

correspondences. Given correspondences, the rigid motion incorporating a rotation matrix $\boldsymbol{R} \in \mathbb{R}^{3\times3}$ and a translation vector $\mathbf{t} \in \mathbb{R}^3$ can be obtained using singular value decomposition (SVD) [5]. STORM first predicts a coarse rigid motion, and then fine-tunes it. In this way, $\mathcal{X}$ and $\mathcal{Y}$ are iteratively aligned together.

3.2 Feature Extraction and Refinement

To yield exact point-to-point correspondences, the learned pointwise features are required to be more discriminative. The discriminative pointwise features are supposed to contain special structural information. To this end, we employ multiple densely-connected EdgeConv [22] layers and two Transformer [17], [21], [23] layers as our feature extraction and refinement method to generate structure-based pointwise features.

Given unaligned point clouds $\mathcal{X}$ and $\mathcal{Y}$, we first perform graph construction to utilize structure information. Specifically, we regard 3D points as vertexes in graphs, and edges are yielded by k-Nearest Neighbors (k-NN) algorithm based on the spatial distance of 3D points. Then, the D-dimension pointwise feature of vertex i at the l-th layer $\mathbf{x}_i^l \in \mathbb{R}^D$ can be calculated by EdgeConv [37] operation:

$$\mathbf{x}_i^l = f(\left\{h_\theta^l(\mathbf{x}_i^{l-1}, \mathbf{x}_i^{l-1} - \mathbf{x}_j^{l-1}) \forall j \in \mathcal{N}_i\right\}), \tag{1}$$

where $\mathcal{N}_i$ denotes the neighbors of vertex i in this graph and h_θ^l denotes the forward mechanism which embeds relative position information into a high-dimensional space. h_θ^l can be implemented with MLP in practice. f denotes the aggregation function e.g., max.

Moreover, it is noted that the vertexes' receptive field is limited in a static graph. For a vertex in a static graph, it can only acquire long-term dependencies by its neighbors. This scheme could bring the risk of over-smoothing and the reduction of structural information. To enlarge points' receptive field, the graph is updated at each layer by using the k-NN algorithm in feature space. Besides, different from DCP [17] and PRNet [21], to let pointwise features further contain discriminative structural information, the hierarchical features in different receptive fields are linked to combine structural information in different graphs. Specifically, we perform channel concatenation to let pointwise features more discriminative. After aggregating local structural information, the generated feature $\mathbf{x}_i^l$ is further linked from hierarchical features. Therefore, the final pointwise feature $\mathbf{x'}_i^l$ of each layer is the concatenation of hierarchical features $(\mathbf{x}_i^l, \mathbf{x}_i^{l-1}, ..., \mathbf{x}_i^0)$ at current and previous layers. This feature concatenation operation is derived in DenseNet [52]. The densely-connected EdgeConv layers improve the flow of structural information and avoid the vanishing gradient problem in neural networks.

Given pointwise features extracted by multiple densely-connected EdgeConv layers, two Transformer layers [17], [21], [23] are employed to further extract contextual structural information of the obtained pointwise features. The Transformer model incorporates encoder and decoder modules, which are based on a multi-head attention mechanism. In more detail, the encoder module can be viewed as a fully connected graph, while the decoder module can be regarded as a bipartite graph. Therefore, they can be used to capture long-term structural dependencies in the input point clouds. In our framework, the contextual pointwise features generated by the first Transformer layer facilitate the overlap prediction module to distinguish the points in overlap from the points in non-overlap. The second Transformer layer further performs feature refinement for overlap region, which captures the long-term dependencies of fine-grained structural information between the predicted points in overlap and the target point cloud to yield partial correspondences.

3.3 Overlap Prediction

Given the input point clouds $\mathcal{X}$ and $\mathcal{Y}$, our feature extraction module utilizes the local and global structure of point clouds to yield more discriminative pointwise features. After that, the overlap prediction module is designed for partial registration and used to detect the overlap between $\mathcal{X}$ and $\mathcal{Y}$. Figure 2 shows the detail of our overlap prediction module.

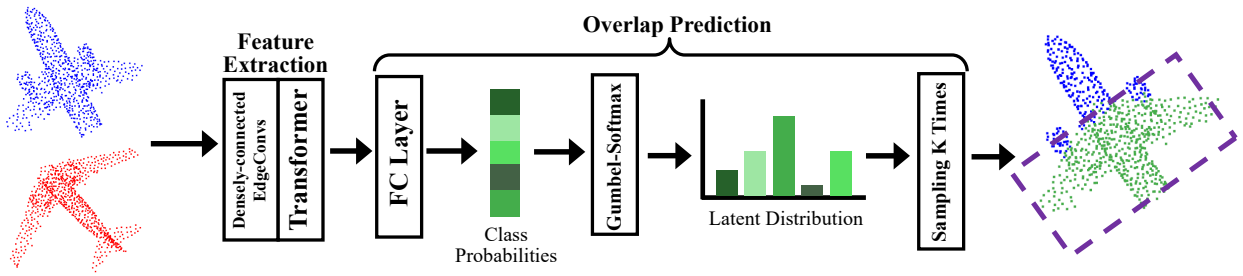


Fig. 2: The detail of overlap prediction in our method. Given pointwise features, our overlap prediction module first predicts a vector $\boldsymbol{\pi}$ denoting class probabilities. Then, Gumbel-Softmax is performed to generate a distribution which describes the latent common structure of the input point clouds. Lastly, we sample K times from the distribution and generate K points representing the overlap.

The overlap prediction module is based on Gumbel-Softmax [51], a differentiable sampling mechanism. Given the Gumbel noise $g_1 ... g_m$ samples drawn from $Gumbel\,(0, 1)$ and class probabilities $\pi_1, \pi_2, \cdots \pi_m$, we can compute a categorical sample z as:

$$z = one\_hot\left(\arg\max_j \left[\left(g_j + \log \pi_j\right)/\tau\right]\right) \tag{2}$$

We approximately replace the $\arg\max$ with the softmax function, and hence Gumbel-Softmax is differentiable. In addition, the temperature τ controls the sharpness of the categorical distribution.

With the defined Gumbel-Softmax distribution, Gumbel Subset Sampling (GSS) [48] is proposed to select the representative set of the point cloud. Given the input pointwise features $\Phi_{\mathcal{X}} \in \mathbb{R}^{M\times D}$, where M represents the number of points and D denotes the embedding dimension, the subset of K pointwise features $\Phi^K_{\mathcal{X}} \in \mathbb{R}^{K\times D}$ can be computed as follows:

$$\Phi^K_{\mathcal{X}} = gumbel\_softmax(W \cdot \Phi^T_{\mathcal{X}}) \cdot \Phi_{\mathcal{X}}, \quad W \in \mathbb{R}^{K\times D}, \tag{3}$$

where W is a learnable weight matrix and can be implemented with an MLP layer in practice, and $\cdot$ denotes matrix multiplication.

It is noted that the permutation of the input points is arbitrary. The proposed module cannot rely on the dependencies between points, and should depend on the permutation invariance of points. However, we observe that GSS is not permutation-invariant. From Eq. 3, we can observe that the selection of subset depends on the learnable weight matrix W. Note that W consists of K different D-dimensional vectors corresponding to K points in the generated subset. The result of multiplying W and $\Phi_{\mathcal{X}}$ can be regarded as the generation of K different sets of class probabilities. Therefore, the generation of subset relies on the permutation of D-dimensional vectors in W. Namely, there are dependencies between points in the generated subset. GSS is not permutation-invariant.

To address this problem, we propose a permutation-invariant overlap prediction module with differentiable sampling to detect the common points between $\mathcal{X}$ and $\mathcal{Y}$. This module can utilize the latent common structure of the input point clouds. Given pointwise features output by the feature extraction module, our proposed overlap prediction module first utilizes an MLP to predict class probabilities $\boldsymbol{\pi} = \{\pi_1, \cdots \pi_m\}$:

$$\boldsymbol{\pi} = W^T \Phi^T_{\mathcal{X}}, \tag{4}$$

where $W \in \mathbb{R}^{D\times 1}$ denotes weights of an MLP layer, and $\boldsymbol{\pi} \in \mathbb{R}^{1\times M}$ describes the probability of each point in $\mathcal{X}$ to be selected as a common point.

Then, we perform Gumbel-Softmax to generate a distribution indicating the latent overlap structure of the input point clouds and draw samples. The sampling point $\boldsymbol{X}_i$. and its corresponding pointwise feature $\Phi^i_{\mathcal{X}}$ can be obtained as follows:

$$\boldsymbol{X}_i = gumbel\_softmax(\boldsymbol{\pi}) \cdot \boldsymbol{X}, \tag{5}$$

$$\Phi^i_{\mathcal{X}} = gumbel\_softmax(\boldsymbol{\pi}) \cdot \Phi_{\mathcal{X}}, \tag{6}$$

where $\boldsymbol{X} \in \mathbb{R}^{M\times 3}$ is a matrix that represents the points in $\mathcal{X}$. In addition, the temperature τ in Gumbel-Softmax is used to adjust the sharpness of the structural information distribution, which is predicted as follows:

$$\tau = mean(W^T \Phi^T_{\mathcal{X}}), \tag{7}$$

where the mean operation denotes averaging the vector to obtain a scalar.

Finally, we sample K times from the distribution. Note that each sample is independent. These K samples indicate the overlap between $\mathcal{X}$ and $\mathcal{Y}$. Based on the above, we can obtain the common points denoted by $\boldsymbol{X}_K \in \mathbb{R}^{K\times 3}$ and their corresponding pointwise feature $\Phi^K_{\mathcal{X}}$.

3.4 Pose Estimation

After overlap detection, we let the generated common points $\boldsymbol{X}_K$ directly find their corresponding points in $\mathcal{Y}$ to yield partial correspondences. Compared with the method in PRNet [21], which detects the keypoints from the input point clouds separately, our method reduces the chance of wrong correspondence generation. We then formulate the partial correspondences as a matrix computed by pointwise feature similarity via dot-product. In fact, due to the noise and outliers point clouds, there are no exact point-to-point correspondences for registration. Therefore, the softmax operation is used here to perform soft point matching. Given the selected pointwise features $\Phi^K_{\mathcal{X}}$ and features $\Phi_{\mathcal{Y}} \in \mathbb{R}^{N\times D}$ extracted from $\mathcal{Y}$, we can obtain the virtual corresponding points $\boldsymbol{Y}_K \in \mathbb{R}^{K\times 3}$:

$$\boldsymbol{Y}_K = softmax(\Phi^K_{\mathcal{X}} \Phi^T_{\mathcal{Y}},) \cdot \boldsymbol{Y}, \tag{8}$$

where $\boldsymbol{Y} \in \mathbb{R}^{N\times 3}$ is a matrix representing the virtual points in $\mathcal{Y}$, and the class probabilities $\pi_1, \pi_2, \cdots \pi_k$ in Gumbel-Softmax denote the pointwise feature similarity.

Given the corresponding point pairs $\boldsymbol{X}_K$ and $\boldsymbol{Y}_K$, we use the weighted least squares method and the point cloud registration problem can be formulated as follows:

$$\underset{\mathbf{w},\mathbf{R},\mathbf{t}}{\arg\min} \sum_{k=1}^{K} w_k(\| \mathbf{R}x_k + \mathbf{t} - y_k \|_2^2), \tag{9}$$

where x_k and y_k are corresponding points in overlap. Here $\mathbf{w} \in \mathbb{R}^K$ is a weight representing the contributions of correspondences in the rigid motion computation. Assigning different weights to point pairs can further weaken the negative impact of wrong correspondences. The assigned weight takes the maximum value of each row of the affinity matrix generated by pointwise feature similarity. Given correspondences and their corresponding weights, the optimal rigid motion can be obtained by SVD.

3.5 The Loss Function of Registration Task

We argue that the key to the performance of point cloud registration is correspondence prediction. Therefore, the two loss functions are employed to directly constrain the overlap prediction and the correspondence generation in our registration method. For overlap prediction, the binary cross entropy loss is adopted as the supervision for learning to sample the points in overlap, and is formulated as follows:

$$L_{op} = BCE(\boldsymbol{\pi}, \boldsymbol{\pi}^*), \tag{10}$$

where $\boldsymbol{\pi}$ describes the probability of each point in $\mathcal{X}$ to be selected as a common point, and $\boldsymbol{\pi}^*$ is the ground truth of the probability of points in overlap. The $\boldsymbol{\pi}^*$ can be obtained by finding the corresponding point pairs between the $\mathcal{X}^{'}$ transformed by the ground truth of the rigid motion and $\mathcal{Y}$. For correspondence generation, we employ the standard cross-entropy loss as the supervision for learning to yield corresponding point pairs. The affinity matrix M representing correspondences predicted by our method can be computed as follows:

$$M = softmax(\Phi_{\mathcal{X}}^K \Phi_{\mathcal{Y}}^T,) \tag{11}$$

The ground truth of affinity matrix M^* can be pre-computed in the same way as $\boldsymbol{\pi}^*$. Based on the above definition, the cross-entropy loss is formulated as follows:

$$L_{corr} = -\frac{\sum(M^* log M)}{\sum M^*} \tag{12}$$

Then, we can calculate the total loss L_p at the p-th iteration as:

$$L_p = L_{corr} + \lambda L_{op}, \tag{13}$$

where λ is a hyper-parameter. Our total loss function is $L = \sum_p^P L_p$, because our registration method predicts the final rigid motion in a coarse-to-fine way. Our method has already obtained satisfactory results for partial registration when the number of iterations of our method is set as 2.

The loss function L_{corr} directly constrains partial correspondence generation. Having obtained corresponding point pairs, the predicted rigid motion $[\boldsymbol{R_{pred}}, \mathbf{t_{pred}}]$ can be recovered using SVD [5]. We align $\mathcal{X}$ to $\mathcal{Y}$, and the transformed points $\boldsymbol{X}' \in \mathbb{R}^{3\times M}$ can be calculated as:

$$\boldsymbol{X}' = \boldsymbol{R}_{pred} \cdot \boldsymbol{X}^T + \mathbf{t}_{pred} \tag{14}$$

3.6 Implementation Details

In our framework, we employ k-NN to construct a graph, and the number of neighbor points is set as 20 similar to DGCNN [37]. 5 densely-connected EdgeConv layers are used here. After that, we follow DCP [17] to deploy Transformer [23] layers to extract contextual structural information. The dimension of pointwise features output by the Transformer is fixed to 512. In addition, the number of sampling points is set based on the point cloud overlap rate and the total number of iterations is selected as 2. The λ in loss function is set as 0.5. We train our model on two GTX 2080Ti GPUs for 150 epochs. Adam [53] is employed as the optimizer and the initial learning rate is set as 0.0001.

4 EXPERIMENTS AND DISCUSSION

In this section, we first introduce the experimental settings and the datasets for point cloud registration. After that, we compare STORM to other methods on point clouds with low overlaps on the ModelNet40 [24] dataset. Then, we conduct ablation study to evaluate the effectiveness of the modules in STORM. Lastly, we compare STORM to other methods on the outdoor KITTI [25] dataset.

4.1 Experimental Settings

We conducted partial point cloud registration experiments on ModelNet40 dataset, which consists of 12,311 CAD models in 40 categories. The number of the point cloud for each model is sampled as 1024. We follow [17], [21], [31] and train these learning-based methods on the training set and test them on the test set of ModelNet40 dataset, while non-learning registration methods are directly tested on the test set. For partial registration, different preprocessed strategies are used to truncate the point clouds in the ModelNet40 dataset to simulate partial scans and generate the input point clouds $\mathcal{X}$ to $\mathcal{Y}$ with different overlaps. More specifically, this preprocessed strategy randomly selects the fastest point pairs in the raw point clouds, and then finds their s nearest neighbors, respectively. The value of s determines the overlaps of $\mathcal{X}$ and $\mathcal{Y}$, and the point cloud overlap rate is defined as the number of point pairs which are at a similar spatial location divided by the number of all points. In this way, s can be set as 768, 700, 640, 600 and 560 to generate point clouds with approximate overlap rates of 0.69, 0.58, 0.47, 0.40 and 0.32, respectively. The rigid transformations are provided as supervised information. Along each axis, the rotation angle is sampled in $[0, 45^\circ]$. The translation is randomly selected in [-0.5, 0.5]. We apply the ground truth of transformation to $\mathcal{Y}$. The learning-based methods for comparison are trained to predict a rigid motion which aligns $\mathcal{X}$ to $\mathcal{Y}$. The following evaluation metrics are used for comparison: mean squared error (MSE), root mean squared error (RMSE), and mean absolute error (MAE), which have been widely used in point cloud registration. The $MSE(\mathbf{R})$, $RMSE(\mathbf{R})$, and $MAE(\mathbf{R})$ in degrees are used for rotation

measurement, while the $MSE(\mathbf{t})$, $RMSE(\mathbf{t})$, and $MAE(\mathbf{t})$ are used for translation.

In addition, we compare our model to other registration methods on the outdoor KITTI [25] dataset, we use the sequences 0 to 5 for training, 6 to 7 for validation, and 8 to 10 for testing. For all LIDAR scans, we use the first scan that is taken at least 5m apart within each sequence to create the input point clouds for registration. This procedure generates 2615 pairs for training, 349 pairs for validation, and 1068 pairs for testing. To make point cloud registration more challenging, we truncate approximately 30% points from the scans to generate point clouds with low overlap, because the pairs of LIDAR scans have a large overlap. Besides, we follow [54] to apply rigid transformations to the target point clouds $\mathcal{Y}$. During training and testing, the yaw angle is sampled in $[-45°, 45°]$, and the other two angles are sampled in $[-15°, 15°]$. For translation, the x and z translations are randomly selected in $[-5m, 5m]$. The following evaluation metrics are used for comparison: $RMSE(\mathbf{R})$, $RMSE(\mathbf{t})$, average rotation error (RE) and translation error (TE) defined in [14], recall, and inference time. The recall is the ratio of successfully aligned samples, and the successful registration is defined if RE is smaller than $5°$ and TE is smaller than $1.5m$. The inference time is measured in seconds.

To evaluate the performance of our proposed method, the following methods are selected for comparison:

1) ICP [1]. ICP is a baseline method of point cloud registration. It alternates between establishing point-to-point correspondences based on spatial distance and computing the rigid motion based on the generated correspondences by SVD [5].
2) Trimmed ICP [26]. Trimmed ICP (TrICP) introduces the least trimmed squares (LTS) into the ICP algorithm to tackle the partial overlap problem.
3) PointNetLK [15]. PointNetLK employs a differentiable Lucas & Kanade algorithm combined with PointNet [16] to compute a rigid motion.
4) DCP [17]. DCP learns pointwise features based on local or global structural information, and generates virtual correspondences utilizing a probabilistic approach.
5) PRNet [21]. PRNet iteratively aligns the partial point clouds together. It establishes keypoint-to-keypoint correspondences. The keypoint detection is based on the L2 distance between feature vectors.
6) IDAM [31]. IDAM is a distance-aware registration method. It generates pointwise features based on both geometric and spatial distance, and employs a two-step point elimination technique to tackle the partial overlap problem.
7) RPM-Net [19]. RPM-Net employs 4D point pair features (PPF) and a Sinkhorn layer to generate correspondences. It is robust to noise and outliers.
8) PREDATOR [11] PREDATOR is a 3D feature extraction method with an overlap-attention block to detect the points lain in overlap.

For ICP, we use its implementation in Open3D [55]. TrICP's implementation in Point Cloud Library (PCL) [56] is used for comparison. For learning-based methods, the implementations released by the authors are used. We use Graph Neural Network (GNN) as their feature extraction module in IDAM [31]. For RPM-Net, we employ the normal estimation method in Open3D to compute point cloud normals. PREDATOR is only a feature extraction method, so we employ random sample consensus (RANSAC) [12] algorithm to prune candidate correspondences which are generated based on the similarity of learning-based features and predict a rigid motion aligning the input point clouds. PREDATOR and RANSAC are called *PRE-RANSAC*. For DCP, the full model with Transformer in DCP [17] is used in our experiments. Because all registration methods are iterative except for DCP, we iterate DCP three times to explore the impact of iteration on registration performance. We call iterative DCP *DCP (iter 3)*.

4.2 Partial Object Registration

In this subsection, we conduct partial registration experiments on the ModelNet40 dataset. To fully investigate the performance of the proposed method, we have conducted experiments under the following settings:

1) Clean point clouds with approximately 0.69 overlap rate.
2) Point clouds with noise and approximately 0.69 overlap ratio.
3) Point clouds with noise, outliers, and approximately 0.69 overlap ratio.
4) Point clouds with noise and approximate overlap ratios of 0.69, 0.58, 0.47, 0.40, and 0.32, respectively.
5) The partial point clouds are generated by the partial manner in RPM-Net [19] and OMNet [32].

After the quantitative experiments, several registration examples predicted by STORM and other methods are shown. Finally, the registration efficiency is given.

TABLE 1: Experimental results on clean point clouds with approximately 0.69 overlap rate.

Model	$MSE(\boldsymbol{R})\downarrow$	$RMSE(\boldsymbol{R})\downarrow$	$MAE(\boldsymbol{R})\downarrow$	$MSE(\mathbf{t})\downarrow$	$RMSE(\mathbf{t})\downarrow$	$MAE(\mathbf{t})\downarrow$
ICP	390.06	19.75	10.67	0.029	0.17	0.13
TrICP	406.02	20.15	11.59	0.020	0.14	0.092
PointNetLK	181.98	13.49	7.03	0.032	0.18	0.12
DCP	63.04	7.94	5.70	0.0050	0.071	0.058
DCP (iter 3)	34.57	5.88	3.40	0.0050	0.071	0.051
PRNet	51.98	7.21	4.42	0.0059	0.077	0.055
IDAM	136.89	11.70	6.84	0.020	0.14	0.090
RPM-Net	4.62	2.15	0.84	0.00014	0.012	0.0058
PRE-RANSAC	1.30	1.14	0.37	0.00012	0.011	0.0023
STORM	0.10	0.32	0.092	0.0000014	0.0012	0.00034

Clean data. We first conduct experiments on the clean point clouds with approximately 0.69 overlap rate. From the results in Table 1, we have the following observations:

1) The non-learning methods like ICP and TrICP, fail to align the point clouds.
2) DCP, PRNet, and IDAM cannot obtain satisfactory results on point clouds with low overlap.
3) DCP (iter 3) obtains better performance than DCP and PRNet.
4) Our model achieves the state-of-the-art performance and obtains lower by 0.82 in $RMSE(\mathbf{R})$ compared to PRE-RANSAC.

The better performance of STORM can be due to the following two reasons. First, the feature extraction and refinement modules incorporating graph-based methods in our method

can capture long-term dependencies in structural information to generate discriminative pointwise features. Then, our proposed overlap prediction module can detect the point cloud overlap utilizing learned contextual information, and facilitate exact correspondence generation. In addition, it is noted that ICP and TrICP cannot align the input point clouds together for partial point cloud registration without good initial transformations. The poor performance is due to their simple point-to-point correspondence estimation based on spatial distance. PRNet, DCP and IDAM perform better than PointNetLK through the utilization of structural information, but their performance is inferior to RPM-Net's, PRE-RANSAC's, and ours. Compared to DCP, the better performance of DCP (iter 3) demonstrates that the methods in an iteration manner can help improve registration performance. However, compared to RPM-Net, PRE-RANSAC and STORM, DCP (iter 3) still cannot obtain comparable performance due to the lack of outlier rejection modules in its registration pipeline.

TABLE 2: Experimental results on point clouds with noise and approximately 0.69 overlap ratio.

Model	$MSE(\boldsymbol{R})\downarrow$	$RMSE(\boldsymbol{R})\downarrow$	$MAE(\boldsymbol{R})\downarrow$	$MSE(\mathbf{t})\downarrow$	$RMSE(\mathbf{t})\downarrow$	$MAE(\mathbf{t})\downarrow$
ICP	380.64	19.51	10.72	0.029	0.17	0.14
TrICP	792.99	28.16	24.11	0.078	0.28	0.24
PointNetLK	192.38	13.87	7.23	0.032	0.18	0.12
DCP	71.06	8.43	6.02	0.0044	0.066	0.051
DCP (iter 3)	37.58	6.13	3.71	0.0056	0.075	0.054
PRNet	53.29	7.30	4.52	0.0069	0.083	0.061
IDAM	104.86	10.24	6.87	0.023	0.15	0.11
RPM-Net	6.55	2.56	1.15	0.00026	0.016	0.0092
PRE-RANSAC	1.77	1.33	0.82	0.00032	0.018	0.0044
STORM	1.69	1.30	0.62	0.000050	0.0071	0.00033

Data with Gaussian Noise. Then, we follow [17], [19], [21], [31] to add Gaussian noise sampled from $\mathcal{N}(0, 0.01)$ and clipped to $[-0.05, 0.05]$ to point clouds. As shown in Table 2, we have the following observations:

1) The performance of PRNet and IDAM is not affected by noise, but they cannot obtain satisfactory registration results.
2) The performance of TrICP is worse than the result of vanilla ICP.
3) Compared to other registration methods, our model still achieves the state-of-the-art performance in the presence of noise.

Note that the performance of TrICP is worse than the result of vanilla ICP. It can be dedicated to the following reasons. First, vanilla ICP and TrICP are implemented in Open3D [55] and PCL [56], respectively. The hyper-parameter settings and implementation details of them are different. Second, TrICP follows the iterative pipeline of ICP and is also a fine registration method. Therefore, TrICP cannot converge on the input point clouds with a large relative pose. Besides, data with additional noise makes the optimization of the trimmed MSE function less robust.

Data with Gaussian Noise and Outliers. We further add 20% of the total points as outliers at random locations in 3D space to the point clouds with noise. The experimental results are shown in Table 3. We have the following observations:

1) The performance of most registration methods degrades on point clouds with noise and outliers.
2) PointNetLK fails to converge, and performs worse than non-learning methods like ICP and TrICP.

TABLE 3: Experimental results on point clouds with noise and outliers. The point cloud overlap rate is approximately 0.69.

Model	$MSE(\boldsymbol{R})\downarrow$	$RMSE(\boldsymbol{R})\downarrow$	$MAE(\boldsymbol{R})\downarrow$	$MSE(\mathbf{t})\downarrow$	$RMSE(\mathbf{t})\downarrow$	$MAE(\mathbf{t})\downarrow$
ICP	590.49	24.30	18.10	0.014	0.12	0.081
TrICP	791.30	28.13	24.12	0.078	0.28	0.24
PointNetLK	1746.40	41.79	32.94	0.12	0.34	0.27
DCP	277.89	16.67	13.66	0.0077	0.088	0.070
DCP (iter 3)	124.55	11.16	8.07	0.020	0.14	0.11
PRNet	207.65	14.41	10.73	0.012	0.11	0.089
IDAM	234.09	15.30	11.28	0.023	0.15	0.12
RPM-Net	34.46	5.87	3.74	0.0052	0.072	0.052
PRE-RANSAC	31.92	5.65	1.38	0.0012	0.035	0.0087
STORM	6.97	2.64	1.50	0.00020	0.014	0.0080

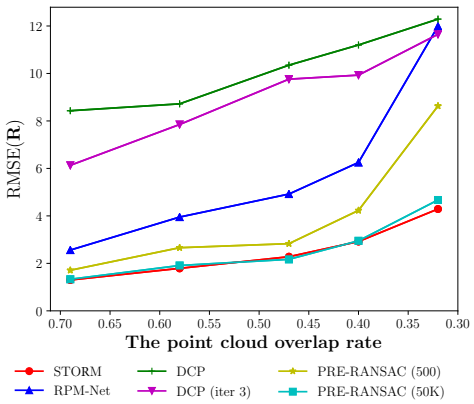


Fig. 3: The experimental results of compared methods on point clouds with noise and lower overlaps in terms of $RMSE(\boldsymbol{R})$.

3) Only RPM-Net, PRE-RANSAC, and STORM obtain satisfactory results. STORM obtains lower by 3 in $RMSE(\mathbf{R})$ compared to PRE-RANSAC.

The experimental results demonstrate that our method is robust to outliers and noise. The robustness of our method is due to virtual partial correspondence generation based on structural information which is not easily affected by noise and outliers. It is noted that PRE-RANSAC also achieves comparable performance due to the proposed overlap-attention block.

Lower Overlap. To explore the impact of point clouds with lower overlaps on registration methods, we further perform partial-to-partial registration experiments on the point clouds with noise and lower overlaps. In more detail, in addition to point clouds with approximately 0.69 overlap rate, we use strategies mentioned in experimental settings to generate partial point clouds with noise and approximately 0.58, 0.47, 0.40 and 0.32 overlap rates, respectively. We train and test the learning-based methods on point clouds with different overlaps. The $RMSE(\mathbf{R})$ and $RMSE(\mathbf{t})$ are used to quantitatively evaluate the performance of registration methods. During the experiments, we found that PointNetLK, PRNet and IDAM fail to converge. Besides, non-learning methods like ICP and TrICP stall in suboptimal local minima and have a poor performance. Therefore, we compare STORM to RPMNet, DCP, DCP (iter 3) and PRE-RANSAC on point clouds with lower overlaps. To explore the importance of RANSAC to Predator, Predator is combined with RANSAC of different iterations (500 and 50,000). The experimental results are shown in Figure 3 and 4. We

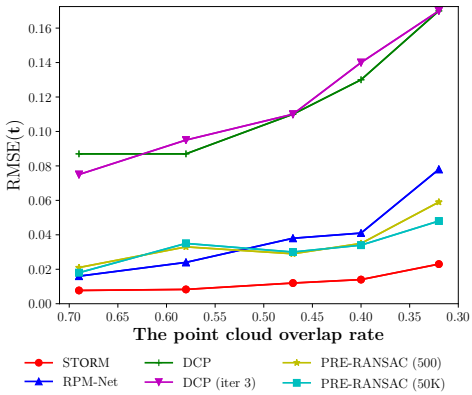


Fig. 4: The experimental results of compared methods on point clouds with noise and lower overlaps in terms of $RMSE(\mathbf{t})$.

have the following observations:

1) When the point cloud overlap rate drops to 0.58, 0.47, 0.40 and 0.32, STORM and PRE-RANSAC (50K) outperform other registration methods a lot and achieve similar performance in $RMSE(\mathbf{R})$.
2) PRE-RANSAC (50K) performs better than PRE-RANSAC (500) especially on point clouds with approximately 0.47 or 0.32 overlap rate.
3) Compared to PRE-RANSAC (50K), STORM obtains better performance in terms of $RMSE(\mathbf{t})$.

The satisfactory registration results of STORM on point clouds with lower overlaps demonstrate that our proposed overlap prediction module utilizes structural information to effectively detect the point cloud overlap and generate partial correspondences. Besides, RANSAC of more iterations can help PREDATOR obtain more comparable performance.

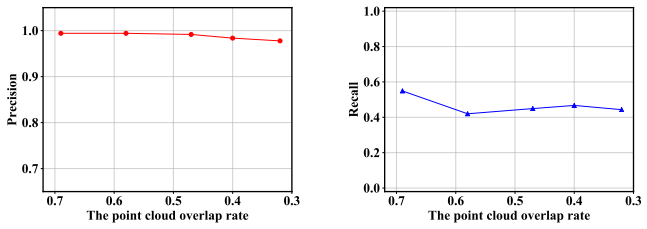
TABLE 4: Experimental results on partial point clouds truncated by the partial manner in RPM-Net. The point cloud overlap rate is approximately 0.70.

Model	$MSE(\boldsymbol{R})\downarrow$	$RMSE(\boldsymbol{R})\downarrow$	$MAE(\boldsymbol{R})\downarrow$	$MSE(\mathbf{t})\downarrow$	$RMSE(\mathbf{t})\downarrow$	$MAE(\mathbf{t})\downarrow$
ICP	415.63	20.40	12.65	0.036	0.19	0.12
TrICP	417.72	20.44	12.37	0.019	0.14	0.097
PointNetLK	778.58	27.90	18.67	0.064	0.25	0.18
DCP	179.21	13.39	9.97	0.0058	0.076	0.057
DCP (iter 3)	63.65	7.98	4.85	0.0072	0.085	0.063
PRNet	83.58	9.14	5.60	0.0094	0.097	0.071
IDAM	152.77	12.36	7.04	0.0086	0.093	0.054
RPM-Net	15.44	3.93	1.39	0.0019	0.044	0.015
PRE-RANSAC	13.67	3.70	0.71	0.0015	0.038	0.0043
OMNet	13.83	3.72	1.31	0.0015	0.039	0.015
STORM	4.97	2.23	0.54	0.00017	0.013	0.0032

Different Partial Manners. In this experiment, we use the partial manner in RPM-Net [19] to generate partial clean point clouds. The division manner in OMNet [32] is used to split the ModelNet40 dataset into training, validation, and test sets. We retrain DCP (iter 3), PRNet, IDAM, PREDATOR, and STORM on the training set, and compare their evaluation results to the experimental results of ICP, PointNetLK, DCP, RPM-Net, and OMNet recorded in [32]. The evaluation results are shown in Table 4. We have the following observations:

1) Compared to other registration methods, OMNet achieves the third-best performance. PRE-RANSAC and STORM outperform it.
2) STORM still achieves satisfactory performance on partial point clouds generated by the partial manner in RPM-Net and OMNet.

OMNet achieves comparable performance due to the exact mask prediction for overlap and effective utilization of global features containing the geometric information of all regions. Besides, the satisfactory performance of STORM demonstrates the generalization of learned structural features to point clouds in unseen categories.



(a) (b)

Fig. 5: The precision and recall of the learned points in overlap region on point clouds with different overlap ratios.

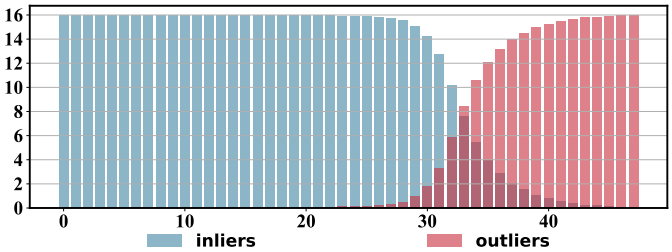


Fig. 6: The distribution of inliers and outliers in the points ranking by the predicted probabilities based on structural information.

Overlap Prediction. STORM proposes an overlap prediction module with differentiable sampling to avoid the negative impact of points in non-overlap on registration performance. To validate the effectiveness of the proposed overlap prediction module based on structural information, we evaluate the precision and recall of the learned points in the overlap region on point clouds with different overlap ratios. The quantitative results are shown in Figure 5. As shown in Figure 5, the accuracy of our overlap prediction is approximately 100%, while the recall is approximately 50%. The results demonstrate that although the proposed overlap prediction module cannot yield all points fallen in overlap, it can effectively avoid the negative impact of irrelevant points on partial correspondence generation. Moreover, the overlap prediction module detects the overlap based on the probabilities of points in overlap. The probabilities are learned by contextual structural information of the input point clouds. To further validate that our overlap prediction module utilizes structural information, we rank the points based on the probabilities and count the number of inliers and outliers defined by the ground truth of relative poses. Specifically, the ranked points are divided into 48 intervals, and the number of inliers and outliers in each interval is counted, respectively. The distribution of inliers and outliers is shown in Figure 6. Note that this experiment is conducted on the point clouds with approximately 0.69 overlap ratio.

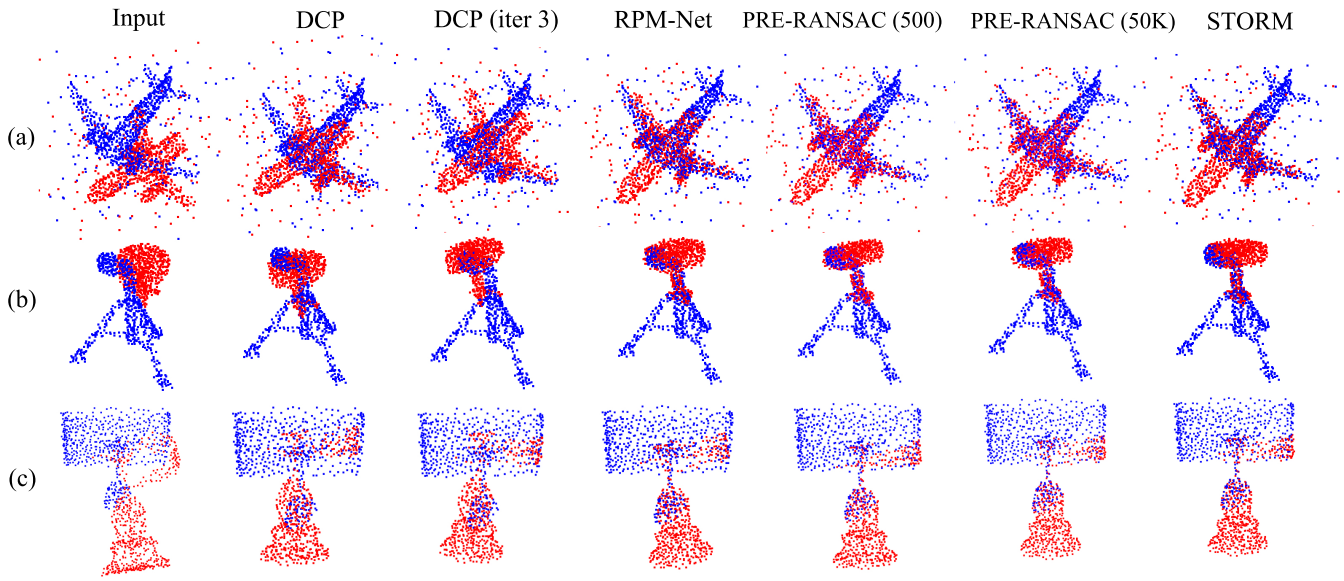


Fig. 7: Qualitative registration examples. (a) Point clouds with approximately 0.69 overlap rate, noise and outliers. (b) Point clouds with approximately 0.47 overlap rate and noise. (c) Point clouds with approximately 0.32 overlap rate and noise.

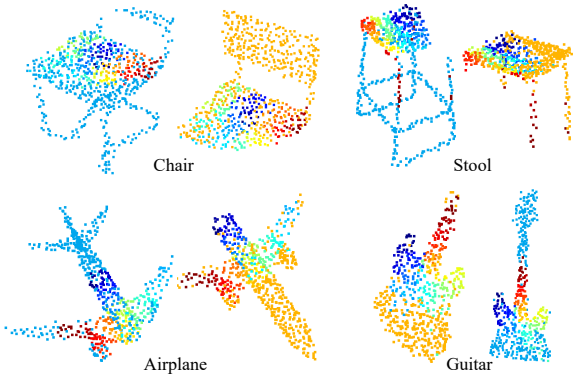


Fig. 8: The visualization of the relevance of the learned point embeddings using t-SNE [57].

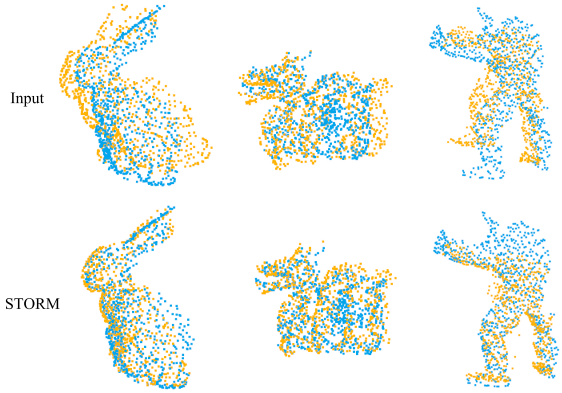


Fig. 9: Qualitative registration examples on the Stanford 3D Scanning Repository [58]. The first row is the input point clouds, and the second row is the point clouds aligned by STORM.

Figure 6 shows the large difference in the distribution of inliers and outliers. The different distribution indicates that our overlap prediction module with differentiable sampling can effectively obtain inliers based on the learned contextual structural information of the input point clouds.

Visualization. Figure 7 shows several examples of registered point clouds with noise, outliers and low overlap from the ModelNet40 dataset. For partial-to-partial registration, STORM obtains a more desirable visualization of registered point clouds with noise and outliers compared to DCP, RPM-Net and PRED-RANSAC. In addition, to validate the effect of learned pointwise features containing structural information on registration performance, we follow [6] to visualize the relevance of the pointwise features. Specifically, we employ t-SNE [57] to map the pointwise features after overlap prediction to a scalar space and colorize the corresponding points with the spectral color map based on the yielded scalars. The visualization of the relevance of the learned pointwise features is shown in Figure 8. The satisfactory visualization demonstrates that STORM can utilize structural information to generate the pointwise features facilitating point cloud registration. Moreover, to verify the generalization ability of STORM, different from [21], we directly test STORM on the Stanford 3D dataset [58] using the model trained on the ModelNet40 dataset rather than fine-tune the trained model. The Stanford 3D dataset includes 3D partial scan meshes, and we sample 768 points on these 3D meshes separately to generate point clouds. Figure 9 shows several registration examples. As shown in Figure 9, STORM can still achieve satisfactory performance on partial scans on an unseen dataset.

Registration Efficiency. To evaluate the computational efficiency of these methods, we have recorded the average inference time on the test set, and the testing bed is an Nvidia GTX 2080Ti GPU and a 2.10GHz Intel(R) Xeon(R) Silver 4110 CPU. The inference time comparison is demonstrated in Table 5. As shown in Table 5, ICP and TrICP are faster

TABLE 5: The inference time comparison of different methods (in microseconds). The number of input points is 768.

Model	Inference Time (ms)
ICP	11
TrICP	17
PointNetLK	127
DCP	21
DCP (iter 3)	63
PRNet	106
IDAM	32
RPM-Net	100
PRE-RANSAC	95
STORM	69

TABLE 6: The computation cost of STORM for different sizes of the input point clouds. The inference time is measured in milliseconds (ms).

	The size of input point clouds						
	256	512	768	1024	1280	1536	2048
Time (ms)	63.97	66.57	69.12	70.18	71.24	79.70	93.29
GFLOPs	15.15	24.94	34.73	44.51	54.30	63.86	83.66

than learning-based methods, because the backends of them are highly optimized and paralleled in Open3D [55] and PCL [56], respectively. Note that DCP is the fastest learning-based registration method, because it predicts a rigid motion in a one-shot manner, while other registration methods are all iterative. Jointly considering the registration performance and the inference time of models, STORM achieves the trade-off between the efficiency and the accuracy of registration. Furthermore, we provide the memory and computation cost of STORM for different sizes of the input point clouds. As shown in Table 6, as the number of input points increases, the inference time and GFLOPs of STORM do not increase a lot. Specifically, the inference time of STORM on 2048 points is only approximately 1.5 times to that on 256 points. Besides, the computation cost of STORM increases by 10 GFLOPs for every 256 points increase. The results demonstrate that STORM is still efficient on relatively large scale of points.

4.3 Ablation Study

TABLE 7: Results of ablation study. DGCNN denotes that we replace densely-connected EdgeConv layers in our framework with plain EdgeConv layers used in DCP [17] and PRNet [21]. $Top-K$ denotes that we use PRNet's method to perform keypoint detection. No sampling denotes that we remove the proposed overlap prediction module from our framework. (The point cloud overlap rate is approximately 0.69.)

Model	$RMSE(\boldsymbol{R})\downarrow$	$MAE(\boldsymbol{R})\downarrow$	$RMSE(\mathbf{t})\downarrow$	$MAE(\mathbf{t})\downarrow$
DGCNN	1.92	0.98	0.010	0.0058
$Top-K$	5.33	3.43	0.050	0.037
No sampling	4.24	2.72	0.046	0.037
STORM	1.30	0.62	0.0071	0.0033

In this section, we conduct the ablation study to analyze the effectiveness of our registration pipeline. The ablation study is performed on point clouds with noise and approximately 0.69 overlap rate. More specifically, four sub-experiments are conducted to evaluate effectiveness of the feature extraction method, our proposed overlap prediction module, the number of sampling points in the overlap prediction module, and the loss function of our method, respectively. RMSE and MAE are used as evaluation metrics.

First, in the feature extraction step, we replace the densely-connected EdgeConv layers [22] by plain EdgeConv layers [17], [37] and results are shown in Table 7. The results demonstrate that the utilization of the densely-connected EdgeConv layers is more effective. We argue that the operation of feature concatenation is beneficial to yield discriminative pointwise features for exact correspondence generation.

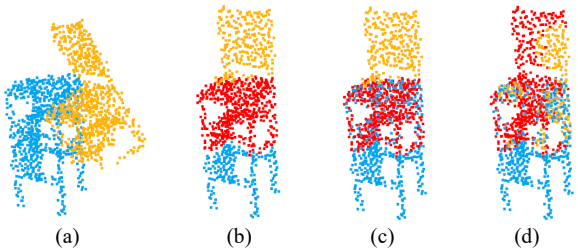


Fig. 10: The red points denote that the points lay in overlap. (a) The input point clouds. (b) The ground truth of overlap. (c) The overlap predicted by the model with the proposed overlap prediction module. (d) The overlap predicted by the model with the $Top-K$ mechanism.

Then, to verify the effectiveness of our proposed overlap prediction module on registration performance, the following variants are compared with our origin model: 1) Replacing our proposed overlap prediction module with differentiable sampling by $Top-K$ operation employed in PRNet [21]. 2) Removing the overlap prediction module from our registration method. The experimental results are also shown in Table 7. It is noted that the model with the proposed overlap prediction module obtains lower by 3.91 and 2.82 in $RMSE(\boldsymbol{R})$ compared to the model with the $Top-K$ mechanism and the model without overlap prediction. The results demonstrate that the $Top-K$ mechanism has a negative impact on registration performance, while our proposed overlap prediction module facilitates partial correspondence generation to align partial point clouds well. To intuitively demonstrate the effect of our proposed module, the qualitative predicted overlap is shown in Figure 10. Note that we draw the red points denoting the overlap region on a complete object aligned by the ground truth of a rigid motion to make the visualization of overlap clearer. We have the following observations:

1) The points predicted by our proposed overlap prediction module with differentiable sampling mostly lie in the point cloud overlap.
2) The keypoint detection using a $Top-K$ mechanism based on the L2-norm of features cannot detect the overlap.

The qualitative results indicate that our proposed overlap prediction module draws samples from the latent distribution describing the common structure of the input point clouds, and thus the overlap can be exactly detected, while $Top-K$ operation based on the L2-norm of features cannot reflect the common points of the input point clouds.

Equipped with our proposed overlap prediction module, the partial correspondences can be further established based on feature similarity. The visualization of partial correspondences predicted by STORM is shown in Figure 11. To make the visualization of the predicted correspondences more intuitively, we apply the predicted rigid motion to the source point cloud $\mathcal{X}$, and translate the transformed point cloud $\mathcal{X}'$ a short distance. We use the lines in a purple color denoting the correspondences between the sampling points in $\mathcal{X}$ and their virtual corresponding points in $\mathcal{Y}$. As shown in Figure 11, our method with overlap prediction can generate the exact correspondences in point cloud overlap.

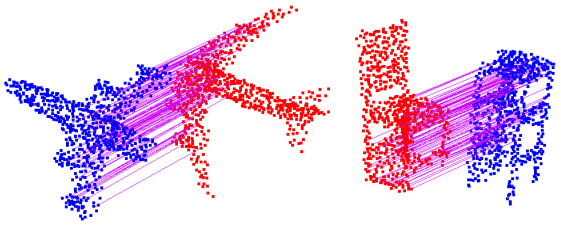


Fig. 11: Qualitative partial correspondence examples predicted by STORM.

TABLE 8: Experimental results with respect to different numbers of sampling points predicted by the proposed overlap prediction module on registration accuracy. The input point clouds are with noise and the overlap rate is approximately 0.69.

# Sampling points	$RMSE(\boldsymbol{R})\downarrow$	$MAE(\boldsymbol{R})\downarrow$	$RMSE(\mathbf{t})\downarrow$	$MAE(\mathbf{t})\downarrow$
256	3.20	2.00	0.022	0.013
384	1.75	0.98	0.0088	0.0054
512	1.30	0.62	0.0071	0.0033

After that, in the overlap prediction module, the experiments with respect to the number of sampling points are conducted. As shown in Table 8, the more sampling points, the lower the registration errors in general. The method with 384 sampling points approximately achieves satisfactory performance as the model with 512 sampling points, while the method with 256 sampling points performs slightly worse.

TABLE 9: Experimental results with respect to the loss function. The input point clouds are with noise and the overlap rate is approximately 0.69.

Loss Function	$RMSE(\boldsymbol{R})\downarrow$	$MAE(\boldsymbol{R})\downarrow$	$RMSE(\mathbf{t})\downarrow$	$MAE(\mathbf{t})\downarrow$
L_{pose}	3.57	2.34	0.025	0.017
L_{corr}	5.24	3.48	0.033	0.021
$L_{pose}+L_{op}$	3.18	2.05	0.021	0.014
$L_{corr}+L_{op}$	1.30	0.62	0.0071	0.0033

Finally, we investigate the effectiveness of the loss function employed in our method. We denote the binary cross entropy loss for overlap detection as L_{op} and the cross-entropy loss for correspondence generation as L_{corr}. Besides, the rigid motion loss function used in [17], [19], [21] is denoted as L_{pose}, which is compared with the loss function in our method, and the quantitative results are shown in Table 9. In general, the method with L_{corr} achieves better performance than the method with L_{pose}. It shows that L_{corr} which directly constrains the correspondence generation facilitates the convergence of model parameters. The results also demonstrate that L_{op} facilitates the exact partial correspondence generation and improves the performance of partial-to-partial registration.

TABLE 10: The quantitative results of registration methods on the outdoor KITTI dataset.

Model	$RMSE(\boldsymbol{R})\downarrow$	$RE\downarrow$	$RMSE(\mathbf{t})\downarrow$	$TE\downarrow$	$Recall\uparrow$	Time (s)
ICP	15.34	22.87	2.28	3.67	1.1%	0.30
TrICP	14.37	20.20	2.35	3.44	10.7%	0.27
FGR	60.80	39.06	2.27	2.95	34.7%	1.95
3DRegNet	101.44	100.78	3.45	5.27	0.3%	0.014
RPMNet	5.40	5.78	1.40	1.80	43.3%	0.14
PRE-RANSAC	0.29	0.44	0.15	0.14	99.7%	1.32
STORM	2.16	2.27	0.64	0.70	88.6%	0.070

4.4 Additional Evaluation on Outdoor Dataset

We compare STORM to other registration methods on the outdoor KITTI [25] dataset. In addition to the methods STORM, RPM-Net, PRE-RANSAC, ICP, and TrICP for partial object registration, the following outlier rejection methods for large scale datasets are for comparison:

1) 3DRegNet [13]. Given correspondences generated by Fast Point Feature Histogram (FPFH) features [12], 3DRegNet employs a classification network to classify the generated matches into inliers or outliers and predict a rigid motion aligning point clouds.
2) FGR [4]. Utilizing the initial correspondences yielded by FPFH, FGR directly optimizes a robust objective function to solve the rigid motions rather than establish correspondences iteratively.

Note that we also try to combine FPFH with DGR [14] which proposes a 6D convolutional network to estimate the probabilities of correct correspondences. However, during training, the loss values of DGR are infinite, and DGR cannot converge.

The input to registration methods is unaligned LIDAR scans downsampled with 30cm voxel size from the original KITTI dataset. As the grounds in LIDAR scans have little geometric information, for STORM and RPM-Net which are end-to-end learning-based registration methods, the points in the grounds are removed from the unaligned point clouds by RANSAC algorithm. Then, the input point clouds to STORM and RPM-Net are uniformly downsampled to 1024 points. Besides, for correspondence classification methods, 3DRegNet and FGR, we employ FPFH to generate putative correspondences. The quantitative registration results are shown in Table 10. We have the following observations:

1) First, STORM and PRE-RANSAC outperform other registration methods significantly.
2) As shown in the results from Table 10, our proposed STORM method is the second most efficient registration method and with also the second-best registration accuracy.
3) Given partial LiDAR scans, TrICP performs better than ICP.
4) Although FGR has a poor performance in terms of RMSE, RE, and TE, the ratio of successful registration of it is higher than ICP, TrICP, and 3DRegNet.
5) Although 3DRegNet is the fastest method, it performs the worst among these registration methods.

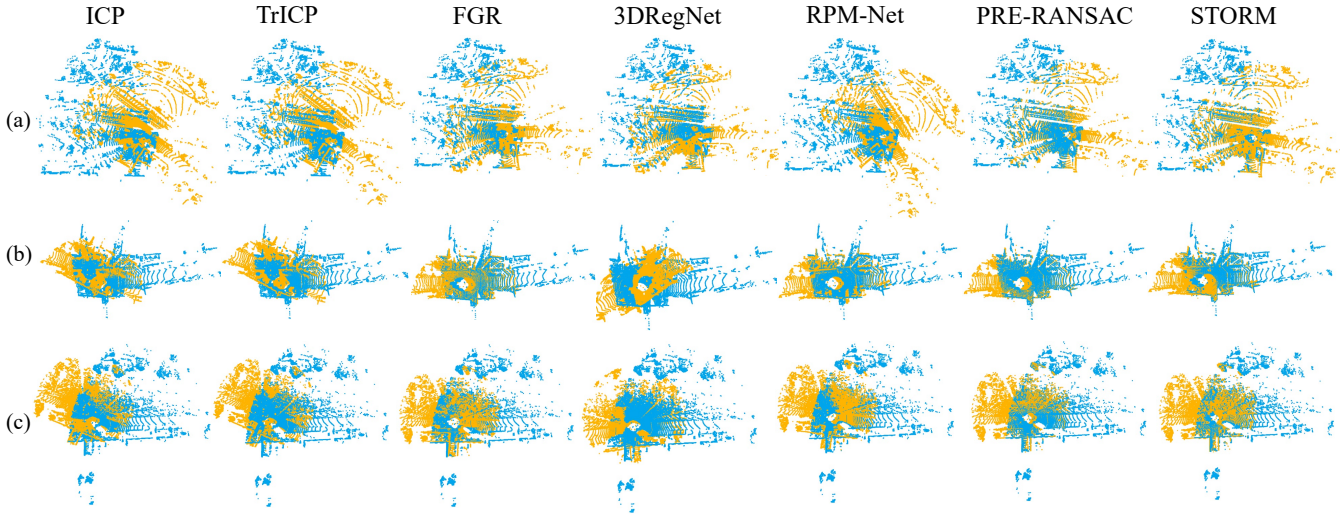


Fig. 12: The qualitative registration examples aligned by STORM and other registration methods from the test set of the partial KITTI dataset.

We can notice that although the input point clouds are very sparse, our proposed STORM method can still effectively utilize the coarse structural information to align the LIDAR scans well. STORM can perform well on the registration of partial objects and large scale of LIDAR scans. The several qualitative registration examples aligned by STORM and other registration methods are shown in Figure 12.

5 Conclusion

In this paper, we present STORM, a structure-based overlap matching method with an overlap prediction module for partial point cloud registration. Our overlap prediction module with differentiable sampling detects the points in overlap and facilitates the generation of correspondences in overlap, and we believe this module can also inspire the research in point cloud completion or other domains in 3D vision. Our framework can predict exact partial correspondences and align partial point clouds well utilizing contextual structural information of the input point clouds. The registration experiments on point clouds with different overlap ratios show that our proposed method obtains improvement compared to the state-of-the-art methods. Most methods perform worse when overlap rate decreases, while the proposed STORM method can still perform well even when the overlap ratio is small.

References

[1] P. J. Besl and N. D. McKay, "Method for registration of 3-D shapes," in *Sensor Fusion IV: Control Paradigms and Data Structures*, vol. 1611, 1992, pp. 586–606.

[2] A. Segal, D. Haehnel, and S. Thrun, "Generalized-ICP," in *Robotics: Science and Systems*, vol. 2, no. 4. Seattle, WA, 2009, p. 435.

[3] J. Yang, H. Li, D. Campbell, and Y. Jia, "Go-ICP: A globally optimal solution to 3D ICP point-set registration," vol. 38, no. 11. IEEE, 2015, pp. 2241–2254.

[4] Q.-Y. Zhou, J. Park, and V. Koltun, "Fast global registration," in *European Conference on Computer Vision*. Springer, 2016, pp. 766–782.

[5] K. S. Arun, T. S. Huang, and S. D. Blostein, "Least-squares fitting of two 3-d point sets," *IEEE Transactions on pattern analysis and machine intelligence*, no. 5, pp. 698–700, 1987.

[6] C. Choy, J. Park, and V. Koltun, "Fully convolutional geometric features," in *Proceedings of the IEEE International Conference on Computer Vision*, 2019, pp. 8958–8966.

[7] J. Du, R. Wang, and D. Cremers, "DH3D: Deep hierarchical 3D descriptors for robust large-scale 6DOF relocalization," in *European Conference on Computer Vision*, 2020, pp. 744–762.

[8] H. Deng, T. Birdal, and S. Ilic, "3D local features for direct pairwise registration," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 3244–3253.

[9] 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," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 6359–6367.

[10] R. Spezialetti, S. Salti, and L. D. Stefano, "Learning an effective equivariant 3D descriptor without supervision," in *Proceedings of the IEEE International Conference on Computer Vision*, 2019, pp. 6401–6410.

[11] S. Huang, Z. Gojcic, M. Usvyatsov, A. Wieser, and K. Schindler, "PREDATOR: Registration of 3D point clouds with low overlap," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 4267–4276.

[12] R. B. Rusu, N. Blodow, and M. Beetz, "Fast point feature histograms (FPFH) for 3D registration," in *IEEE International Conference on Robotics and Automation*, 2009, pp. 3212–3217.

[13] G. D. Pais, S. Ramalingam, V. M. Govindu, J. C. Nascimento, R. Chellappa, and P. Miraldo, "3DRegNet: A deep neural network for 3D point registration," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 7193–7203.

[14] C. Choy, W. Dong, and V. Koltun, "Deep global registration," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 2514–2523.

[15] Y. Aoki, H. Goforth, R. A. Srivatsan, and S. Lucey, "PointNetLK: Robust & efficient point cloud registration using PointNet," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 7163–7172.

[16] C. R. Qi, H. Su, K. Mo, and L. J. Guibas, "PointNet: Deep learning on point sets for 3D classification and segmentation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 652–660.

[17] Y. Wang and J. M. Solomon, "Deep Closest Point: Learning representations for point cloud registration," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 3523–3532.

[18] V. Sarode, A. Dhagat, R. A. Srivatsan, N. Zevallos, S. Lucey, and H. Choset, "MaskNet: A fully-convolutional network to estimate inlier points," in *2020 International Conference on 3D Vision (3DV)*. IEEE, 2020, pp. 1029–1038.

[19] Z. J. Yew and G. H. Lee, "RPM-Net: Robust point matching using learned features," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 11 824–11 833.

[20] R. Sinkhorn, "A relationship between arbitrary positive matrices and doubly stochastic matrices," *The annals of mathematical statistics*, vol. 35, no. 2, pp. 876–879, 1964.

[21] Y. Wang and J. M. Solomon, "PRNet: Self-supervised learning for partial-to-partial registration," in *Advances in Neural Information Processing Systems*, 2019, pp. 8812–8824.

[22] K. Zhang, M. Hao, J. Wang, C. W. de Silva, and C. Fu, "Linked Dynamic Graph CNN: Learning on point cloud via linking hierarchical features," 2019.

[23] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in Neural Information Processing Systems*, 2017, pp. 5998–6008.

[24] Z. Wu, S. Song, A. Khosla, F. Yu, L. Zhang, X. Tang, and J. Xiao, "3D shapeNets: A deep representation for volumetric shapes," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 1912–1920.

[25] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for autonomous driving? the KITTI vision benchmark suite," in *IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, 2012, pp. 3354–3361.

[26] D. Chetverikov, D. Svirko, D. Stepanov, and P. Krsek, "The trimmed iterative closest point algorithm," in *Object Recognition Supported by User Interaction for Service Robots*, vol. 3, 2002, pp. 545–548.

[27] S. Du, G. Xu, S. Zhang, X. Zhang, Y. Gao, and B. Chen, "Robust rigid registration algorithm based on pointwise correspondence and correntropy," *Pattern Recognition Letters*, vol. 132, pp. 91–98, 2020.

[28] X. Wang, X. Zhu, S. Ying, and C. Shen, "An accelerated and robust partial registration algorithm for point clouds," *IEEE Access*, vol. 8, pp. 156 504–156 518, 2020.

[29] Z. Dang, F. Wang, and M. Salzmann, "Learning 3D-3D correspondences for one-shot partial-to-partial registration," 2020.

[30] M. Cuturi, "Sinkhorn distances: Lightspeed computation of optimal transport," in *Advances in Neural Information Processing Systems*, 2013, pp. 2292–2300.

[31] J. Li, C. Zhang, Z. Xu, H. Zhou, and C. Zhang, "Iterative distance-aware similarity matrix convolution with mutual-supervised point elimination for efficient point cloud registration," in *European Conference on Computer Vision*, 2020, pp. 378–394.

[32] H. Xu, S. Liu, G. Wang, G. Liu, and B. Zeng, "OMNet: Learning overlapping mask for partial-to-partial point cloud registration," *arXiv preprint arXiv:2103.00937*, 2021.

[33] A. Hertz, R. Hanocka, R. Giryes, and D. Cohen-Or, "PointGMM: A neural GMM network for point clouds," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 12 054–12 063.

[34] H. Deng, T. Birdal, and S. Ilic, "PPFNet: Global context aware local features for robust 3D point matching," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 195–205.

[35] Y. Shen, C. Feng, Y. Yang, and D. Tian, "Mining point cloud local structures by kernel correlation and graph pooling," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 4548–4557.

[36] M. Dominguez, R. Dhamdhere, A. Petkar, S. Jain, S. Sah, and R. Ptucha, "General-purpose deep point cloud feature extractor," in *2018 IEEE Winter Conference on Applications of Computer Vision*, 2018, pp. 1972–1981.

[37] Y. Wang, Y. Sun, Z. Liu, S. E. Sarma, M. M. Bronstein, and J. M. Solomon, "Dynamic graph CNN for learning on point clouds," *ACM Transactions On Graphics (ToG)*, vol. 38, no. 5, pp. 1–12, 2019.

[38] L. Wang, Y. Huang, Y. Hou, S. Zhang, and J. Shan, "Graph attention convolution for point cloud semantic segmentation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 10 296–10 305.

[39] M. Simonovsky and N. Komodakis, "Dynamic edge-conditioned filters in convolutional neural networks on graphs," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 3693–3702.

[40] Q. Xu, X. Sun, C.-Y. Wu, P. Wang, and U. Neumann, "Grid-GCN for fast and scalable point cloud learning," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 5661–5670.

[41] W. Shi and R. Rajkumar, "Point-GNN: Graph neural network for 3D object detection in a point cloud," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 1711–1719.

[42] V. Sarode, X. Li, H. Goforth, Y. Aoki, R. A. Srivatsan, S. Lucey, and H. Choset, "PCRNet: Point cloud registration network using PointNet encoding," vol. abs/1908.07906, 2019.

[43] J. Yang, Z. Cao, and Q. Zhang, "A fast and robust local descriptor for 3d point cloud registration," *Information Sciences*, vol. 346, pp. 163–179, 2016.

[44] G. Georgakis, S. Karanam, Z. Wu, J. Ernst, and J. Košecká, "End-to-end learning of keypoint detector and descriptor for pose invariant 3D matching," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 1965–1973.

[45] M. Khoury, Q.-Y. Zhou, and V. Koltun, "Learning compact geometric features," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 153–161.

[46] S. Suwajanakorn, N. Snavely, J. J. Tompson, and M. Norouzi, "Discovery of latent 3D keypoints via end-to-end geometric reasoning," in *Advances in Neural Information Processing Systems*, 2018, pp. 2059–2070.

[47] C. R. Qi, L. Yi, H. Su, and L. J. Guibas, "PointNet++: Deep hierarchical feature learning on point sets in a metric space," in *Advances in Neural Information Processing Systems*, 2017, pp. 5099–5108.

[48] J. Yang, Q. Zhang, B. Ni, L. Li, J. Liu, M. Zhou, and Q. Tian, "Modeling point clouds with self-attention and gumbel subset sampling," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 3323–3332.

[49] I. Lang, A. Manor, and S. Avidan, "SampleNet: differentiable point cloud sampling," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 7578–7588.

[50] Q. Hu, B. Yang, L. Xie, S. Rosa, Y. Guo, Z. Wang, N. Trigoni, and A. Markham, "RandLA-Net: Efficient semantic segmentation of large-scale point clouds," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 11 108–11 117.

[51] E. Jang, S. Gu, and B. Poole, "Categorical reparametrization with Gumbel-Softmax," in *Proceedings International Conference on Learning Representations 2017*, 2017.

[52] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 4700–4708.

[53] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2014.

[54] H. Wei, Z. Qiao, Z. Liu, C. Suo, P. Yin, Y. Shen, H. Li, and H. Wang, "End-to-end 3d point cloud learning for registration task using virtual correspondences," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2020, pp. 2678–2683.

[55] Q.-Y. Zhou, J. Park, and V. Koltun, "Open3D: A modern library for 3D data processing," 2018.

[56] R. B. Rusu and S. Cousins, "3D is here: Point Cloud Library (PCL)," in *IEEE International Conference on Robotics and Automation (ICRA)*, Shanghai, China, May 9-13 2011.

[57] L. Van der Maaten and G. Hinton, "Visualizing data using t-sne." *Journal of machine learning research*, vol. 9, no. 11, 2008.

[58] G. Turk and M. Levoy, "Zippered polygon meshes from range images," in *Proceedings of the 21st annual conference on Computer graphics and interactive techniques*, 1994, pp. 311–318.