

# Deep Graph-based Spatial Consistency for Robust Non-rigid Point Cloud Registration

Zheng Qin<sup>1</sup> Hao Yu<sup>2</sup> Changjian Wang<sup>1</sup> Yuxing Peng<sup>1</sup> Kai Xu<sup>1\*</sup>

<sup>1</sup>National University of Defense Technology <sup>2</sup>Technical University of Munich

## Abstract

We study the problem of outlier correspondence pruning for non-rigid point cloud registration. In rigid registration, spatial consistency has been a commonly used criterion to discriminate outliers from inliers. It measures the compatibility of two correspondences by the discrepancy between the respective distances in two point clouds. However, spatial consistency no longer holds in non-rigid cases and outlier rejection for non-rigid registration has not been well studied. In this work, we propose Graph-based Spatial Consistency Network (GraphSCNet) to filter outliers for non-rigid registration. Our method is based on the fact that non-rigid deformations are usually locally rigid, or local shape preserving. We first design a local spatial consistency measure over the deformation graph of the point cloud, which evaluates the spatial compatibility only between the correspondences in the vicinity of a graph node. An attention-based non-rigid correspondence embedding module is then devised to learn a robust representation of non-rigid correspondences from local spatial consistency. Despite its simplicity, GraphSCNet effectively improves the quality of the putative correspondences and attains state-of-the-art performance on three challenging benchmarks. Our code and models are available at <https://github.com/qinzheng93/GraphSCNet>.

## 1. Introduction

Non-rigid point cloud registration is a fundamental and critical problem in computer graphics, computer vision, and robotics. It aims at recovering the non-rigid warping function that transforms a source point cloud to a target one. In practice, the two point clouds are usually incomplete and share partial and even low overlap, which considerably increases the difficulty of registration.

Estimating the warping function relies on extracting accurate correspondences. Benefiting from the recent advances in deep point representation [46, 53, 56, 57], learning-based matching methods [13, 17, 20, 24, 31, 45, 47, 55, 58]

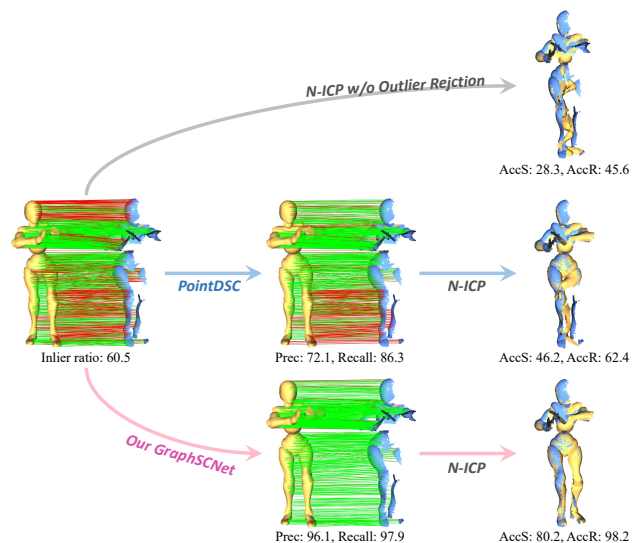


Figure 1. GraphSCNet Overview. Given a set of putative correspondences for non-rigid registration, GraphSCNet can accurately prune the outliers among them while preserve the inliers, which contributes to significantly better registration results.

have obtained significantly high quality of putative correspondences. However, similar success has yet to be achieved in deformable cases. Under significant deformation, these methods are inevitably prone to outliers, which can drastically degrade the accuracy of registration.

Outlier rejection is a common technique for robust point cloud registration. However, most existing methods focus on rigid registration. An effective method of outlier pruning for non-rigid registration has so far been missing. On the one hand, a rigid transformation can be pinned down by a small set of inlier correspondences (*e.g.*, a rotation can be determined by 3 non-collinear inliers) such that sampling consensus methods (*e.g.*, RANSAC [19]) can effectively remove the outliers and recover the alignment transformation in a hypothesize-and-verify manner. However, non-rigid registration requires dense and thoroughly-distributed correspondences to precisely describe the deformation, thus preventing the application of sampling consensus methods. On the other hand, rigid transformations preserve Euclidean distance between every pair of points. This spatial consis-

\*Corresponding author: kevin.kai.xu@gmail.com.

tency provides a strong necessary condition for finding inlier correspondences and has been extensively adopted in rigid registration [3, 11, 27, 28]. Apparently, such spatial consistency does not hold for non-rigid cases. These difficulties make outlier rejection for non-rigid registration a challenging problem.

We propose an outlier rejection network named *Graph-based Spatial Consistency Network* (GraphSCNet) tailored for non-rigid registration. Our method is designed around the local rigidity of non-rigid deformations, *i.e.*, non-rigid deformations are locally isometric such that the local shape of the point cloud is approximately preserved. We first design a *graph-based local spatial consistency* measure on the deformation graph [51] built over the source point cloud. It measures the geometric compatibility between the correspondences in the vicinity of a given graph node. Based on this measure, we propose an attention-based *graph-based correspondence embedding* module to extract *spatial-consistency-aware features* for correspondences, which are further used for discriminative classification of inlier and outlier. Thanks to the powerful local spatial consistency, our method can effectively prune outliers in putative correspondences while keeping as many inliers as possible. To our knowledge, our method is the first *learning-based outlier rejection for non-rigid point cloud registration*. Extensive experiments on three challenging benchmarks demonstrate clear superiority of our method. In particular, GraphSCNet outperforms the recent state-of-the-art NDP [32] by over 10% on AccS and AccR for both high- and low-overlap scenarios on the 4DMatch benchmark [31].

Our main contributions include:

- An outlier rejection network for non-rigid point cloud registration which is, to our knowledge, the first learning-based approach to outlier correspondence pruning for non-rigid scenarios.
- A graph-based local spatial consistency which measures the local geometric compatibility between correspondences within a local region.
- An attention-based correspondence embedding module which encodes the local spatial consistency for learning robust correspondence representation.

## 2. Related Work

**Point cloud correspondence.** Extracting accurate correspondences between point clouds plays a crucial role in computer vision and graphics tasks. Detection-based methods first extract geometrically-discriminative keypoints and their descriptors, either with hand-crafted [26, 48, 49, 54] or learning-based [1, 4, 13, 16, 17, 20, 24, 62] descriptors, which are then matched as correspondences. However, it is difficult to detect repeatable keypoints between point clouds, especially in low-overlap cases, such that detection-based methods still suffer from low inlier ratio. Recently,

detection-free methods [31, 47, 60, 61] bypass keypoint detection by considering all possible point pairs in a coarse-to-fine matching pipeline, which significantly improves matching and registration accuracy. There are also methods dedicated to non-rigid matching by explicitly modeling shape deformation [21, 50, 55] or leveraging functional maps [18, 35, 41]. And scene flow estimation methods [36, 45, 52, 58] predict the frame-to-frame motion of points in the scene. Although great progress has been made, existing methods are still prone to outliers, which significantly harms the registration performance.

**Non-rigid registration.** To describe the non-rigid deformation, the warping function can be formulated into different representations, *e.g.*, dense displacement field [31], dense affine transformation field [32], and embedded deformation graph [51]. Dense displacement field computes a 3D motion vector for each point in the scene, which is the most direct way to represent deformation. Dense affine transformation field computes an affine transformation for each point, which can better model complex deformation. Neural Deformation Pyramid [32] establishes a hierarchical dense affine transformation field with multiple MLPs for coarse-to-fine non-rigid registration. And embedded deformation graph [51] parameterizes the deformation with a set of graph nodes connected with undirected edges, where each node is associated with an affine transformation. This can be efficiently solved by the Non-rigid Iterative Closest Point (N-ICP) algorithm [29]. NNRT [8] proposes a differentiable N-ICP solver for end-to-end training, and [9] learns a deformation graph in a data-driven manner. There are still other warping function formulations, and we refer the readers to [15] for more details.

**Outlier rejection for point cloud registration.** Pruning outliers in rigid registration has been broadly studied. The most popular methods are RANSAC [19] and its variants [5, 6, 14], which solve for the rigid transformation in a hypothesize-and-verify manner. However, they suffer from slow convergence and could degenerate under high outlier ratio. Other methods [11, 28] leverage spatial consistency to suppress outliers. Recent learning-based methods [3, 12, 42] filter outliers with a neural network. PointDSC [3] designs a spatial consistency non-local module to prune outliers and attains promising rigid registration performance. Nevertheless, due to complex deformations, similar success has yet to be achieved in non-rigid registration. A closely related work to ours is [23], which extends the traditional spectral matching [28] technique to geodesic space. However, it computes pairwise geodesic distances between correspondences, which is time-consuming. And geodesic distance could be erroneous and unstable due to occlusion. In this work, we fill this gap with GraphSCNet for efficient and accurate non-rigid outlier pruning.

### 3. Method

#### 3.1. Overview

Given a source point cloud  $\mathcal{P} = \{\mathbf{p}_i \in \mathbb{R}^3 \mid i = 1, \dots, N\}$  and a target point cloud  $\mathcal{Q} = \{\mathbf{q}_i \in \mathbb{R}^3 \mid i = 1, \dots, M\}$ , non-rigid registration aims to recover the warping function  $\mathcal{W} : \mathbb{R}^3 \rightarrow \mathbb{R}^3$  that transforms  $\mathcal{P}$  to  $\mathcal{Q}$ . To solve for the warping function, a set of correspondences  $\mathcal{C} = \{(\mathbf{x}_i, \mathbf{y}_i) \in \mathbb{R}^6 \mid \mathbf{x}_i \in \mathcal{P}, \mathbf{y}_i \in \mathcal{Q}\}$  between two point clouds are first extracted. Then the warping function  $\mathcal{W}$  can be solved by minimizing the following cost function:

$$E = \lambda_c E_{\text{corr}} + \lambda_r E_{\text{reg}}, \quad (1)$$

where  $E_{\text{corr}}$  is a correspondence term which minimizes the residuals of the correspondences after being warped, and  $E_{\text{reg}}$  is a regularization term to encourage smoothness of deformations. Nevertheless, the putative correspondences usually contain numerous outliers, which significantly harms the registration accuracy. Due to complex deformations, it is difficult to filter outliers in non-rigid registration. In this work, we first present the graph-based local spatial consistency which measures the compatibility of correspondences within a local region, and then propose an outlier rejection network for non-rigid registration.

#### 3.2. Graph-based Local Spatial Consistency

Spatial consistency is a widely used criterion [3, 11, 28] to select inlier correspondences in rigid registration, *e.g.*, length consistency which preserves the distance between every pair of points under arbitrary rigid transformations. Given two correspondences  $c_i = (\mathbf{x}_i, \mathbf{y}_i)$  and  $c_j = (\mathbf{x}_j, \mathbf{y}_j)$ , the spatial consistency between them is computed as:

$$\theta_{i,j}^* = [1 - \frac{\delta_{i,j}^2}{\sigma_d^2}]_+, \quad (2)$$

where  $[\cdot]_+ = \max(0, \cdot)$ ,  $\delta_{i,j} = \|\mathbf{x}_i - \mathbf{x}_j\| - \|\mathbf{y}_i - \mathbf{y}_j\|$  is the difference between the respective distances in two point clouds, and  $\sigma_d$  is a hyper-parameter to control the sensitivity to distance variation. According to length consistency,  $\delta_{i,j}$  should be small if they are both inliers, making  $\theta_{i,j}^*$  close to 1. But if there is at least one outlier,  $\delta_{i,j}$  tends to be large due to the random distribution of the outliers, so  $\theta_{i,j}^*$  should be 0. See Fig. 2(a) for a detailed illustration. This provides strong geometric support to reject outliers in rigid scenarios.

However, global spatial consistency no longer holds in non-rigid scenarios, especially between two inliers far from each other, as the points in different parts of the scene could follow inconsistent movements (see Fig. 2(b)). But as noted in [25], the local geometric shape is expected to be preserved and the warping function should be locally isometric and nearly rigid, *i.e.*, local rigidity of deformations. Inspired by this insight, we propose to adopt spatial consistency in a local scope and devise a novel *graph-based local*

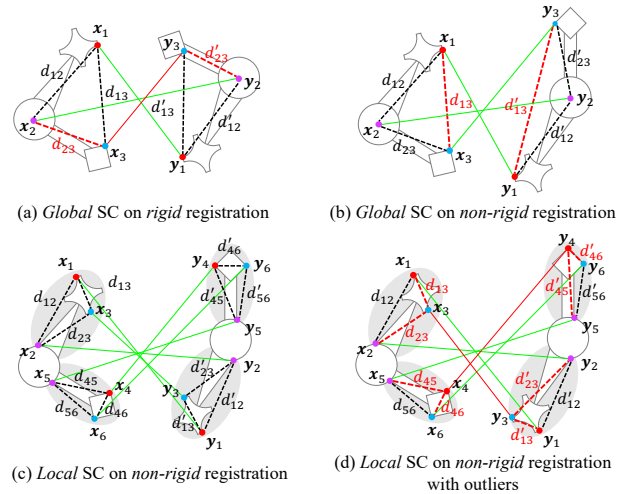


Figure 2. Graph-based local spatial consistency for non-rigid registration. The green lines represent the inliers while the outliers are in red. And the inconsistent distances between two correspondences are also highlighted in red dotted lines. (a) In rigid scenarios, the distances are identical between any two inliers, while being inconsistent if outliers exist. (b) In non-rigid scenarios, global spatial consistency does not hold as the distances between inliers could be different due to irregular movements. (c-d) Our graph-based local spatial consistency measures the distances between two correspondences within a local region based on local rigidity of deformations.

*spatial consistency*. Our method is based on the deformation graph [51] built over the source point cloud. We first sample a set of nodes  $\mathcal{V} = \{\mathbf{v}_j \in \mathbb{R}^3 \mid j = 1, \dots, V\}$  from  $\mathcal{P}$  using *uniform furthest point sampling*. We start from an arbitrary point in  $\mathcal{P}$  and iteratively add the furthest point to the sampled nodes as a new node. The sampling process is repeated until the distances from all points in  $\mathcal{P}$  to their nearest nodes are within  $\sigma_n$ . Then, we assign each correspondence  $c_i$  to its  $k$ -nearest nodes  $\mathcal{N}_i$  according to the distances in  $\mathcal{P}$ . Here  $\mathcal{N}_i$  is constructed according to the Euclidean distance. Given two points in a local region, their Euclidean distance is sufficiently consistent across two point clouds, but is more robust to occlusion than the geodesic distance. The set of correspondences assigned to a node  $\mathbf{v}_j$  is denoted as  $\mathcal{C}_j = \{c_i \mid \mathbf{v}_j \in \mathcal{N}_i\}$ . At last, our graph-based local spatial consistency is defined by computing Eq. (2) on the correspondence pairs assigned to a common node:

$$\theta_{i,j} = \begin{cases} [1 - \delta_{i,j}^2 / \sigma_d^2]_+, & c_i \in \mathcal{C}_v \wedge c_j \in \mathcal{C}_v \\ 0, & \text{otherwise} \end{cases} \quad (3)$$

Based on local rigidity,  $\theta_{i,j}$  is expected to be close to 1 if  $c_i$  and  $c_j$  are both inliers and be 0 otherwise. Fig. 2 compares our local spatial consistency with the global consistency.

An alternative way to define local spatial consistency is to construct a  $k$ NN graph around each correspondence in-

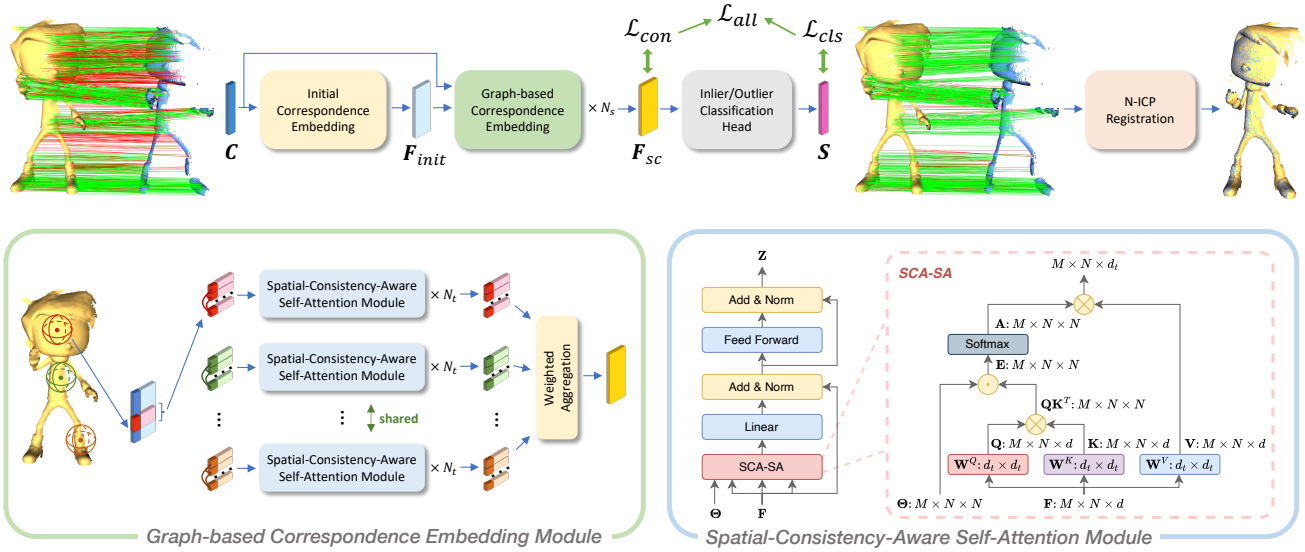


Figure 3. Pipeline of GraphSCNet. Given a set of putative correspondences  $\mathcal{C}$ , our method first extracts initial features  $\mathbf{F}_{init}$  from the point coordinates. The features are enhanced by a stack of graph-based non-rigid correspondence embedding module which encodes the local spatial consistency. The spatial-consistency-aware features  $\mathbf{F}_{sc}$  are then used to predict the confidence scores  $\mathbf{S}$ . At last, N-ICP is used to estimate the warping function.

stead of the sampled nodes. However, this manner could have two main problems. First, it requires more computation and memory usage to compute local spatial consistency around *every* correspondence. This seriously restricts its scalability to large point clouds or dense correspondences. Second, this fashion is sensitive to the density of putative correspondences. In practice, the distribution of correspondences could be extremely biased over the point cloud, and thus this manner is prone to be affected by the dense regions. On the contrary, as our method is designed around uniformly sampled nodes, it has great advantage in efficiency and is naturally robust to density variation. Please refer to Sec. 4.4 for more detailed comparisons.

### 3.3. Non-rigid Outlier Rejection Network

Based on the local spatial consistency, we then propose an attention-based *Graph-based Spatial Consistency Network (GraphSCNet)* for non-rigid outlier rejection. Given a set of putative correspondences, GraphSCNet leverages the graph-based local spatial consistency to remove the outliers from them. The overall pipeline is illustrated in Fig. 3.

**Initial feature embedding.** For each input correspondence, we first concatenate the coordinates of the two endpoints into a 6-d vector  $\mathbf{c}_i = [\mathbf{x}_i; \mathbf{y}_i]$ , which is then normalized to  $\hat{\mathbf{c}}_i$  by subtracting the average over all correspondences. Next,  $\hat{\mathbf{c}}_i$  is transformed using Fourier positional encoding in [39]. As mentioned in [32], low-frequency encoding benefits fitting relatively rigid motion while high-frequency one can better model highly non-rigid motion. Recalling our goal to better capture local rigidity, we use

relatively low frequency to encode the correspondences:

$$\mathbf{d}_i = [\hat{\mathbf{c}}; \sin(2^{-1}\hat{\mathbf{c}}); \cos(2^{-1}\hat{\mathbf{c}})] \in \mathbb{R}^{18}. \quad (4)$$

At last, the encoded correspondence matrix  $\mathbf{D} \in \mathbb{R}^{|\mathcal{C}| \times 18}$  is projected to a high-dimension feature matrix  $\mathbf{F}_{init} \in \mathbb{R}^{|\mathcal{C}| \times d}$  by a shallow MLP, which is used as the initial correspondence embedding. And group normalization [59] and LeakyReLU are used after each layer in the MLP.

**Graph-based correspondence embedding.** With the initial correspondence embedding, we then design a *Graph-based Correspondence Embedding Module* to enhance the feature representation of the correspondences with attention mechanism. The structure of this module is shown in Fig. 3 (bottom). Our method is based on the deformation graph constructed in Sec. 3.2 and consists of three steps.

First, we collect for each node  $\mathbf{v}_j$  the correspondences in  $\mathcal{C}_j$  and their associated features denoted as  $\mathbf{F}_j \in \mathbb{R}^{|\mathcal{C}_j| \times d}$ . Note that a correspondence could be assigned to more than one nodes and the nodes with  $\mathcal{C}_j = \emptyset$  are ignored. We also collect the local spatial consistency of the correspondence pairs in  $\mathcal{C}_j$ , denoted as  $\Theta_j \in \mathbb{R}^{|\mathcal{C}_j| \times |\mathcal{C}_j|}$ .

Next, we refine the features for the correspondences by a stack of *Spatial-Consistency-Aware Self-Attention (SCA-SA)* module. Specifically, the feature matrix  $\mathbf{F}_j$  is first projected into the query  $\mathbf{Q}_j$ , key  $\mathbf{K}_j$  and value  $\mathbf{V}_j$ :

$$\mathbf{Q}_j = \mathbf{F}_j \mathbf{W}^Q, \quad \mathbf{K}_j = \mathbf{F}_j \mathbf{W}^K, \quad \mathbf{V}_j = \mathbf{F}_j \mathbf{W}^V, \quad (5)$$

where  $\mathbf{W}^Q, \mathbf{W}^K, \mathbf{W}^V \in \mathbb{R}^{d \times d}$  are the projection weights for query, key and value, respectively. Inspired by [3], we leverage the local spatial consistency to *reweight* the atten-



tion scores in the original attention computation [56]:

$$\mathbf{Z}'_j = \text{LN}\left(\mathbf{F}_j + \text{MLP}\left(\text{Softmax}\left(\Theta_j \frac{\mathbf{Q}_j \mathbf{K}_j^T}{\sqrt{d}}\right) \mathbf{V}_j\right)\right), \quad (6)$$

where  $\text{LN}(\cdot)$  is layer normalization [2]. By injecting the graph-based local spatial consistency into self-attention, the correspondence pairs with strong spatial consistency are encouraged to have large attention scores, while the attention scores of the incompatible pairs are expected to be suppressed. This could push the outliers away from the inliers in the feature space, thus making the resultant features more discriminative. The attention features are further projected by a two-layer feedforward network with residual connection to obtain the final output features:

$$\mathbf{Z}_j = \text{LN}(\mathbf{Z}'_j + \text{MLP}(\mathbf{Z}'_j)). \quad (7)$$

Fig. 3 (bottom right) illustrates the structure and the computation graph of this module.

At last, for each correspondence, we consider its spatial compatibility w.r.t. different nodes and aggregate the features from all the nodes where it belongs as the final output features:

$$\mathbf{h}_i = \sum_{j \in \mathcal{N}_i} \alpha_{i,j} \mathbf{z}_i^j, \quad (8)$$

where  $\alpha_{i,j}$  is the skinning factor as in DynamicFusion [40]:

$$\alpha_{i,j} = \frac{\exp(-\|\mathbf{x}_i - \mathbf{v}_j\|^2 / (2\sigma_n^2))}{\sum_{k \in \mathcal{N}_i} \exp(-\|\mathbf{x}_i - \mathbf{v}_k\|^2 / (2\sigma_n^2))}. \quad (9)$$

In non-rigid scenarios, it is unreliable to predict whether one correspondence is inlier or not from merely a single local area as there could be large deformation in it. On the contrary, our method considers all neighboring regions, which could improve the robustness of the extracted features.

**Classification head.** Given the spatial-consistency-aware features  $\mathbf{F}_{\text{sc}} \in \mathbb{R}^{|\mathcal{C}| \times d}$  of the correspondences, we further adopt a three-layer MLP to predict the confidence score  $s_i$  being an inlier for each correspondence. Group normalization [59] and LeakyReLU are used after the first two layers in the MLP, and sigmoid activation is applied after the last layer. The correspondences whose confidence scores are above a certain threshold  $\tau_s$  are selected as inliers and the others are removed as outliers.

### 3.4. Deformation Estimation

After obtaining the pruned correspondences, an embedded deformation graph [51] is computed as the final warping function. We first construct a deformation graph  $\hat{\mathcal{G}} = \{\hat{\mathcal{V}}, \hat{\mathcal{E}}\}$  with a set of graph nodes  $\hat{\mathcal{V}}$  and undirected edges  $\hat{\mathcal{E}}$  connecting them. The nodes are sampled from  $\mathcal{P}$  as described in Sec. 3.2 with a distance threshold of  $\sigma_g$ . Each point in  $\mathcal{P}$  are assigned to its  $k_g$  nearest nodes and two

nodes are connected by an edge if there exists a point assigned to both of them.  $\mathcal{W}$  can then be approximated by a collection of local *rigid* transformations  $\{(\mathbf{R}_j, \mathbf{t}_j)\}$  associated with each node  $\hat{\mathbf{v}}_j$ :

$$\mathcal{W}(\mathbf{p}_i) = \sum_{j \in \mathcal{N}_i} \alpha_{i,j} (\mathbf{R}_j (\mathbf{p}_i - \hat{\mathbf{v}}_j) + \mathbf{t}_j + \hat{\mathbf{v}}_j), \quad (10)$$

where  $\alpha_{i,j}$  is computed as in Eq. (9). Our final optimization objective is shown as in Eq. (1), where the correspondence term is the mean squared distance between the correspondences and an as-rigid-as-possible [25] regularization term is applied to constrain the smoothness of deformations:

$$\begin{aligned} E_{\text{corr}} &= \sum_{(\mathbf{x}_i, \mathbf{y}_i) \in \mathcal{C}} \|\mathcal{W}(\mathbf{x}_i) - \mathbf{y}_i\|_2^2 \\ E_{\text{reg}} &= \sum_{(\mathbf{v}_i, \mathbf{v}_j) \in \mathcal{E}} \|\mathbf{R}_i (\mathbf{v}_j - \mathbf{v}_i) + \mathbf{v}_i + \mathbf{t}_i - (\mathbf{v}_j + \mathbf{t}_j)\|_2^2 \end{aligned} \quad (11)$$

This problem can be efficiently solved by Non-rigid ICP (N-ICP) algorithm [29, 51]. Note that although embedded deformation is used, GraphSCNet is agnostic to deformation models and thus can facilitate any correspondence-based non-rigid registration methods.

### 3.5. Loss Functions

Our model is trained with two types of loss functions, including a classification loss and a consistency loss. The overall loss function is computed as  $\mathcal{L}_{\text{all}} = \mathcal{L}_{\text{cls}} + \lambda \mathcal{L}_{\text{con}}$ .

**Classification loss.** We formulate the prediction of the confidence scores of the correspondences as a binary classification problem. As inliers and outliers are usually very imbalanced in the putative correspondences, we supervise the confidence scores with a binary focal loss [34]. The label of each correspondence  $c_i = (\mathbf{x}_i, \mathbf{y}_i)$  is computed as:

$$s_i^* = \begin{cases} 1, & \|\mathcal{W}^*(\mathbf{x}_i) - \mathbf{y}_i\| < \tau_d \\ 0, & \text{otherwise} \end{cases}, \quad (12)$$

where  $\mathcal{W}^*$  is the ground-truth deformation. And the classification loss is computed as:

$$\mathcal{L}_{\text{cls}} = -s_i^* (1 - s_i)^\gamma \log(s_i) - (1 - s_i^*) s_i^\gamma \log(1 - s_i), \quad (13)$$

where  $\gamma = 2$  is the focusing hyper-parameter as in [34].

**Consistency loss.** Inspired by PointDSC [3], we further adopt an auxiliary feature consistency loss so that the inliers are close to each other in the feature space and are far away from the outliers. However, due to the complexity of non-rigid deformations, feature consistency could not hold between two distant inlier correspondences. For this reason, we propose to supervise the feature consistency in each local region. For two correspondences  $c_x, c_y \in \mathcal{C}_j$  of node  $\mathbf{v}_j$ ,

we first compute their feature consistency as:

$$\delta_{x,y} = \left[1 - \frac{\|\hat{\mathbf{h}}_x - \hat{\mathbf{h}}_y\|^2}{\sigma_f^2}\right]_+, \quad (14)$$

where  $\hat{\mathbf{h}}_x$  and  $\hat{\mathbf{h}}_y$  are the correspondence features which are normalized onto a unit hyper-sphere, and  $\sigma_f$  is a learnable tolerance parameter. The consistency loss is computed as:

$$\mathcal{L}_{\text{con}} = \frac{1}{|\mathcal{V}|^2} \sum_{\mathbf{v}_j \in \mathcal{V}} \frac{1}{|\mathcal{C}_j|^2} \sum_{\mathbf{c}_x \in \mathcal{C}_j} \sum_{\mathbf{c}_y \in \mathcal{C}_j} \in \|\delta_{x,y} - \delta_{x,y}^*\|, \quad (15)$$

where the ground-truth targets  $\delta_{x,y}^* = 1$  if  $c_x$  and  $c_y$  are both inliers and  $\delta_{x,y}^* = 0$  otherwise.

## 4. Experiments

We evaluate the efficacy of GraphSCNet on three challenging benchmarks: 4DMatch [31] (Sec. 4.1), CAPE [38, 44] (Sec. 4.2) and DeepDeform [10] (Sec. 4.3). Extensive ablation studies are also provided to better understand our design choices (Sec. 4.4). More implementation details and network settings are introduced in the appendix.

**Metrics.** Following [31, 32], we mainly evaluate 4 metrics in the experiments: (1) *3D End Point Error* (EPE), the average errors over all warped points under the estimated and the ground-truth warp functions, (2) *3D Accuracy Strict* (AccS), the fraction of points whose EPEs are below 2.5cm or relative errors are below 2.5%, (3) *3D Accuracy Relaxed* (AccR), the fraction of points whose EPEs are below 5cm or relative errors are below 5%, and (4) *Outlier Ratio* (OR), the fraction of points whose relative errors are above 30%.

### 4.1. Evaluations on 4DMatch

**Dataset.** 4DMatch [31] is a challenging synthetic benchmark for non-rigid point cloud registration, which is constructed using the animation sequences from DeformingThings4D [33]. It consists of 1232 sequences for training, 176 for validation and 353 for testing. The point cloud pairs in the testing sequences are divided into 4DMatch and 4DLoMatch based on a overlapping ratio threshold of 45%. We use the preprocessed data from NDP [32] which removes the testing pairs with nearly-rigid movements to better evaluate the performance on non-rigid scenarios.

**Comparisons with state-of-the-art methods.** We first compare GraphSCNet to previous state-of-the-art non-rigid registration and scene flow estimation methods: NSFP [30], Nerfies [43], PointPWC-Net [58], FLOT [45], DGFm [18], SyNoRiM [22], and NDP [32]. To evaluate the generality of our method, we adopt two recent deep correspondence extractors in the experiments, Leopard [31] and GeoTransformer [47]. As shown in Tab. 1, our method outperforms the baselines by a large margin on both benchmarks, indicating the effectiveness of GraphSCNet. On the two most

| Model                    | 4DMatch      |             |             |            | 4DLoMatch    |             |             |             |
|--------------------------|--------------|-------------|-------------|------------|--------------|-------------|-------------|-------------|
|                          | EPE          | AccS        | AccR        | OR         | EPE          | AccS        | AccR        | OR          |
| NSFP [30]                | 0.265        | 8.7         | 18.7        | 65.0       | 0.495        | 0.4         | 1.6         | 84.8        |
| Nerfies [43]             | 0.280        | 12.7        | 25.4        | 58.9       | 0.498        | 1.1         | 3.0         | 82.2        |
| PointPWC-Net [58]        | 0.182        | 6.3         | 21.5        | 52.1       | 0.279        | 1.7         | 8.2         | 55.7        |
| FLOT [45]                | 0.133        | 7.7         | 27.2        | 40.5       | 0.210        | 2.7         | 13.1        | 42.5        |
| DGFm [18]                | 0.152        | 12.3        | 32.6        | 37.9       | 0.148        | 1.9         | 6.5         | 64.6        |
| SyNoRiM [22]             | 0.099        | 22.9        | 49.9        | 26.0       | 0.170        | 10.6        | 30.2        | 31.1        |
| NDP [32]                 | 0.077        | 61.3        | 74.1        | 17.3       | 0.177        | 26.6        | 41.1        | 33.8        |
| GraphSCNet (ours) + [31] | <b>0.042</b> | 70.1        | 83.8        | <b>9.2</b> | <u>0.102</u> | 40.0        | <b>59.1</b> | <b>17.5</b> |
| GraphSCNet (ours) + [47] | <u>0.043</u> | <b>72.3</b> | <b>84.4</b> | <u>9.4</u> | <u>0.121</u> | <b>41.0</b> | <u>58.3</u> | <u>21.0</u> |

Table 1. Comparisons with previous state-of-the-art methods on 4DMatch and 4DLoMatch. **Boldfaced** numbers highlight the best and the second best are underlined.

| Model                      | 4DMatch     |             |             |             | 4DLoMatch   |             |             |             |
|----------------------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|
|                            | Prec        | Recall      | AccS        | AccR        | Prec        | Recall      | AccS        | AccR        |
| <i>Leopard</i> [31]        |             |             |             |             |             |             |             |             |
| w/o outlier rejection      | 78.3        | 100.0       | 54.2        | 67.8        | 49.5        | 100.0       | 17.4        | 29.9        |
| VFC [37]                   | 83.6        | <u>93.2</u> | 63.6        | 76.4        | 54.6        | <u>84.1</u> | 26.2        | 40.3        |
| PointCN [42]               | 87.0        | 89.0        | 63.2        | 78.1        | 71.8        | 75.6        | 31.6        | 50.7        |
| PointDSC [3]               | <u>88.7</u> | 92.2        | <u>66.3</u> | <u>80.3</u> | <u>74.5</u> | 80.3        | <u>35.2</u> | <u>53.8</u> |
| GraphSCNet (ours)          | <b>93.0</b> | <b>95.7</b> | <b>70.1</b> | <b>83.8</b> | <b>83.0</b> | <b>88.6</b> | <b>40.0</b> | <b>59.1</b> |
| <i>oracle</i>              | 100.0       | 100.0       | 74.7        | 87.5        | 100.0       | 100.0       | 48.9        | 68.9        |
| <i>GeoTransformer</i> [47] |             |             |             |             |             |             |             |             |
| w/o outlier rejection      | 81.0        | 100.0       | 65.5        | 79.8        | 61.0        | 100.0       | 31.4        | 49.4        |
| VFC [37]                   | 83.0        | <u>96.0</u> | 67.1        | 79.6        | 63.2        | <b>91.6</b> | 33.8        | 50.5        |
| PointCN [42]               | 84.8        | 92.0        | 67.1        | 81.0        | 70.1        | 79.0        | 35.0        | 53.3        |
| PointDSC [3]               | <u>88.0</u> | 93.9        | <u>69.2</u> | <u>82.2</u> | <u>73.7</u> | 81.8        | <u>37.7</u> | <u>55.0</u> |
| GraphSCNet (ours)          | <b>92.2</b> | <b>96.9</b> | <b>72.3</b> | <b>84.4</b> | <b>82.6</b> | <b>86.8</b> | <b>41.0</b> | <b>58.3</b> |
| <i>oracle</i>              | 100.0       | 100.0       | 77.4        | 87.6        | 100.0       | 100.0       | 49.3        | 66.3        |

Table 2. Comparisons with outlier rejection baselines on 4DMatch and 4DLoMatch. **Boldfaced** numbers highlight the best and the second best are underlined.

important metrics *AccS* and *AccR*, our method significantly surpasses the previous best NDP by 11 percentage points (pp) on 4DMatch and 14 pp on 4DLoMatch. Note that benefiting from the high-quality correspondences, our method achieves the new state-of-the-art results simply with N-ICP and achieves 10 times acceleration than NDP (0.2s vs. 2s).

**Comparisons with outlier rejection methods.** We compare to one traditional outlier rejection method, VFC [37], and two recent learning-based methods for *rigid* registration, PointCN from 3DRegNet [42] and PointDSC [3], to evaluate the efficacy of our method. We also report the precision and recall of the predicted inliers to compare the inlier classification performance. For fair comparison, we adopt similar network macro-architecture for all the models and use the same configurations in N-ICP. For PointDSC, 2048 correspondences are randomly sampled to avoid too huge memory footprint. We show the results on two correspondence extractors (Leopard and GeoTransformer) to compare the generality of the methods. And the results using the ground-truth inliers are also reported as *oracle*. As shown in Tab. 2, the models with outlier rejection significantly surpass the models that do not prune outliers. And our method outperforms PointCN and PointDSC by a large margin on both benchmarks and attains very close results to the *oracle*, demonstrating the strong effectiveness of our design.

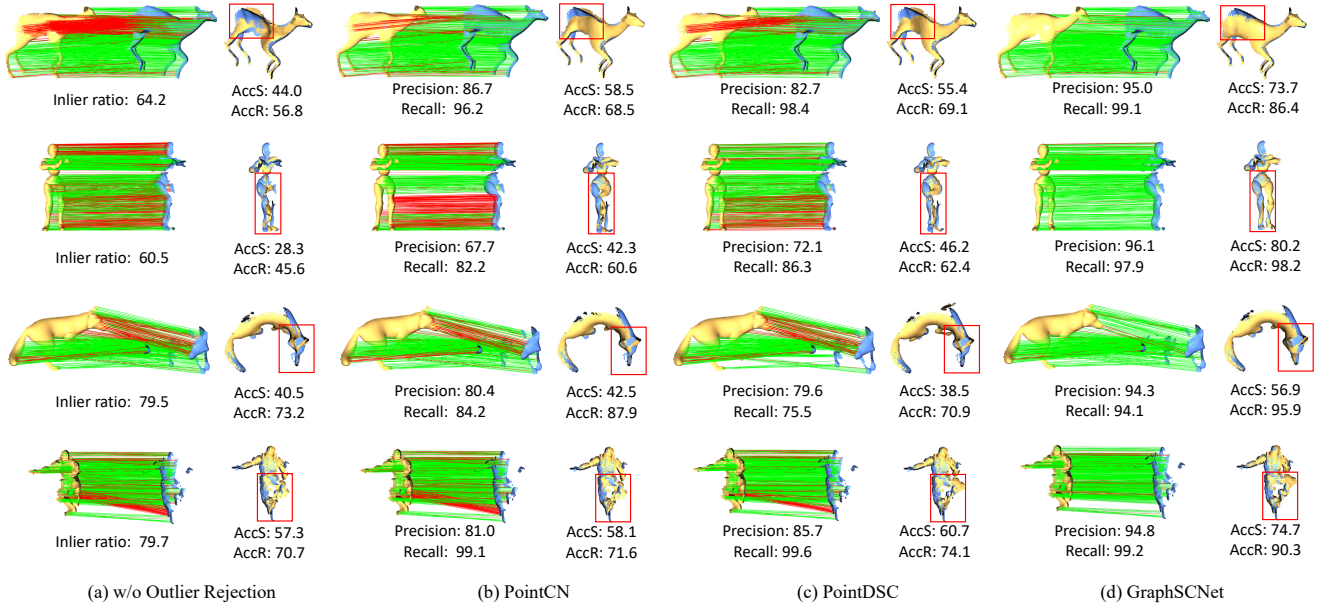


Figure 4. Comparison of different methods on 4DMatch and 4DLoMatch. Our method provides much better outlier rejection results in low-overlap and large-deformation scenes and achieves better registration results. Benefiting from the more accurate correspondences, our method successfully recover the geometry in non-overlap regions (see the registration results enclosed by the red boxes).

Note that our method attains both better precision and recall, especially in low-overlap scenarios, which means it *rejects more outliers while preserving more inliers*. This guarantees more thoroughly-distributed correspondences, facilitating more accurate non-rigid registration.

**Qualitative results.** Fig. 4 visualizes the correspondences and the registration results of different methods. Compared with the baselines, GraphSCNet prunes outliers more accurately while preserves more inliers, especially in low-overlap or large-deformation scenarios. And our method performs quite well in the scenes with symmetry (see the 2<sup>nd</sup> row) or complex geometry (see the 4<sup>th</sup> row). As there is little interference from outliers, our method successfully recover the geometry in non-overlap regions (see the registration results enclosed by the red box).

## 4.2. Generalization from 4DMatch to CAPE

**Dataset.** CAPE [38, 44] contains the complete scans of dynamic clothed humans. It consists of 15 human subjects and provides accurate 3D mesh registrations. We use the data preprocessed by [22] where each point cloud contains 8192 points. To better study the performance on large deformations, we first align each point cloud pair with a rigid transformation by solving a mean least square problem [7], and ignore the pairs whose mean residuals are below 10cm. At last, we obtain 11288 point cloud pairs for evaluation.

**Quantitative results.** We investigate the generality of our method on CAPE. To this end, we train all the models on 4DMatch and directly evaluate the models on CAPE without fine-tuning. The input correspondences are extracted with GeoTransformer [47] which is also trained 4DMatch.

| Model                               | Prec        | Recall      | EPE          | AccS        | AccR        | OR          |
|-------------------------------------|-------------|-------------|--------------|-------------|-------------|-------------|
| <i>Generalization to CAPE</i>       |             |             |              |             |             |             |
| w/o outlier rejection               | 38.0        | 100.0       | 0.143        | 20.9        | 41.3        | 68.2        |
| PointCN [42]                        | 43.9        | <u>65.0</u> | 0.132        | 28.3        | 50.3        | 63.6        |
| PointDSC [3]                        | <u>55.8</u> | 63.2        | 0.122        | 34.4        | <u>56.1</u> | 60.5        |
| GraphSCNet (ours)                   | <b>69.3</b> | <b>83.6</b> | <b>0.090</b> | <b>47.5</b> | <b>67.1</b> | <b>50.1</b> |
| <i>Generalization to DeepDeform</i> |             |             |              |             |             |             |
| w/o outlier rejection               | 43.3        | 100.0       | 0.146        | 17.8        | 37.0        | 64.0        |
| PointCN [42]                        | 47.1        | <b>68.0</b> | 0.149        | 18.9        | 38.3        | 63.9        |
| PointDSC [3]                        | <u>54.1</u> | <u>65.1</u> | <u>0.139</u> | <u>21.7</u> | <u>42.1</u> | <u>61.4</u> |
| GraphSCNet (ours)                   | <b>60.7</b> | 64.2        | <b>0.134</b> | <b>24.1</b> | <b>44.2</b> | <b>59.1</b> |

Table 3. Generalization results from 4DMatch to DeepDeform and CAPE. **Boldfaced** numbers highlight the best and the second best are underlined.

As shown in Tab. 3(top), GraphSCNet achieves significant improvements over the baseline methods. Our method surpasses the second best PointDSC by over 13 pp on precision and AccS, 20 pp on recall, and 11 pp on AccR. Note that our method not only achieves better precision, but very high recall, indicating that it prunes more outliers while preserves more inliers. As the human pose variations in CAPE are relatively large, the baseline methods fail to effectively distinguish inlier and outlier correspondences. Nevertheless, our method is still effective and has strong robustness thanks to the local spatial consistency. Please refer to the appendix for more detailed qualitative results.

## 4.3. Generalization from 4DMatch to DeepDeform

**Dataset.** DeepDeform [10] consists of real-world partial RGB-D images scanned by a RGB-D camera. It contains 400 scenes with over 390K RGB-D frames. We project the depth images into point clouds and leverage the dense scene

| Model                     | 4DMatch     |             |             |             | 4DLoMatch   |             |             |             |
|---------------------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|
|                           | Prec        | Recall      | AccS        | AccR        | Prec        | Recall      | AccS        | AccR        |
| (a.1) $\sigma_n = 0.04$   | 91.4        | 96.4        | 70.6        | 83.5        | 79.3        | 85.2        | 38.4        | 56.5        |
| (a.2) $\sigma_n = 0.08^*$ | <u>92.2</u> | <b>96.9</b> | <b>72.3</b> | <b>84.4</b> | <u>82.6</u> | <u>86.8</u> | 41.0        | 58.3        |
| (a.3) $\sigma_n = 0.12$   | <b>92.7</b> | 96.4        | <u>72.0</u> | <u>84.2</u> | <b>82.9</b> | 86.4        | <b>41.8</b> | <b>58.9</b> |
| (a.4) $\sigma_n = 0.16$   | 92.1        | <u>96.8</u> | 71.7        | 84.0        | 82.4        | <b>87.1</b> | 41.7        | 58.8        |
| (a.5) $\sigma_n = 0.32$   | 89.2        | 95.4        | 69.5        | 82.4        | 78.6        | 83.7        | 39.3        | 56.6        |
| (b.1) w/ FPS*             | <b>92.2</b> | 96.9        | 72.3        | <b>84.4</b> | <b>82.6</b> | <b>86.8</b> | <b>41.0</b> | <b>58.3</b> |
| (b.2) w/ RS               | 92.1        | <b>97.0</b> | <b>72.5</b> | <b>84.4</b> | 82.2        | 86.1        | <b>41.0</b> | 58.1        |
| (c.1) Nodes w/ FPS*       | <b>92.2</b> | <b>96.9</b> | <b>72.3</b> | <b>84.4</b> | <b>82.6</b> | <b>86.8</b> | <b>41.0</b> | <b>58.3</b> |
| (c.2) $k$ NN graph        | 91.4        | 96.5        | 71.9        | 84.0        | 79.5        | 85.3        | 40.1        | 57.4        |
| (d.1) $k = 1$             | 89.1        | 96.6        | 59.7        | 69.4        | 75.6        | 85.9        | 23.6        | 31.4        |
| (d.2) $k = 3$             | <u>92.0</u> | <b>97.1</b> | <u>71.1</u> | 84.0        | 81.6        | 86.6        | 38.3        | 55.5        |
| (d.3) $k = 6^*$           | <b>92.2</b> | 96.9        | <b>72.3</b> | <b>84.4</b> | <b>82.6</b> | <b>86.8</b> | <b>41.0</b> | <b>58.3</b> |
| (d.4) $k = 9$             | <b>92.2</b> | <u>96.6</u> | 69.8        | 83.0        | <b>82.6</b> | 85.8        | <u>38.6</u> | <u>56.5</u> |

Table 4. Ablation studies on 4DMatch and 4DLoMatch. Asterisk (\*) indicates the default settings in our method. **FPS**: furthest point sampling. **RS**: random sampling. **Boldfaced** numbers highlight the best and the second best are underlined.

flow annotations to construct the point cloud pairs. And we adopt the same preprocessing as in Sec. 4.2 with a threshold of 5cm to remove the nearly-rigid pairs. As a result, we obtain 1011 point cloud pairs for evaluation.

**Quantitative results.** Following Sec. 4.2, we train all the models on 4DMatch and directly test them on DeepDeform without fine-tuning to investigate the generality of our method to real-world scenarios. And GeoTransformer trained 4DMatch is adopted to generate the initial correspondences. As shown in Tab. 3(bottom), PointCN achieves only marginal improvements over the model without outlier rejection. As it determines outliers based on only the coordinates of each single correspondence without considering the geometry of point clouds, its generality is unsatisfactory. PointDSC obtains considerably better results than PointCN benefiting from the global spatial consistency. However, it still lacks the capability to handle non-rigid deformations. On the contrary, our method outperforms the three baseline methods by a large margin as it leverages local rigidity to remove outliers, which better models the deformations. These results have demonstrated the strong transferability and generality of our method to unseen domains.

#### 4.4. Ablation Studies

We further conduct extensive ablation studies to provide a better understanding of the design choices in GraphSCNet. In the following experiments, we use GeoTransformer as the prior correspondence extractor.

**Node sampling.** We first study the influence of the distribution of nodes. First, we vary the distance threshold  $\sigma_n$  from 0.04 to 0.32 for node sampling. Note that we do not change the node sampling settings in N-ICP. As shown in Tab. 4(a), our method attains similar performance under different  $\sigma_n$ , and the performance get worse if  $\sigma_n$  is too small or too large. If  $\sigma_n$  is too small, each local region is limited so that there could not be enough context. But if  $\sigma_n$  is too

large, the local spatial consistency could be broken.

Next, we replace the uniform furthest point sampling with uniform random sampling in Tab. 4(b), and two sampling methods achieve comparable results. The model with furthest point sampling performs slightly better as it generates more stably distributed nodes. These results prove the strong robustness of our method to the distribution of nodes.

**Graph construction.** We further study the influence of the graph structure to compute local spatial consistency. First, we build a  $k$ NN graph which connects each correspondence to its  $k$  nearest correspondences as described in Sec. 3.2, where  $k = 32$  to fit the GPU memory and the same network architecture is used for outlier rejection. From Tab. 4(c), our method surpasses this counterpart on all the metrics, especially in low-overlap cases. We argue that our advantage is two-fold. First, each correspondence can only be connected to limited neighbors in the  $k$ NN graph due to high memory usage, which fails to provide enough geometric context. Second, the learned features are predominated by the spatial areas with high densities in the  $k$ NN graph such that the representation ability is degraded.

Next, we vary the number of nodes  $k$  where each correspondence is assigned when building the graph in Tab. 4(d). The results are significantly degraded when  $k = 1$ . On the one hand, considering only one local region for each correspondence harms the robustness of the learned features to large deformation. On the other hand, the geometric context in a local region is insufficient when  $k$  is too small. And the performance also gets worse with a large  $k$ . The distances between correspondences and nodes could be too far such that the local spatial consistency is broken in this case.

## 5. Conclusion

We have proposed GraphSCNet to effectively prune outlier correspondences for non-rigid point cloud registration. Based on the local rigidity of deformations, we introduce a graph-based local spatial consistency criterion to measure the compatibility of two correspondences. Next, we further design a non-rigid correspondence embedding module which leverages the local spatial consistency to extract correspondence features. The spatial-consistency-aware correspondence features are further used to filter outliers. To our knowledge, this is the first learning-based outlier rejection method for non-rigid registration. Extensive experiments on three benchmarks have demonstrated the efficacy of our method. However, as our method is based on deformation graphs and local rigidity, it could have difficulty in modeling sudden geometric changes. In the future, we would like to extend our method for end-to-end non-rigid registration.

**Acknowledgement.** This work is supported in part by the NSFC (62132021), the National Key R&D Program of China (2018AAA0102200, 2021ZD0140408), and the Logistics Major Research Plan (145BHQ090003000X03).



## References

- [1] Sheng Ao, Qingyong Hu, Bo Yang, Andrew Markham, and Yulan Guo. Spinnet: Learning a general surface descriptor for 3d point cloud registration. In *CVPR*, 2021. 2
- [2] Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. Layer normalization. *arXiv preprint arXiv:1607.06450*, 2016. 5
- [3] Xuyang Bai, Zixin Luo, Lei Zhou, Hongkai Chen, Lei Li, Zeyu Hu, Hongbo Fu, and Chiew-Lan Tai. Pointdsc: Robust point cloud registration using deep spatial consistency. In *CVPR*, 2021. 2, 3, 4, 5, 6, 7
- [4] Xuyang Bai, Zixin Luo, Lei Zhou, Hongbo Fu, Long Quan, and Chiew-Lan Tai. D3feat: Joint learning of dense detection and description of 3d local features. In *CVPR*, 2020. 2
- [5] Daniel Barath and Jiří Matas. Graph-cut ransac. In *CVPR*, 2018. 2
- [6] Daniel Barath, Jana Noskova, Maksym Ivashechkin, and Jiri Matas. Magsac++, a fast, reliable and accurate robust estimator. In *CVPR*, 2020. 2
- [7] Paul J Besl and Neil D McKay. Method for registration of 3-d shapes. In *Sensor fusion IV: control paradigms and data structures*, 1992. 7
- [8] Aljaz Bozic, Pablo Palafox, Michael Zollhöfer, Angela Dai, Justus Thies, and Matthias Nießner. Neural non-rigid tracking. *NeurIPS*, 2020. 2
- [9] Aljaz Bozic, Pablo Palafox, Michael Zollhofer, Justus Thies, Angela Dai, and Matthias Nießner. Neural deformation graphs for globally-consistent non-rigid reconstruction. In *CVPR*, 2021. 2
- [10] Aljaz Bozic, Michael Zollhofer, Christian Theobalt, and Matthias Nießner. Deepdeform: Learning non-rigid rgb-d reconstruction with semi-supervised data. In *CVPR*, 2020. 6, 7
- [11] Zhi Chen, Kun Sun, Fan Yang, and Wenbing Tao. Sc2-pcr: A second order spatial compatibility for efficient and robust point cloud registration. In *CVPR*, 2022. 2, 3
- [12] Christopher Choy, Wei Dong, and Vladlen Koltun. Deep global registration. In *CVPR*, 2020. 2
- [13] Christopher Choy, Jaesik Park, and Vladlen Koltun. Fully convolutional geometric features. In *CVPR*, 2019. 1, 2
- [14] Ondřej Chum, Jiří Matas, and Josef Kittler. Locally optimized ransac. In *JPRS*, 2003. 2
- [15] Bailin Deng, Yuxin Yao, Roberto M Dyke, and Juyong Zhang. A survey of non-rigid 3d registration. In *Computer Graphics Forum*, 2022. 2
- [16] Haowen Deng, Tolga Birdal, and Slobodan Ilic. Ppf-foldnet: Unsupervised learning of rotation invariant 3d local descriptors. In *ECCV*, 2018. 2
- [17] Haowen Deng, Tolga Birdal, and Slobodan Ilic. Ppfnet: Global context aware local features for robust 3d point matching. In *CVPR*, 2018. 1, 2
- [18] Nicolas Donati, Abhishek Sharma, and Maks Ovsjanikov. Deep geometric functional maps: Robust feature learning for shape correspondence. In *CVPR*, 2020. 2, 6
- [19] Martin A Fischler and Robert C Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 1981. 1, 2
- [20] Zan Gojcic, Caifa Zhou, Jan D Wegner, and Andreas Wieser. The perfect match: 3d point cloud matching with smoothed densities. In *CVPR*, 2019. 1, 2
- [21] Thibault Groueix, Matthew Fisher, Vladimir G Kim, Bryan C Russell, and Mathieu Aubry. 3d-coded: 3d correspondences by deep deformation. In *ECCV*, 2018. 2
- [22] Jiahui Huang, Tolga Birdal, Zan Gojcic, Leonidas J Guibas, and Shi-Min Hu. Multiway non-rigid point cloud registration via learned functional map synchronization. *IEEE TPAMI*, 2022. 6, 7
- [23] Qi-Xing Huang, Bart Adams, Martin Wicke, and Leonidas J Guibas. Non-rigid registration under isometric deformations. In *Computer Graphics Forum*, 2008. 2
- [24] Shengyu Huang, Zan Gojcic, Mikhail Usvyatsov, Andreas Wieser, and Konrad Schindler. Predator: Registration of 3d point clouds with low overlap. In *CVPR*, 2021. 1, 2
- [25] Takeo Igarashi, Tomer Moscovich, and John F Hughes. As-rigid-as-possible shape manipulation. *ACM TOG*, 2005. 3, 5
- [26] Andrew E Johnson and Martial Hebert. Using spin images for efficient object recognition in cluttered 3d scenes. *IEEE TPAMI*, 1999. 2
- [27] Junha Lee, Seungwook Kim, Minsu Cho, and Jaesik Park. Deep hough voting for robust global registration. In *ICCV*, 2021. 2
- [28] Marius Leordeanu and Martial Hebert. A spectral technique for correspondence problems using pairwise constraints. In *ICCV*, 2005. 2, 3
- [29] Hao Li, Robert W Sumner, and Mark Pauly. Global correspondence optimization for non-rigid registration of depth scans. In *Computer graphics forum*, 2008. 2, 5
- [30] Xueqian Li, Jhony Kaesemodel Pontes, and Simon Lucey. Neural scene flow prior. *NeurIPS*, 2021. 6
- [31] Yang Li and Tatsuya Harada. Leopard: Learning partial point cloud matching in rigid and deformable scenes. In *CVPR*, 2022. 1, 2, 6
- [32] Yang Li and Tatsuya Harada. Non-rigid point cloud registration with neural deformation pyramid. In *NeurIPS*, 2022. 2, 4, 6
- [33] Yang Li, Hikari Takehara, Takafumi Taketomi, Bo Zheng, and Matthias Nießner. 4dcomplete: Non-rigid motion estimation beyond the observable surface. In *ICCV*, 2021. 6
- [34] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *ICCV*, 2017. 5
- [35] Or Litany, Tal Remez, Emanuele Rodola, Alex Bronstein, and Michael Bronstein. Deep functional maps: Structured prediction for dense shape correspondence. In *ICCV*, 2017. 2
- [36] Xingyu Liu, Charles R Qi, and Leonidas J Guibas. Flownet3d: Learning scene flow in 3d point clouds. In *CVPR*, 2019. 2
- [37] Jiayi Ma, Ji Zhao, Jinwen Tian, Alan L Yuille, and Zhuowen Tu. Robust point matching via vector field consensus. *IEEE TIP*, 2014. 6

- [38] Qianli Ma, Jinlong Yang, Anurag Ranjan, Sergi Pujades, Gerard Pons-Moll, Siyu Tang, and Michael J Black. Learning to dress 3d people in generative clothing. In *CVPR*, 2020. 6, 7
- [39] Ben Mildenhall, Pratul P Srinivasan, Matthew Tancik, Jonathan T Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. In *ECCV*, 2020. 4
- [40] Richard A Newcombe, Dieter Fox, and Steven M Seitz. Dynamicfusion: Reconstruction and tracking of non-rigid scenes in real-time. In *CVPR*, 2015. 5
- [41] Maks Ovsjanikov, Mirela Ben-Chen, Justin Solomon, Adrian Butscher, and Leonidas Guibas. Functional maps: a flexible representation of maps between shapes. *ACM TOG*, 2012. 2
- [42] G Dias Pais, Srikumar Ramalingam, Venu Madhav Govindu, Jacinto C Nascimento, Rama Chellappa, and Pedro Miraldo. 3dregnet: A deep neural network for 3d point registration. In *CVPR*, 2020. 2, 6, 7
- [43] Keunhong Park, Utkarsh Sinha, Jonathan T Barron, Sofien Bouaziz, Dan B Goldman, Steven M Seitz, and Ricardo Martin-Brualla. Nerfies: Deformable neural radiance fields. In *ICCV*, 2021. 6
- [44] Gerard Pons-Moll, Sergi Pujades, Sonny Hu, and Michael J Black. Clothcap: Seamless 4d clothing capture and retargeting. *ACM TOG*, 2017. 6, 7
- [45] Gilles Puy, Alexandre Boulch, and Renaud Marlet. Flot: Scene flow on point clouds guided by optimal transport. In *ECCV*, 2020. 1, 2, 6
- [46] Charles R Qi, Hao Su, Kaichun Mo, and Leonidas J Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *CVPR*, 2017. 1
- [47] Zheng Qin, Hao Yu, Changjian Wang, Yulan Guo, Yuxing Peng, and kai Xu. Geometric transformer for fast and robust point cloud registration. In *CVPR*, 2022. 1, 2, 6, 7
- [48] Radu Bogdan Rusu, Nico Blodow, and Michael Beetz. Fast point feature histograms (fpfh) for 3d registration. In *ICRA*, 2009. 2
- [49] Radu Bogdan Rusu, Nico Blodow, Zoltan Csaba Marton, and Michael Beetz. Aligning point cloud views using persistent feature histograms. In *IROS*, 2008. 2
- [50] Mahdi Saleh, Shun-Cheng Wu, Luca Cosmo, Nassir Navab, Benjamin Busam, and Federico Tombari. Bending graphs: Hierarchical shape matching using gated optimal transport. In *CVPR*, 2022. 2
- [51] Robert W Sumner, Johannes Schmid, and Mark Pauly. Embedded deformation for shape manipulation. In *SIGGRAPH*. 2007. 2, 3, 5
- [52] Zachary Teed and Jia Deng. Raft-3d: Scene flow using rigid-motion embeddings. In *CVPR*, 2021. 2
- [53] Hugues Thomas, Charles R Qi, Jean-Emmanuel Deschaud, Beatriz Marcotegui, François Goulette, and Leonidas J Guibas. Kpconv: Flexible and deformable convolution for point clouds. In *ICCV*, 2019. 1
- [54] Federico Tombari, Samuele Salti, and Luigi Di Stefano. Unique signatures of histograms for local surface description. In *ECCV*, 2010. 2
- [55] Giovanni Trappolini, Luca Cosmo, Luca Moschella, Riccardo Marin, Simone Melzi, and Emanuele Rodolà. Shape registration in the time of transformers. *NeurIPS*, 2021. 1, 2
- [56] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *NeurIPS*, 2017. 1, 5
- [57] 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 TOG*, 2019. 1
- [58] Wenxuan Wu, Zhi Yuan Wang, Zhuwen Li, Wei Liu, and Li Fuxin. Pointpwc-net: Cost volume on point clouds for (self-) supervised scene flow estimation. In *ECCV*, 2020. 1, 2, 6
- [59] Yuxin Wu and Kaiming He. Group normalization. In *ECCV*, 2018. 4, 5
- [60] Hao Yu, Fu Li, Mahdi Saleh, Benjamin Busam, and Slobodan Ilic. Cofinet: Reliable coarse-to-fine correspondences for robust pointcloud registration. *NeurIPS*, 2021. 2
- [61] Hao Yu, Zheng Qin, Ji Hou, Mahdi Saleh, Dongsheng Li, Benjamin Busam, and Slobodan Ilic. Rotation-invariant transformer for point cloud matching. *arXiv preprint arXiv:2303.08231*, 2023. 2
- [62] Andy Zeng, Shuran Song, Matthias Nießner, Matthew Fisher, Jianxiong Xiao, and Thomas Funkhouser. 3dmatch: Learning local geometric descriptors from rgb-d reconstructions. In *CVPR*, 2017. 2